ADONIS®

Version 3.9

Volume 4

Method Definition

and

Administrator's Manual

www.boc-eu.com

Volume 1: Introduction

Volume 2: User's manual

Volume 3: Method manual

Volume 4: Method definition and administrator's manual

Volume 5: Installation and Database administrator's manual

BOC Group - addresses:

Austria: BOC Asset Management GmbH
A-1010 Vienna, Baeckerstrasse 5

BOC Information Technologies Consulting AG
BOC Information Systems GmbH
A-1010 Vienna, Wipplingerstrasse 1

BOC Unternehmensberatung GmbH
A-1010 Vienna, Rabensteig

Germany: BOC Information Technologies Consulting GmbH
Mosse-Palast am Leipziger Platz
D-10117 Berlin, Vossstrasse 22

Ibérica: BOC Business Objectives Consulting Ibérica, S.A.
E-28006 Madrid, Calle Velàzquez, 71

Ireland: BOC Information Technologies Consulting Ltd.
Haddington Hall
IRL-Dublin 4, 80 Haddington road

Greece: BOC Information Technologies Consulting EPE
GR-10680 Athen, Mavromichali 16

Poland: BOC Information Technologies Consulting Sp. z o.o.
PL-00-679 Warschau, ul. Wilcza 44

Hotline: hotline@boc-eu.com

Internet: http://www.boc-eu.com
http://www.boc-us.com

Table of Contents

Table of Contents

Preface

Welcome to

the **integrated Business Process and Knowledge Management Toolkit** from the *Management Office* of the BOC Group.

ADONIS offers you a wide range of various functionality in the following operational areas:

E-Business:
E-business application development, as well as constant and integrated modelling of business models, Business Processes, products, IT systems and IT infrastructures.

Business process management:
Acquisition, modelling, analysis, simulation, evaluation, documentation and realisation of Business Processes.

Process-based application development:
Application development with workflow, CASE-, groupware and object technology as well as process-based introduction of standard software.

Process Warehouses:
Operational data management and Evaluation based on business process and integrated solutions with different workflow products.

Knowledge management - Training and Learning:
Production of training aids through graphic models and process-orientated knowledge management.
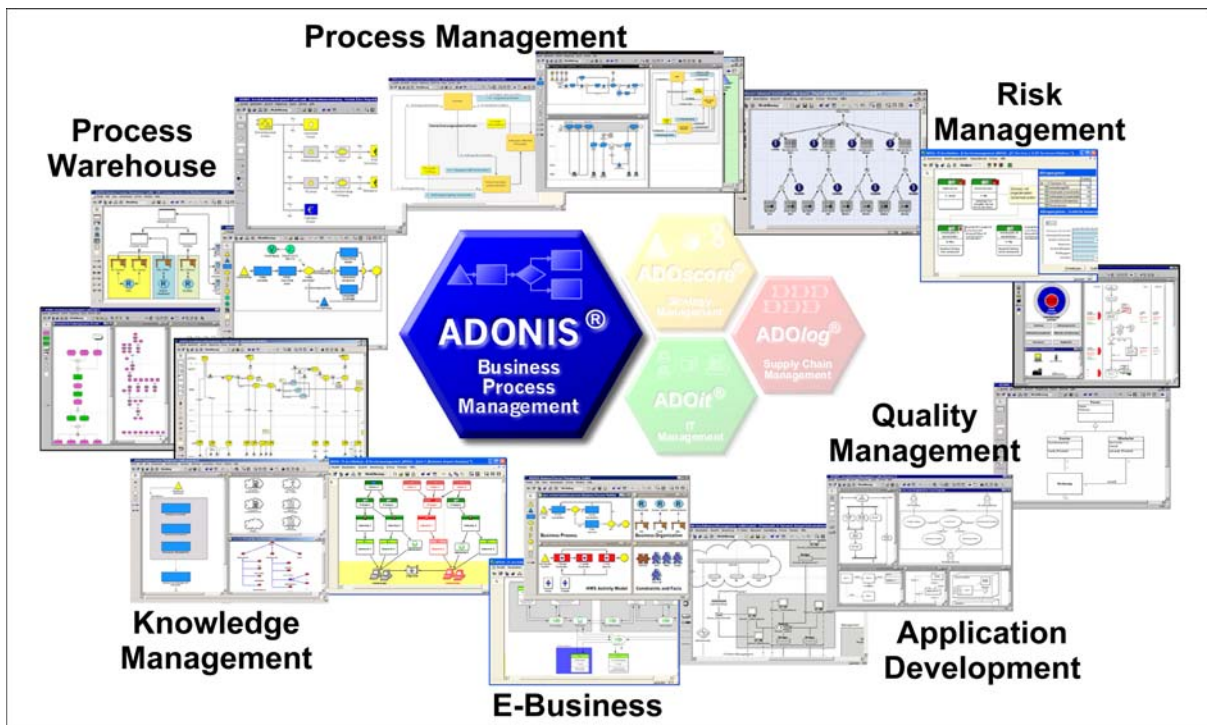
ADONIS is a client/server multi-user system, which has an object-oriented structure. Additionally, **ADONIS** has a remarkable adaptation possibility, so it can be configured according to your needs and developed according to your requirements ("ADONIS-customising").

We hope that our tool meets your requirements and that you have a lot of fun working with ADONIS.

Your BOC Team

Vienna, spring 2006

1. Using the ADONIS Manuals

Three different user groups who use ADONIS can be distinguished:

- **System Administrators**, (in particular **Database Administrators**),
- **ADONIS Administrators** and
- **ADONIS Users**.

System Administrators install ADONIS and create the ADONIS databases. Additionally they can save and restore ADONIS databases. For database administration please refer to ADONIS Database Administration manual.

ADONIS Administrators are responsible for the configuration of ADONIS ("customising"), as well as for the administration of users, libraries, models and components.

ADONIS Users work with the ADONIS Business Process Management Toolkit. They acquire information, model and document their Business Processes and Working Environment (Organisational structures). In addition, the Business Process and Working Environment models constructed can be analysed, simulated, evaluated and if necessary transformed into an operative basis.

If ADONIS is installed as a stand-alone version, one person may carry out all the above roles.

The following ADONIS manuals exist for the different user groups:

- System-Administrators: **"Installation and Database Administration manual"**
- ADONIS-Administrators:
    - **Administration manual**
    - **Method manual**
- ADONIS Users:
    - **Introduction**
    - **User's manual**
    - **Method manual**

2.   About this Manual

The following manual describes the administration of ADONIS version 3.9. Using this manual, it will be possible for you to use the functionality of the ADONIS Administration Toolkit.

It is assumed, that the user of this manual possesses basic knowledge of the use of graphical operating systems and the handling of the mouse.

2.1   Structure of the Manual

The manual is split into several parts. Each part consists of chapters and subchapters, which are numbered using the Arabic system. A link to a (sub)chapter consists of the number of the corresponding chapter and the corresponding page number.

The title of the actual part is quoted in the headlines of each page. The pages of the manual are continuously in Arabic numerals (the table of contents in roman numerals).

Each picture in the manual is also numbered. Within the manual a reference to any picture will include its number and corresponding page number.

2.2   Contents of the Manual

The Administration manual consists of the following sections:

- ADONIS
- General information on the Administration Toolkit
- Administration Toolkit components
- Appendix

General hints as well as an overview of ADONIS and its features can be found in the "ADONIS" section.

The " General Information on the Administration Toolkit" contains general information (regarding components) about the Administration Toolkit like e.g. use elements, ADONIS-Browser, user interface.

The components of the Administration Toolkit (User Management, Library Management, Model Management and Component Management) are described in the "Administration Toolkit" section.

General information on the Administration Toolkit can be found in the "Appendix".

3. New Features in ADONIS Version 3.9

If you have already used ADONIS version 3.81, you will find below a short description:

More detailed information can be found in the PDF file "`News in ADONIS 39.pdf`" which can be found on the ADONIS installation CD in the folder "`books`".

3.1 New Features in the Business Process Management Toolkit

3.1.1 General Innovations

- When using ADONIS 3.9 in Windows XP environments, all graphical elements of the user interface are now shown in the **"XP Look"**.

- ADONIS 3.9 can either be **launched with a German or an English user interface**. An appropriate option can be found in the login dialogue (and as a parameter `-lang` for the command line).

   Note: When using application libraries originating from older ADONIS versions, a library extension is necessary for enabling a multilingual view and unleashing all new possibilities.

- Because of an internal switch to MVC Architecture, **several views at a time of one model** are possible. Furthermore, the model view is now always consistent with the real model contents - it is no longer necessary to refresh the model view manually. The resulting performance gain allows for keeping the model view up-to-date at all times without burdening older computer systems unnecessarily - manual refreshing is no longer necessary. As a consequence, the dialogue "Overview Representation" was removed. All preview windows are now instantly drawn or refreshed.

- With ADONIS 3.9, the traditional "ADONIS Standard Application Library" is replaced with the new **"ADONIS BPMS Modelling Method"**. This library, as its predecessor, is based on the BPMS Paradigm, but contains some new model types and special features.

   Note: Models based on the "ADONIS Standard Application Library" can be transferred trouble-free and effortless to the "ADONIS BPMS Modelling Method" using the import option "Import objects from different library".

- The new tool **"ADONIS Navigator"** shows an overview of the active model in a so-called "docking window" (can be moved across the user interface and docked to special areas of the application window). The part of the model, which is currently visible in the model window, is highlighted an can be used for navigating through the model.

   Note: The Navigator replaces the function "Viewable Area" from the zoom dialogue. At the same time, the accuracy of the preview was improved.

- The new tool **"ADONIS Inspector"** (another "docking window") supplies the user with various details about the active modelling environment.

- **The functional range of the ADONIS Explorer was substanially extended**:

   - Models can be copied via the functions "Copy" (Ctrl+C) and "Paste" (Ctrl+V). This creates a copy of the model, the references pointing to the model remain with the original copy.

   - Model groups can be copied via the functions "Copy" (Ctrl+C) and "Paste" (Ctrl+V).

- Model references can be copied to other model groups via the function "Model reference(s) -> Copy".

- Model references can be removed from model groups via the function "Model reference(s) -> Delete".

- Model groups and models can be moved via the functions "Cut" (Ctrl+X) and "Paste" (Ctrl+V).

- The last model reference of a model can no longer be deleted, to prevent models being moved to the model pool unintentionally. Of course, the model itself can be deleted as ever; this naturally removes the model reference too. (Note: The Pool contains all models not accessible via model groups. They can be moved to the model group structure by the ADONIS administrator.)

- **Editing browser cells in the ADONIS browser is now more comfortable**: For simple attributes (numbers, strings) editing can be simply started by navigating to the cell and starting typing (without preceding double click).

- In the menu "Model", a new menu entry **"Recent models"** provides a list of all models recently opened by the active user.

- Performance has been improved in different areas.

3.1.2   Modelling

- The "Undo" functionality was extended and now offers both **multiple undo and redo** steps. In addition, the scope of undo-able functions was extended; creating objects and connectors and changing the view mode is now also included.

- **Some new functions in the Graphical Model Editor**:

  - The maximum size of the drawing area was increased to 50m x 50m (until now 4,5m x 4,5m).

  - During moving or resizing objects, now the actual state of the object is shown (no longer only a silhouette). This e.g. enables to see during resizing, how the text in a Note will look.

  - The model view was redesigned to prevent jittering during refresh operations.

- **Showing submodels**: An object containing a reference to a different model (e.g. a Subprocess object) can be "expanded". This means that instead of the object, a view of the drawing area of the referenced model is shown.

  - With the "show submodels" function all subordinated models over an arbitrary number of levels can be sown (even recursive!).

  - Showing/hiding submodels is possible both for single objects and for whole models.

  - Submodels can be shown even if they belong to a different model type than the main model.

  - Expanded objects are still accessible (e.g. for opening the notebook).

  - Changes to a model are instantly refreshed in all models having the changed model shown as submodel.

  - Printing and generating graphics from the expanded view is possible.

  - The expanded submodel view can be reset via the context menu or a special hotspot (upper right corner of the expanded view).

- **New features in the Modelling Bar**: Swimlanes are now shown in the Modelling Bar like all other modelling objects. There they can be selected and used as usual. Furthermore, the context menu of the Modelling Bar was substantially extended: View modes can be changed here as well (up to now only via the menu "View") and parts of the contents of the Modelling Bar (e.g. swimlanes or connectors) can now explicitly be hidden without changing the view mode.

- The **search algorithm** was extended and now allows, apart from the classic search for visualised attribute values, **searching all attribute values** (model and object attributes) either in the active or in all loaded models. The search result is shown in a "docking window" (with additional features). On clicking on a search result, the appropriate model is shown and the object containing the searched pattern is highlighted. In addition, all attribute values contained in the search result can be edited directly. The search results remain available as search history. This enables navigating based on different results.

- **Connector representation**: The dialogues "Connector marks" and "Representation of connector edges" from earlier ADONIS versions are now united in one dialogue "Connector representation". Thus, all connector settings are accessible at the same time.

- Now it is possible to **assign Attribute Profile References** in the tabular modelling too.

3.1.3 Import/Export

- **New ADL 3.9 code**: "Class Attributes" of relations (graphical representation, Notebook definition etc.) are no longer exported with the standard ADL export. If this redundant attribute information is needed for some reasons, it is still possible to export in ADL 3.81 code (as it is in ADL 3.0 code).

- The ADL Import can now handle **over-long reference attributes** without problems.

- The XML files generated by the XML Export are now **UTF-8 coded** (Unicode).

3.2 New Functionality in the Administration Toolkit

New features in Administration Toolkit are included in the following components or functions:

- **General news and improvements** (see chap. 3.2.1, p. 7)
- **User Management** (see chap. 3.2.2, p. 7)
- **Library Management** (see chap. 3.2.3, p. 8)
- **Model Managemet** (see chap. 3.2.4, p. 8)
- **LEO, AdoScript and Expressions** (see chap. 3.2.5, p. 9)

3.2.1 General news and improvements

- When using ADONIS 3.9 in Windows XP environments, all graphical elements of the user interface are now shown in the **"XP Look"**.

- ADONIS 3.9 can either be **launched with a German or an English user interface**. The available languages can be found in the file `lang.ini` listed as Windows Language IDs. The first entry is always the default language. This ensures that e.g. in a German version English can be determined as the default language. Via command line a language can be determined using the parameter `-lang` together with either an ISO-639 code or a Windows Language ID.

  Examples: `-langen` for English or `-lang1031` for German.

- Performance has been improved in different areas.

3.2.2 User Management

- Before starting the import of all system user groups and their users of a selected domain, it is now possible to **select certain system user groups** via a new option. Afterwards, only the users of

the selected groups are imported. Especially in large domains with many system user groups this speeds things considerably up.

3.2.3 Library Management

- It is now possible to define **restriction rules for passwords** (e.g. "at least six characters"). This rule is checked when a user changes his password.

- The **definition possibilities for page layouts** were substantially enhanced. Now a lot of new elements, like lines or multiple bitmaps, can be used.

- **Incrementing the change counter** in referencing models after renaming (pool) objects can be disabled.

- Because of the multilingualism feature, **self-made menu entries can be made multilingual** too.

- Set **filters in administration queries** can now be seen via the button text in the settings.

- **The GraphRep features were substantially extended**:

  - In **PEN** elements, pen style (dashed, dotted) and line thickness can now be combined. Furthermore, various styles for line end caps and joints are available.

  - In addition to Bitmaps, the following **graphic file formats** can now be visualised:

    GIF, JPEG, ICO, PNG, TARGA, TIFF, WBMP, XPM.

  - Basic shapes can now be transformed into arbitrary forms using **clipping commands** (CLIP_RECT, CLIP_ROUNDRECT, CLIP_POLY, CLIP_ELLIPSE) and combining operators.

  - **Colour gradients** can now be painted.

  - Using the BEZIER element **Bézier curves** can be drawn.

  - **Loops** (WHILE, FOR) can be programmed.

  - The two **connector marks** of a pair can now be displayed differently - depending on being at the outgoing or the ingoing side of the connector.

  - Direct access to certain **information from Record attributes** now allows for displaying whole tables on the drawing area or showing different attribute content depending on the current context.

  - Displaying different attribute values depending on the user interface language is possible.

- The same attribute can now appear in more than one chapter of a **Notebook**. Furthermore, the Notebook structure can now be made language-dependent: Depending on the chosen user interface language, different attributes can be shown.

- For the **Predefined Analysis/Evaluation Queries**, there now is a new type of input field: the "Relation field", containing all relations of the model type in question.

3.2.4 Model Management

- **New ADL 3.9 code**: "Class Attributes" of relations (graphical representation, Notebook definition etc.) are no longer exported with the standard ADL export. If this redundant attribute information is needed for some reasons, it is still possible to export in ADL 3.81 code (as it is in ADL 3.0 code).

- The ADL Import can now handle **over-long reference attributes** without problems.

3.2.5 LEO, AdoScript and Expressions

- **New Events:** AfterEditAttributeValue, BeforeCreateModel, DiscardRelationInstance.

- **Extended Event:** RenameModelThread.

- The **AdoScript** command REBUILD_DRAWING_AREA is no longer needed, as the drawing area is now always refreshed automatically whenever necessary. Existing commands in old scripts are ignored.

- It is now possible to **get the current user interface language** via AdoScript in order to deduce language-specific reactions.

- Extensions to the **MessagePort "AdoScript":** CREATE_OUTPUT_WIN, OUT, SET_OUTPUT_WIN_SUBTITLE, FILE_EXISTS, SERVICE.

- Extensions to the **MessagePort "Core":** MOVE_INCOMING_INTERREFS.

- Extensions to the **MessagePort "CoreUI":** SET_OBJ_FOREGROUND, SET_OBJ_BACKGROUND, RESET_OBJ_FOREGROUND, RESET_OBJ_BACKGROUND.

- Extensions to the **MessagePort "Application":** SET_MENU_ITEM_HDL, REMOVE_CONTEXT_MENU_ITEM.

- Extensions to the **MessagePort "Modeling":** SET_ATTR_ACCESS_MODE, GEN_GFX_STR, GEN_GFX_FILE, GENERATE_GFX, COMPUTE_REGION_IMAGE_MAP, SET_REPRESENTATION, GET_REPRESENTATION, CLEAR_UNDO_REDO.

4.   Note

<u>Designation of people within this documentation:</u>

We would like to explicitly state here that wherever the third person singular is used within this manual, this is intended to include female as well as male persons. The use of the male form of description is not intended to be discriminatory in any way but is simply used in order to ensure consistent descriptions.

<u>Pictures within this documentation:</u>

The pictures within this documentation were created using ADONIS 3.9 for Windows 2000 and Windows XP. If you are using different operating system, the appearance of some screens may be slightly different.

> **ATTENTION: Further publication of this documentation in any form (including copies) is not permitted without the expressed permission of the BOC Group.**

Part I
ADONIS

Successful and dynamic enterprises can achieve decisive competitive advantage through the ability to adapt their Business Processes quickly to the rapidly changing market conditions and through the active arrangement of their core competencies. The increasing dynamic, globalisation and increasing competition make efficient Business Process Management an essential goal.

To enable this, the Procedural Modelling, Analysis, Simulation and Evaluation of Business Processes is a decisive success factor. The goals of Business Process Management are the optimisation of both the processes of an enterprise as well as the resources and technology which execute those processes.

The ADONIS Business Process Management Toolkit, which was developed by BOC in co-operation with the University of Vienna offers essential tool support for re-engineering and reorganisation of projects.

ADONIS was specifically designed for the particular needs of financial services organisations and provides functionality primarily for the following application areas:

- Business Process Optimisation / Business Process Re-engineering (BPR)

- Quality Management / ISO 9000 Certification and Maintenance

- Controlling (Process Costing)

- Personnel Management (Personnel and Resource planning)

- Organisation Management (Enterprise documentation, Job descriptions etc.)

- Information Management (Creation of technical concepts for IT systems, Interfaces to Workflow and CASE systems, Introduction of ERP Systems)

- Creation of electronic handbooks which can be made available over an intranet with the use of powerful multi media functionality

- Evaluation of Business Processes (Benchmarking, Monitoring, "Should-be" comparison)
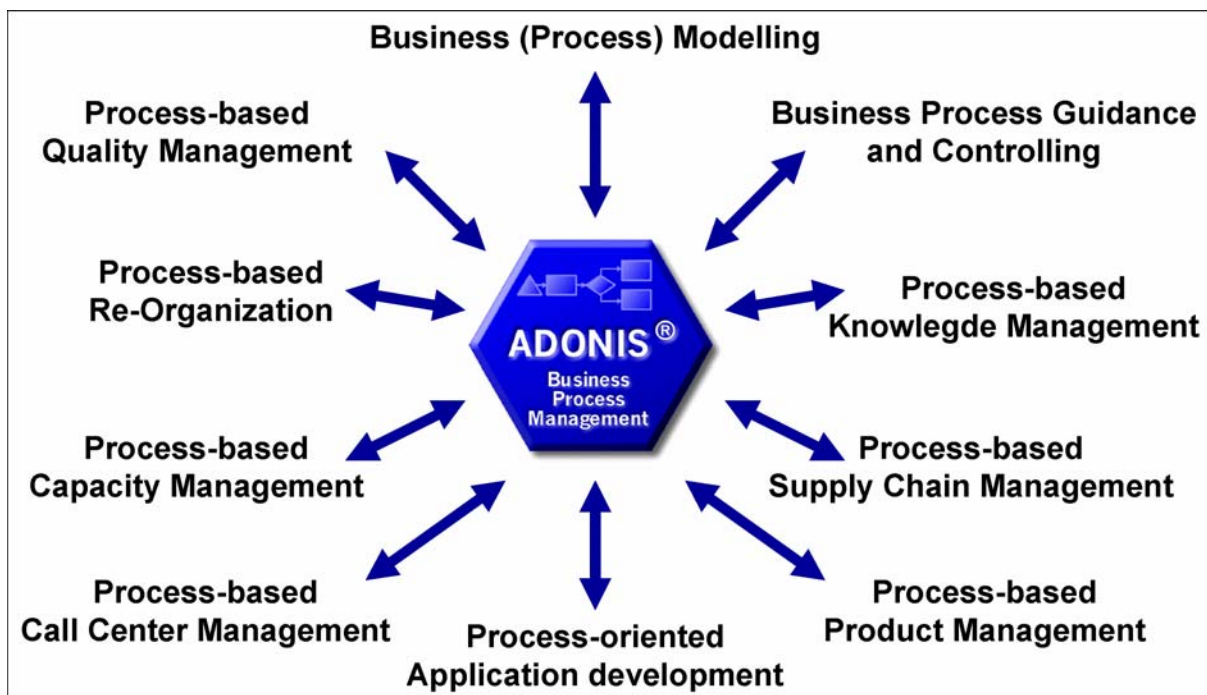
Figure 1: Business Process Management using ADONIS

The "Meta-concept" of ADONIS means that through customising, the tool can be configured to optimally suit the particular requirements of any user. The ADONIS user can decide by himself how he should build the processes and how he can best use the ADONIS mechanisms. As ADONIS is method independent, the management of Business Processes on different levels is guaranteed.

ADONIS® as a tool for business process management and also complies with the requirements for software ergonomics (see chap. 21., p. 694).

1. General Notes

The use of the ADONIS software is primarily subject to the conditions of the licence agreement. Insofar as not agreed otherwise in the licence agreement, the following terms and conditions for usage of the ADONIS software apply:

The customer is informed that all rights relating to the ADONIS software, in particular the copyright and inclusion of all documents and information supplied in the course of contract negotiation as well as any which are currently pending belong to BOC.

BOC transfers to the customer, the non-exclusive and time restricted right to use the ADONIS software in accordance with the conditions of usage specified in the licence agreement.

"Usage" is the total or partial copying (inputting) of the machine-readable software into the specified machine for the processing of the instructions or data contained therein.

The customer may produce a copy of the ADONIS software for backup purposes only. No BOC copyright notice displayed on the software or in the documentation may be changed, amended, replaced or deleted. Reproduction of the documentation is not permitted; additional copies can be obtained from BOC at the current fee.

A separate licence is required for the usage of the ADONIS software on a machine other than the specified machine, except in instances where the specified machine is out of order; in this case the temporary usage of the software on another machine is permitted. The customer agrees to inform BOC without delay of the serial number and location of the specific machine on which ADONIS has been installed, if requested to do so by BOC.

The customer is not permitted to wholly or partially translate back into source code, reverse engineer or decompile the software.

The customer agrees not to make the ADONIS software, including any copies thereof, available to third parties at any time.

ADONIS is a registered trademark of BOC. Copyright BOC Group 1996-2006. All rights reserved.

All product and companies names (see chap. 2., p. 14) are subject to the copyright law of each company. The usage of product and brand names in this manual may not be construed as permitting third parties free usage of such names in breach of copyright legislation

>   **Hint:** The program ADONIS contains some software, which has been developed by the Apache Software Foundation (http://www.apache.org/).

As outlined in the terms and conditions of the licence agreement, BOC guarantees the functionality of the software. In so far as not agreed otherwise in the licence agreement, the guarantee period extends six months from the end of the software purchase period.

The liability of BOC is as outlined in the licence agreement. Insofar as legally permitted, BOC will not accept liability for atypical damage, solely financial damage, loss or damage to recorded data, direct or indirect damage, lost profits, expected but unachieved savings or for damages arising from third party claims against the customer.

2. Trademarks

Company	Trademark
Apache Software Foundation	*Xerces*
BOC Asset Management GmbH	*ADOit*,
	ADOlog,
	ADONIS,
	ADOscore,
	BPMS,
	Management Office
Intel Corporation	*Pentium*
International Business Machines Corporation	*DB2*,
	DB2 Client Application Enablers,
	DB2 Connect,
	DB2 Universal Database,
	FlowMark,
	Informix,
	Lotus Notes,
	LOVEM,
	MQSeries,
	Notes,
	OS/2 Warp,
	Rational Rose,
	WebSphere
Macrovision Corporation	*InstallShield*
Microsoft Corporation	*BizTalk*,
	Excel,
	Microsoft,
	SharePoint,
	Visio,
	Visual Basic,
	Webdings,
	Windows,
	Windows NT,
	Wingdings
microTOOL GmbH	*case/4/0*
	objectiF

Company	**Trademark**
Oracle Corporation	*Oracle*
SAP AG	*R/3 System*
X/Open Company Limited	*UNIX*

3. ADONIS Administration Toolkit

The ADONIS Administration Toolkit supports the effective use of ADONIS. The administration of ADONIS users and user groups, ADONIS libraries as well as ADONIS models and model groups is carried out within the ADONIS Administration Toolkit.

More detailed information on the Administration Toolkit can be found in the Administration manual as well as in the online documentation of the toolkit.

The ADONIS Administration Toolkit consists of the following components:

User Management

The User Management component allows the ADONIS administrator to create and authorise ADONIS users. By assigning ADONIS users to ADONIS user groups, the access rights of the users to model groups and models can be defined. The creation, alteration and deletion of users and user groups is also possible within this component. In addition, it is possible to export ADONIS users and user groups into **UDL files** as well as to **import them from UDL files**. UDL stands for **U**ser **D**efinition **L**anguage. With the help of UDL Import/Export it is possible to transfer ADONIS users and user groups between ADONIS databases. UDL export can also be used as a backup mechanism for your ADONIS users and user groups.

Library Management

In ADONIS every user has access to an ADONIS application library with which he can create models. The ADONIS BPMS Application Library is the default group of objects and relations that support modelling. Within the Library Management component, ADONIS application libraries can be imported, exported and deleted. Library and class attributes can also be altered (customised) within this component.

Editing and checking the library and class attributes of the application libraries takes place within this component. Additionally, application libraries can be exported into ABL files and ABL files can be imported into the ADONIS database. (ABL = **A**DONIS **B**inary **L**anguage). With the help of ABL import and export, application libraries can be transferred to different ADONIS databases. ABL export can also serve as a method of backing up your application libraries.

Model Management

Within the Model Management component, the ADONIS administrator can import, export and delete ADONIS models, model groups and application models. In addition, model groups can be created and altered. The access rights of user groups (and therefore users) are assigned to ADONIS model groups. Model management allows you to set up the access rights to the models of the ADONIS database.

Attribute Profile Management

In the Attribute Profile Management, the ADONIS administrator may add, delete, edit, export, or import the ADONIS attribute profiles and attribute profile groups. Attribute profiles represent the repository

concept in ADONIS. The attribute profile represents one or more attributes, that may be reused for any object and can be centrally maintained. Attribute profiles are embedded in the user rights system, to enable unified and controlled care over the attribute profiles.

Component Management

Using the Component Management, additional components of **ADONIS** can be made available to the ADONIS users. Furthermore, the current configuration of the Business Process Management Toolkit can be checked within this component.

> **Hint:** A more detailed description of the ADONIS Administration Toolkit can be found in the ADONIS Administration manual.

4.   ADONIS Business Process Management Toolkit

You can construct your models using the ADONIS Business process management Toolkit.

The Business Process Management Toolkit provides a large number of components and modules, which enable yout to enrich, analyse, simulate and evaluate your models.

The ADONIS Business Process Management Toolkit consists of the following components:

Information Acquisition

Information Acquisition supports you in gathering information, which is important (or necessary) to successfully model your Business Processes and Working Environments. One of the methods provided is the use of the **acquisition tables or HOMER component**, which runs through a link to Microsoft Excel. Data can be entered in these tables (excel sheets) and this data can then be exported to an ADL file and imported into the ADONIS Business Process Management Toolkit.

Modelling

The Modelling Component is the heart of the ADONIS Business Process Management Toolkit. It allows you to build the models you require (e.g. Business Process or Working Environments). You can create and amend your own models (and the attribute values of the objects used) using the graphical editor (**model editor**) provided. Additionally, it is also possible to input attribute values through a tabular view of the model.

If you have never modelled business processes before, we recommend that you read the section "Fundamentals of Modelling" in the user manual. Some basic tips to ensure the clarity and quality of your models can be found there.

Analysis

Within the Analysis Component, **queries** on your ADONIS models can be run and **relation tables** or **predefined charts** can be produced. Both predefined and user-definable queries are provided in ADONIS. The query language in ADONIS is AQL (**A**DONIS **Q**uery **L**anguage). Creation of queries will be defined by the ADONIS administrator and will be provided to an ADONIS user.

The results of a query can be displayed either as a table or graphically. The results can also be exported to an ASCII file. In this way you can process the results further in another application (e.g. spreadsheet, word processor etc.).

An **Analytical Evaluation** of business process models can also be carried out.

Simulation

The Simulation of Business Processes and Working Environments is executed in the Simulation Component. Four simulation algorithms are available in ADONIS: **Path Analysis** simulates the Business Process models only, while the **Capacity Analysis, Workload Analysis (steady state), and Work-**

load Analysis (fixed time period) simulate the Business Process models *and* the corresponding Working Environment models.

With the help of **ADONIS agents** it is possible to calculate non-standard results during simulation.

 Evaluation

The Evaluation Component offers mechanisms for the Evaluation of "should-be" models as real running processes.

The Evaluation Component provides the following areas of functionality **comparative representation of results**, **Evaluation of real-time audit trails of the workflow management system MQSeries Workflow / WebSphere MQ** (IBM) and **pre-defined evaluation queries**.

As an <u>additional</u> option, the **ADONIS Process Costing** component as well as the **dynamic Evaluation modules** can be integrated with the toolkit.

The ADONIS Process Costing Component supports the optimisation of costs, especially overhead costs and supports the identification and evaluation of possibilities for savings. Dynamic Evaluation Modules, based on the simulation algorithm "Capacity Analysis", enable the period-related evaluation (e.g. Human Resource Management), for which the calculation algorithms can be customised.

 Import/Export

The Import/Export Component provides the option of **exporting** ADONIS models, model groups and application models **into ADL or XML files** as well as **importing** them into the ADONIS database from such files. ADL stands for **A**DONIS **D**efinition **L**anguage. With the help of this component you can transfer ADONIS models, model groups and application models into a different ADONIS database. Additionally, ADL or XML Export can serve as a back-up mechanism for your models, model groups and application models.

The **Documentation** (part of the Import/Export Component) provides you with the possibility to transfer your ADONIS models into an electronic model documentation (e.g. HTML, XML data) or in a paper documentation (e.g. DOC files for Microsoft Word). In this way it is possible to distribute the contents of your models either in document format (through a word processing program) or over an Intranet.

> **Hint:** You will find a detailed description of the ADONIS Business Process Management Toolkit within the section "Business Process Management Toolkit" in the ADONIS user manual.

5.   ADONIS Product Palette

ADONIS version 3.9 is offered as:

●   ADONIS Business Edition (see chap. 5.1, p. 20) or

●   ADONIS Professional Edition (see chap. 5.2, p. 21).

The following **additional components** are not contained in the standard configuration of ADONIS (Business or Professional Edition) and can be acquired additionally and integrated into ADONIS (into any existing modelling method):

●   "Process Cost Analysis" (see chap. 5.3.1, p. 21)

●   "Personnel and Capacity Management" (see chap. 5.3.2, p. 21)

●   "Call Centre Management" (see chap. 5.3.3, p. 22)

●   "case/4/0 interface" (see chap. 5.3.4, p. 22)

To use ADONIS (Business and Professional Edition) a set of different **modelling methods** (see chap. 5.4, p. 23) is available.

By utilising our **interfaces** (see chap. 5.5, p. 23) data from various applications can be exchanged with ADONIS (Business and Professional Edition).

Additionally ADONIS supports further **standards und management methods** (see chap. 5.6, p. 25) for the process management.

>   **Hint:** For further information regarding products and services of the BOC Group please contact your ADONIS consultant.

5.1   ADONIS Business Edition

Within the "**ADONIS Business Edition**" the following components of the ADONIS Business Process Management Toolkit are available:

●   Modelling

●   Analysis

●   Documentation

●   Import/Export

>   **Hint:** The components "Process Cost Analysis (Process Costing Component)", "Personal- and Capacity Management", "Call Centre Management" as well as "case/4/0 interface" are additional components and therefore they are not contained in the ADONIS Business Edition.

When needed it is possible to acquire new components or further modules within a component of the ADONIS Business Process Management Toolkit, which then can be provided for the ADONIS users

by entering a new licence key. Alternatively it is also possible to further restrict the composition of the ADONIS Business Edition.

> **Hint:** When using the ADONIS Business Edition the smart-icons of unsupported components will not be displayed in the component bar. The same applies to smart-icons in the quick-access bars. If the ADONIS user or the ADONIS administrator did not change the configuration, only the smart icons corresponding to the functionality of ADONIS Business Edition will be displayed.

5.2   ADONIS Professional Edition

Within the "**ADONIS Professional Edition**" all components of the ADONIS Business Process Management Toolkit are available.

> **Hint:** The components "Process Cost Analysis (Process Costing Component)", "Personal- and Capacity Management", "Call Centre Management" as well as "case/4/0 interface" are additional components and therefore are not contained in the ADONIS Professional Edition.

5.3   Additional ADONIS Components

The additional components:

- "Process Cost Analysis"
- "Personnel and Capacity Management"
- "Call Centre Management"
- "Case/4/0 interface"
- "ObjectiF interface"

can be integrated into your existing ADONIS installation. They are not contained in the default configuration of ADONIS (Business or Professional Edition).

5.3.1   ADONIS Process Cost Analysis

The **Process Cost Analysis** supports cost optimisation, especially overhead costs. The system is capable to set informative analysis on the available savings possibilities.

The ADONIS Process Cost Analysis simulates the Business Process models. This identifies further potential for analysis and optimisation of the process costs - the course of the control structure can contain decisions and some parallel structures and above this many optional process levels are allowed.

5.3.2   Personnel and Capacity Management

The **Personnel and Capacity Management** supports the time period based evaluation of Business Processes and Working Environments. In particular the periods of validity assigned in the period-related versioning will be used.

The **key figures** determined in the Personnel and Capacity Management **can be corporate-specific configured**. Besides quantitative evaluation **quality evaluation** will also be supported. For instance,

questions such as "What are the roles and skills that I need to realise my Business Processes?" can be answered.

5.3.3   Call Centre Management

The **Call Centre Management Component** supports personnel planning (FTEs) and skills management within Call and Multimedia Service Centres.

The processes in these centres are used as the basis and input for crucial information regarding time, costs and knowledge. For instance:

- How long does a helpdesk query or an address change take?

- What capacity is needed for a given time interval and which costs do arise?

- How do you document for your employees which actions to take in different scenarios?

5.3.4   case/4/0 Interface

Through the coupling of **case/4/0** from **microTOOL GmbH**, **Business Process oriented application development** is possible, to create the required building blocks for a new system in case/4/0 from ADONIS and to maintain consistent links to the specific activities of the Business Processes. The link is established online from special Business Process models to an Application Development environment.

Through this direct integration from specialised Business Process models to IT models before the construction of an application system, it is possible to identify the Business Process **affected** by the new system. The functionality of ADONIS allowing personnel capacity and cycle time to be calculated is very useful for deriving the qualifications necessary to carry out the Business Processes. This means that when changing existing applications, we already have an effective **qualification system**.

The **IT company map** together with the Business Processes provide comprehensive documentation. As a result of the links between the two toolkits, changes to either the applications or the Business Processes can be accurately described and evaluated. In this way modern **Information and Migration Management** is possible.

ADONIS and case/4/0 also offers components for **the generation of documentation** which can then be distributed over the Internet or an Intranet. Through the coupling with case/4/0 Business Process and application documentation can be integrated and distributed to target groups within the enterprise.

5.3.5   objectiF Interface

Through the coupling of **objectiF** from **microTOOL GmbH Business Process oriented application development** is possible, to create the required building blocks for a new system in objectiF from ADONIS and to maintain consistent links to the specific activities of the Business Processes. The link is established online from specialised Business Process models to an Application Development environment.

Through this direct integration from specialised Business Process models to IT models before the realisation of an application system, it is possible to identify the Business Processes **affected** by the new system. The functionality of ADONIS allowing personnel capacity and cycle time to be calculated is very useful for deriving the qualifications necessary to carry out the Business Processes. This means that when changing existing applications, we already have an effective **qualification system**.

The **IT company map** together with the Business Processes provide comprehensive documentation. Because of the links between the two toolkits, changes to either the applications or the Business

Processes can be accurately described and evaluated. In this way modern **Information and Migration Management** is possible.

ADONIS and objectiF also offers components for **the generation of documentation**, which can then be distributed over the Internet or an Intranet. Through the coupling with case/4/0 Business Process and Application Documentation can be integrated and distributed to target groups within the enterprise.

5.4 ADONIS Modelling Methods

The modelling in ADONIS (Business or Professional Edition) is effected applying a defined methodology (= application library). Within one ADONIS database several methods can be administered and provided to the users.

In addition to the possibility to adapt the modelling methodology to the needs of the customer the ADONIS BPMS modelling method (ADONIS BPMS Application Library) is contained within ADONIS (Business or Professional Edition).

The following (de-facto standards) for modelling methodologies are offered in addition:

- UML method

- LOVEM method

- E-Business method

- EPC method

- QM method (ISO9000/2000)

> **Hint:** For further information regarding customer-specific ADONIS modelling methods please contact your ADONIS consultant.

In addition to the modelling methods mentioned above, ADONIS supports further standards and management methods (see chap. 5.6, p. 25) for the process management.

5.5 ADONIS Interfaces

Using the interfaces to/from ADONIS (Business and Professional Edition) data can be

1. imported from other applications into ADONIS,

2. edited in ADONIS and

3. exported from ADONIS into other applications.

For the tools listed below interfaces already have been realised and can be integrated and adapted for your modelling method (application library) if required:

- Data acquisition:

  - Excel (Microsoft Corporation)

  - HOMER (BOC Asset Management GmbH) - contained in the ADONIS Professional Edition

  - Visio (Microsoft Corporation)

- Workflow Management and Enterprise Integration systems:

- BizTalk Server (Microsoft Corporation)
- EasyFlow (Heyde AG/ TOPAS InformationsTechnologien GmbH)
- Fabasoft Components (Fabasoft)
- IBM MQSeries Workflow (IBM)
- IBM WebSpere MQ Workflow (IBM)
- ORACLE BPEL Process Manager (Oracle)
- Staffware 2000 (TIBCO Software Inc.)
- TIBCO Staffware i10 (TIBCO Software Inc.)
- Unisys Workflow (Unisys)
- WPDL (WfMC - Workflow Management Coalition)
- XPDL export

- Simulation:
  - ARENA (Rockwell Software Corporation
- CASE tools:
  - case/4/0 (microTOOL) - interfaces provided directly in ADONIS (see chap. 5.3.4, p. 22)
  - IBM Rational Rose (IBM)
  - objectiF (microTOOL) - interfaces provided directly in ADONIS (see chap. 5.3.5, p. 22)
  - XMI Import/Export
- Software Engineering:
  - SQS-TEST (SQS Software Quality Systems AG)
- ERP systems:
  - R/3 (SAP) - reference models
  - R/3 (SAP) - Online interface to integrate R/3 calls
  - R/3 (SAP) - HR integration
  - JuraPack (PELI) - Claims management
- Groupware systems:
  - IBM Lotus Notes (IBM) - Export interface for company documentation
  - IBM Lotus Notes (IBM) - Online interface to call and integrate Notes documents
  - SharePoint Team Services für Microsoft Project 2002 Server (Microsoft Corporation)
- BPM tools:
  - ibo (ibo)
- Relational databases:
  - via an ODBC interface
- Document management systems (DMS):
  - DOMEA (OpenText Corporation)
  - IBM Lotus Notes (IBM)
  - PCDOCS (Hummingbird)
  - SharePoint Team Services for Microsoft Project 2002 Server (Microsoft Corporation)

- Document management systems supporting ODMA

> **Hint:** For further information regarding customer-specific ADONIS interfaces please contact your ADONIS consultant.

5.6 Support of Standards and Management Methods

ADONIS supports the following standards and management methods.

> **Hint:** Supporting these standards and management methods is performed by extension and adaption of existing modelling methods (ADONIS application libraries). For detailed information please contact your ADONIS consultant.

- Management methods:
  - TQM (Total Quality Management)
  - BSC (Balanced Scorecards)
  - ERM (Enterprise Risk Management)
  - Six Sigma
  - HL7 (Health Level Seven)
  - Basel II
- Business process management/Business modelling:
  - SOX (Sarbanes-Oxley-Act)
  - BPMN (Business Process Modelling Notation)
  - eEPK (Ereignisgesteuerte Prozeßketten)
  - ISO 900x
  - Zachmann Framework
- Enterprise Application Integration (EAI) and Workflow systems:
  - BPEL (Business Process Execution Language)
  - BPMN (Business Process Modelling Notation)
  - WPDL (Workflow Process Definition Language)
  - XPDL (XML Process Definition Language)
- Software development:
  - MDA (Model Driven Architecture)
  - UML2 (Unified Modelling Language)
  - XMI (XML Metadata Interchange)
- Standards for data exchange:
  - XML (Extensible Markup Language)
  - ODBC (Open Database Conectivity)
  - ODMA (Open Document Management API)

Part I

Part II

General Information about the Administration Toolkit

Hint: All examples and descriptions in this part of the user manual refer exclusively to the ADONIS standard application library (see chap. 20., p. 693). Text and numeric entries in the pictures may differ from those on your screen.

1.   Terms and Context

In this chapter you will find an introductory description of the terms used in ADONIS as well as their context. For a better understanding, these terms and context are graphically represented as follows (see fig. 2).
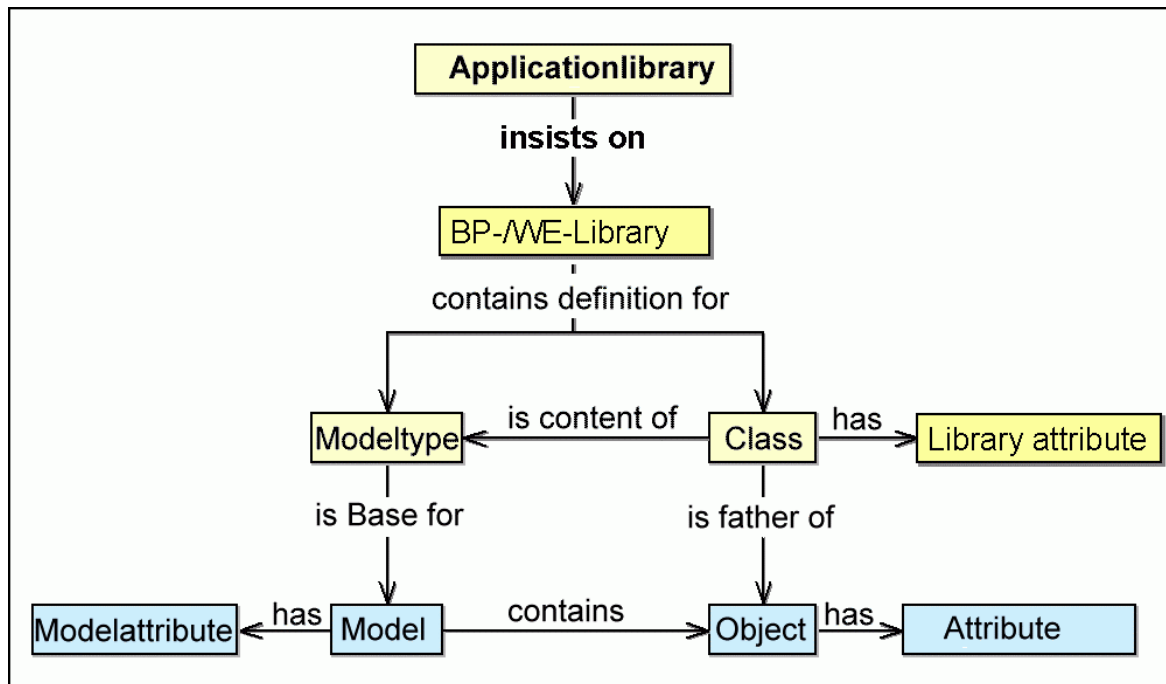
Figure 2: Terms and context in the Administration Toolkit

There is an **application library** for each modelling method in ADONIS. This application library contains all information for a customised implementation of ADONIS.

An application library is made of a **Business Process library (BP library)** containing the information for process models (e.g. Business Process Models) as well as a **Working Environment library (WE library)** containing all information for process models (e.g. organigram).

The BP and WE library have **library attributes** (see chap. 2.1.3, p. 188), which are used in the description of the library for the ADONIS Administrator and in the configuration of Layout, Analysis, Simulation and Evaluation for the ADONIS User (in ADONIS' Business Process Management Toolkit) .

BP and WE library contain the definition for **model types** and **classes** (incl. relation classes).

A **model type** is a grouping of classes. The picture below shows for instance the model types of the ADONIS BPMS Application Library (see fig. 3).
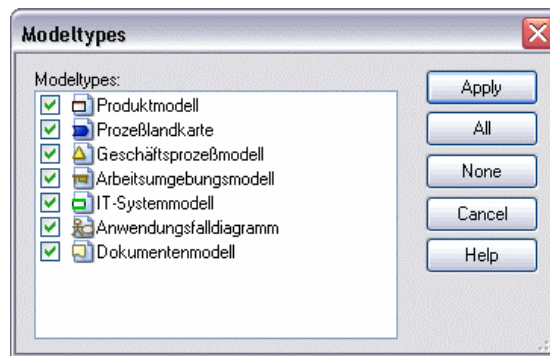
Figure 3: Model types of the ADONIS application library

Classes represent the pattern for the objects created by an ADONIS user (in ADONIS Business Process Management Toolkit). Classes have **class attributes**, which control for instance the graphical display of an object or the arrangement of the object attributes in the ADONIS Notebook in ADONIS Business Process Management Toolkit. In addition, the **(object) attributes** are also defined in classes. Each **(object) attribute** will be assigned to an attribute type (see chap. 5., p. 536) and a standard value during the definition.

Models based on a model type (e.g. Business Process Models) will be created in the ADONIS Business Process Management Toolkit. Models have **model attributes**, which provide general information about the model (e.g. date of creation, status).

A model is made up of **objects**, which are derived (= instanced) from classes. Objects have **(object) attributes**, which contain the information describing the model and its contents.

2. Rights Concept

The permission rights approach in ADONIS enables customised access rights for the ADONIS user to all ADONIS components of the Business Process Management Toolkit (see fig. 4).
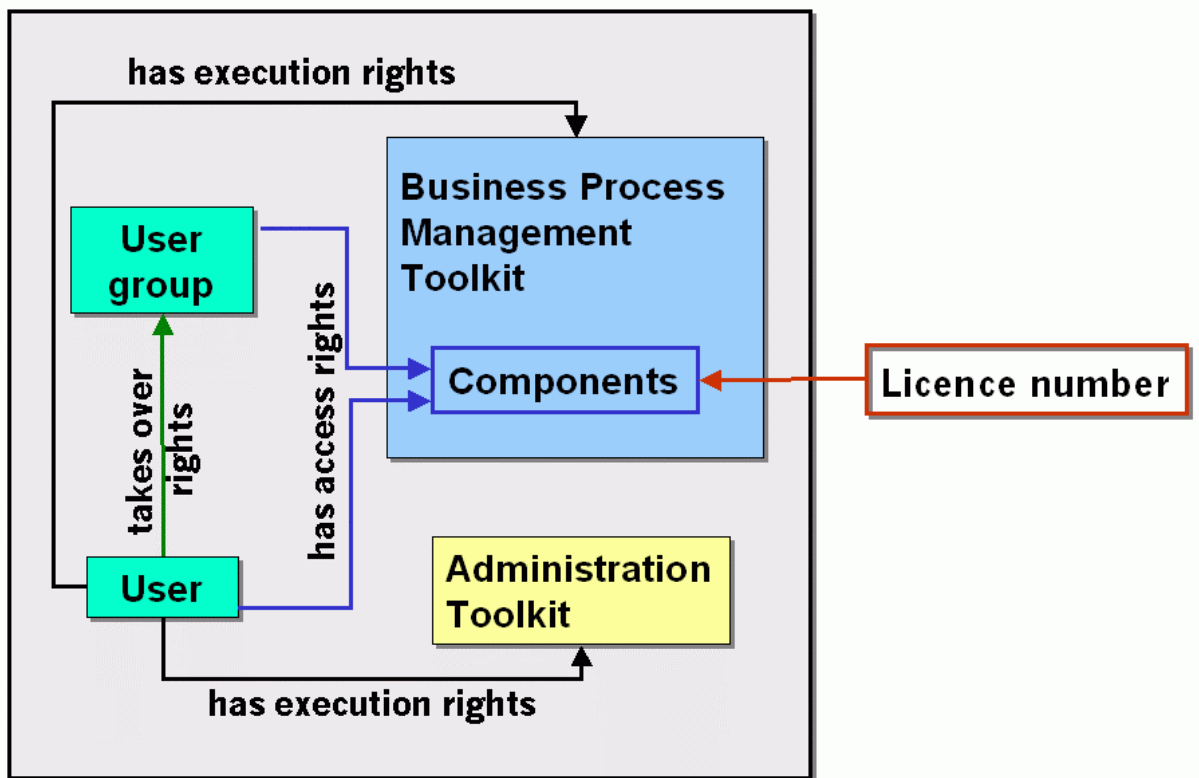
Figure 4: ADONIS-Rights conception

The ADONIS user can be allowed **execution rights** for the ADONIS Business Process Management Toolkit and/or the ADONIS Administration Toolkit, by means of which he can open the toolkit.

In addition, the **access rights** for the components, which are available by the licence number, can also be configured for the ADONIS user and ADONIS user groups. An ADONIS user takes over the access rights of the user group to which he is assigned.

The permission rights for the access to **components** can be gradually limited to remove the unused components (see fig. 5).
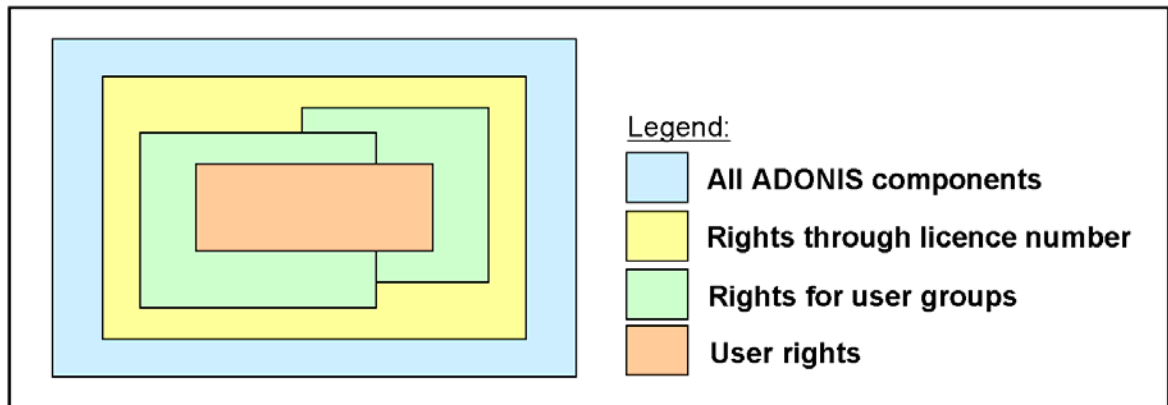
Figure 5: access to components

For all ADONIS components of Business Process Management Toolkit (incl. the additional components of ADONIS), the limitation of rights is possible through:

1. **the licence number during the ADONIS installation** (= customer specific configuration of ADONIS),

2. **the component access for user groups in the Administration Toolkit** (= user group specific access to a subset of the available components due to the customer specific configuration of ADONIS),

3. **the component access for users in the Administration Toolkit** (= user specific access to a subset of the available components due to the user group rights).

    Note: An ADONIS user has access to the maximum of the components available in his user group, i.e. if the access is allowed by a user group and not allowed by another user group, then the user who is assigned to both of these user groups still has the access rights.

For user groups, it is also possible to set access rights to model groups (see chap. 3.3.1.6, p. 298) and to define or view limitations for notebooks (see chap. 2.1.1.2, p. 171).

3. ADONIS User Interface

The ADONIS user interface follows the structure of other applications. However, there are some additional features.

The ADONIS window/user interface (see fig. 6) appears on the screen after the ADONIS Administration Toolkit (see chap. 1.1, p. 528) has been started. This window consists of the following five parts:

- Window bar (see chap. 3.1, p. 32)

- Menu bar (see chap. 3.2, p. 33)

- Component bar (see chap. 3.3, p. 33)

- Quick-access bar (see chap. 3.4, p. 33)

- Workspace (see chap. 3.5, p. 33)
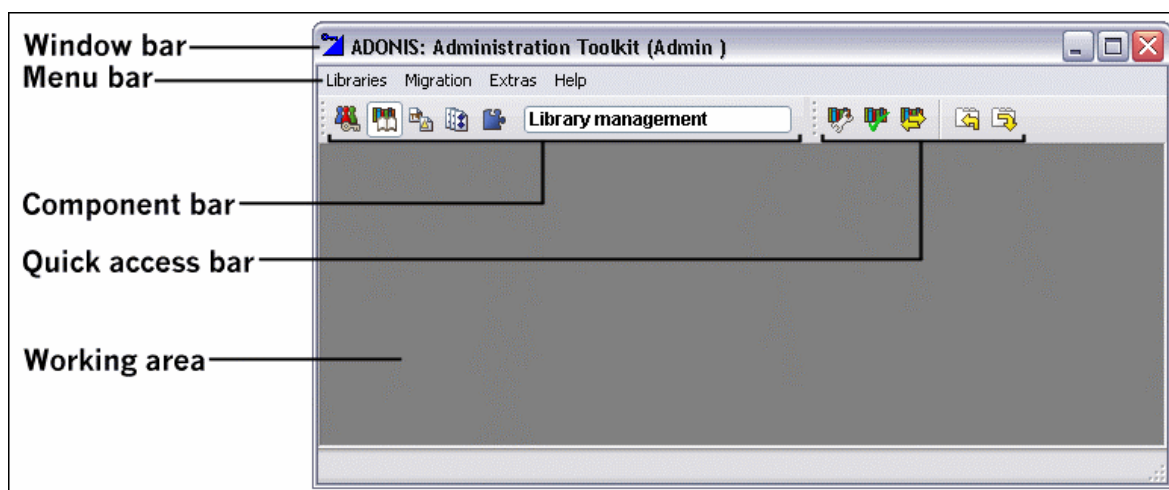
Figure 6: Working area in the Administration Toolkit

3.1 Window Bar

The window bar (header bar) consists of the exit button (small program symbol), information on the working area (see chap. 3., p. 32) (ADONIS component and the user name in brackets) and the buttons "Minimise", "Maximise" and "Close" (see fig. 7).
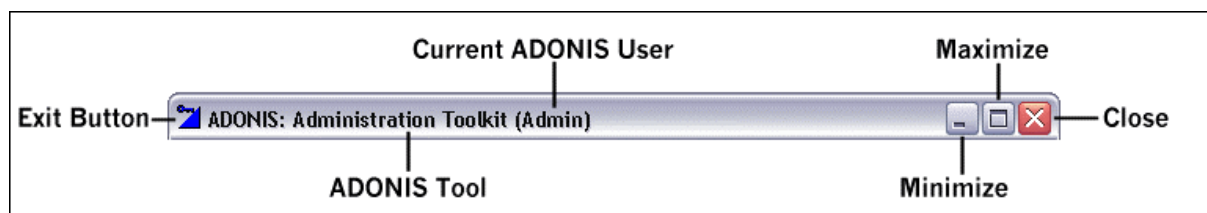
Figure 7: Window bar in the Administration Toolkit

3.2   Menu Bar

The menu bar is placed below the window bar (see chap. 3.1, p. 32).

On activating a component (e.g. by clicking on a smart icon in the component bar (see chap. 3.3, p. 33)), the menus of the respective component are displayed (see fig. 8).

Figure 8: Menu bar (Library Management)

3.3   Component Bar

You can find the component bar below the menu bar (see chap. 3.2, p. 33) at the left hand side.

Figure 9: Component bar (Library Management)

The component bar (see fig. 9) contains the symbols (smart icons) for the single components of the Administration Toolkit (see chap. 3., p. 16) (the components "User Management", "Library Management", "Model Management", "Attribute Profile Management", and "Component Management").

Alternatively you can also activate a component by clicking on the component bar (to the right of the component icons) with the right mouse button. By selection of the required component a popup menu appears. Alternatively, you can open this popup-menu by pressing the function key <F9> and activate a component by using its accelerator (=underlined letter in the component name).

The name of the component currently active is displayed to the right of the component symbols.

3.4   Quick-Access Bar

The quick-access bar is located below the menu bar (see chap. 3.2, p. 33) right next to the component bar (see chap. 3.3, p. 33)

Figure 10: Quick-access bar (Library Management)

The quick-access bar (see fig. 9) offers symbols for actions frequently used when working with this component.

3.5   Workspace

The workspace is the part of the window directly below the component and quick-access bar. (see chap. 3.3, p. 33). It is used especially when working within the modelling component of the Business Process Management Toolkit.

4. Control Elements

Control Elements (buttons, accelerator keys) are found in a number of places in ADONIS in order to provide support.

Control Elements are available:

- **in selection windows** (see chap. 4.1, p. 34) and
- **in ADONIS Notebooks** (see chap. 4.2, p. 39).

4.1 Selection Control Elements

The global selection control elements are available in the context men (right mouse button) of every selection window (e.g. User List, Application Library ADL Export).

The following general functions are available in most windows as an addition to the context-specific functions:

- "Modeltypes" (see chap. 4.1.1, p. 34)

  (only available in model selection lists)
- Refresh (see chap. 4.1.2, p. 35)
- Item search (see chap. 4.1.3, p. 35)
- Save as (see chap. 4.1.4, p. 36)
- "Shrink/Expand" (see chap. 4.1.5, p. 37) with the following options
    - "Show all"
    - "Hide all"
    - "Show selected"
    - "Show selected with sublevels" and
    - "Shrink selected"

  (only available in hierarchical selection lists)
- "Select all items" (see chap. 4.1.6, p. 38)
- "Deselect" (see chap. 4.1.7, p. 38)
- "Selected items" (see chap. 4.1.8, p. 38)
- "Shrink/Expand time related versioning" (see chap. 4.1.9, p. 39)

  (This function is only available in model and attribute profile selection lists if time related versioning is being used.)

4.1.1 Model Types

A list of all model types defined in the application library is displayed in the window "Modeltypes" (see fig. 11).

Hint: If your application library contains a customised model type filter, this filter will appear instead.
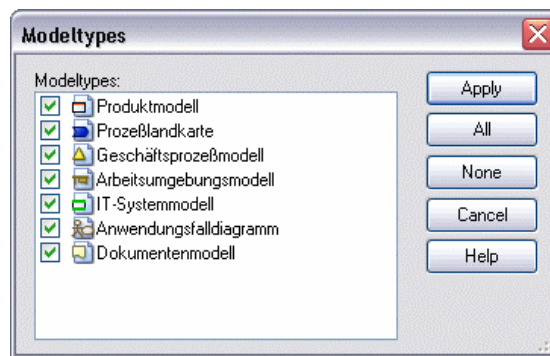
Figure 11: Model types of the ADONIS BPMS Application Library

To the left of each model type, a checkbox indicates, whether the model type is currently displayed (checked ☑) or not (unchecked ☐). Click into the checkbox to change the status.

If you click **"All"**, all model types are activated. By clicking on **"None"**, all model types will be deactivated.

After the successful change, click **"Assign"** to display the updated model selection list, i.e. only the models of the activated model types are displayed.

Hint: By default, all model types are activated.

Hint: The "model types" menu item is only available in model selection lists.

ATTENTION: Changed settings will be reset when exiting ADONIS.

4.1.2 Refresh

By selecting "Refresh" (Icon 〰 or key **<F5>**), the selection list will be updated with the current status of the ADONIS database.

4.1.3 Item Search

By selecting "Item search" (shortkey **<Ctrl>+F**) you can search for entries in the list. When this button has been selected, the window "Entry search" is displayed (see fig. 12).
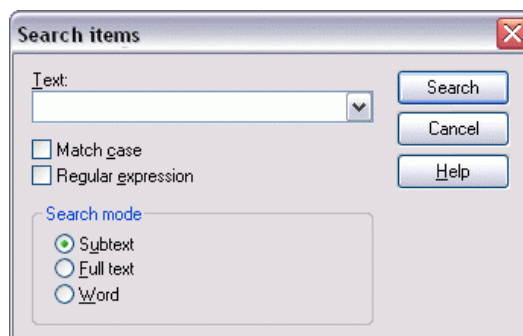
Figure 12: Entry search

Enter the text for which you wish to search into the field "Text" (or select an item already in the list).

The option "Match case" determines whether or not the actual case of the text for which you are searching should be taken into account.

The option "Regular expression" allows a search for regular expressions (see chap. 4., p. 534).

Additionally you can specify the type of search you are carrying out:

- **"Subtext"**: means that the text being searched for can be part of a larger word or sentence.

- **"Full text"**: searches for an entire piece of text, which matches the text for which you are searching.

- **"Word"**: assumes that the piece of text you are searching for is an entire word and will only locate matching words. A word must be a single grouping of text, flanked by blanks, a full-stop or the like.

Start the search by clicking "Search". If entries matching the search criteria are found in the list, the search results will be displayed (see chap. 4.1.3.1, p. 36).

4.1.3.1   Display Search Results

All entries matching the search criteria ae displayed in the window "Found entries "<Search criteria>"" within their hierarchy (see fig. 13).
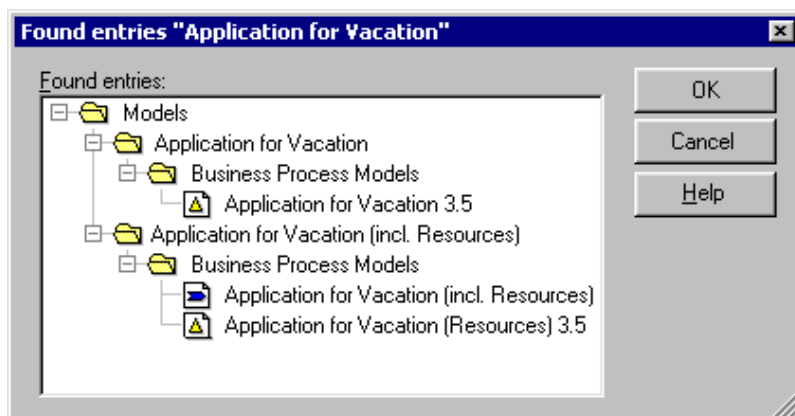
Figure 13: Search results

Select any result in the "Found entries" and then click "OK". The search results window is closed and the selected entries copied into the original selection list. Therefore you can carry out the action (e.g. open model, export models) on the selected search result.

4.1.4   Save as

By selecting "Save as" you can save the contents of the list to a text file. Once this menu item has been selected the "Save as" window is displayed (see fig. 14).
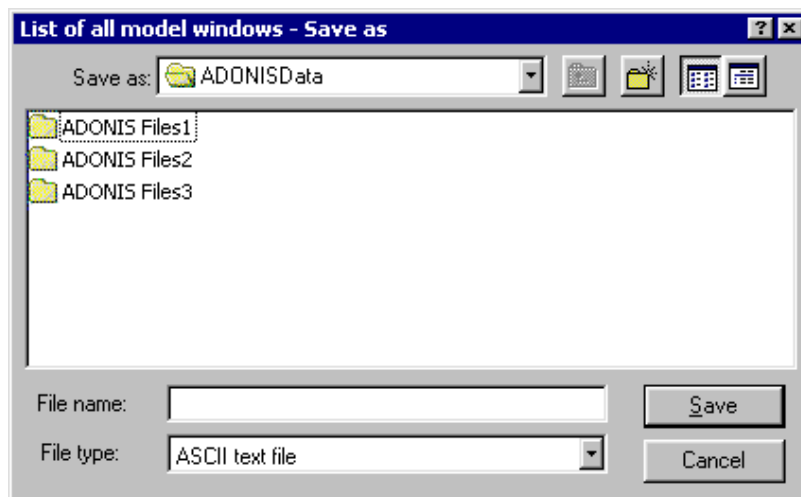
Figure 14: Save as

Enter a file name and select a path where the contents of the list are saved.

4.1.5 Shrink/Expand

Hint: "Expand/Shrink" functions are available only in lists with hierarchical levels

The menu entry "Expand/Shrink" leads to a sub menu, where the hierarchical view in the selection window can be expanded or condensed, the following options are available:

- "Show all"
- "Hide all"
- "Show selected"
- "Show selected with sublevels"
- "Shrink selected"

The options in a nutshell:

Show all:
All hierarchical levels of the list are displayed.

Hot key: **<Ctrl>+Enter**

Hide all:
Only the topmost hierarchical levels of the list are shown.

Hot key: **<Ctrl>+<Shift>+Enter**

Show selected:
This option has the same effect as clicking on the symbol 🔲 and is only available when at least one list entry is selected. Entries that are immediately below the next hierarchy level will remain collapsed.

Hot key: **+**

Show selected with sublevels:
This function is also only available if at least one entry is selected. All entries on all hierarchical levels below this one will be selected - regardless of structure depth.

Hot key: ***

Shrink selected:

> This option has the same effect as clicking on the symbol ⊟ and is only available when, at least one list entry is selected. All entries below the selected entry are collapsed.
>
> Hot key: **-**

4.1.6   Select All Items

By selecting the "Select all items" menu item or using the hot key **<Crtl>+<A>** all selectable items in the list are selected.

> **Hint:** Only the selectable items of the selection list will be selected, i.e. the items shrunk are not selected. To select all the items of a selection list, first select the "expand all" menu item (see chap. 4.1.5, p. 37) and then the menu item "Select all items".

4.1.7   Deselect

With "Deselect", all selected list entries are deselected.

4.1.8   Selected Items

By selecting the "Selected items" menu item all items selected in the list will be displayed in the window "All selected items" (see fig. 15).
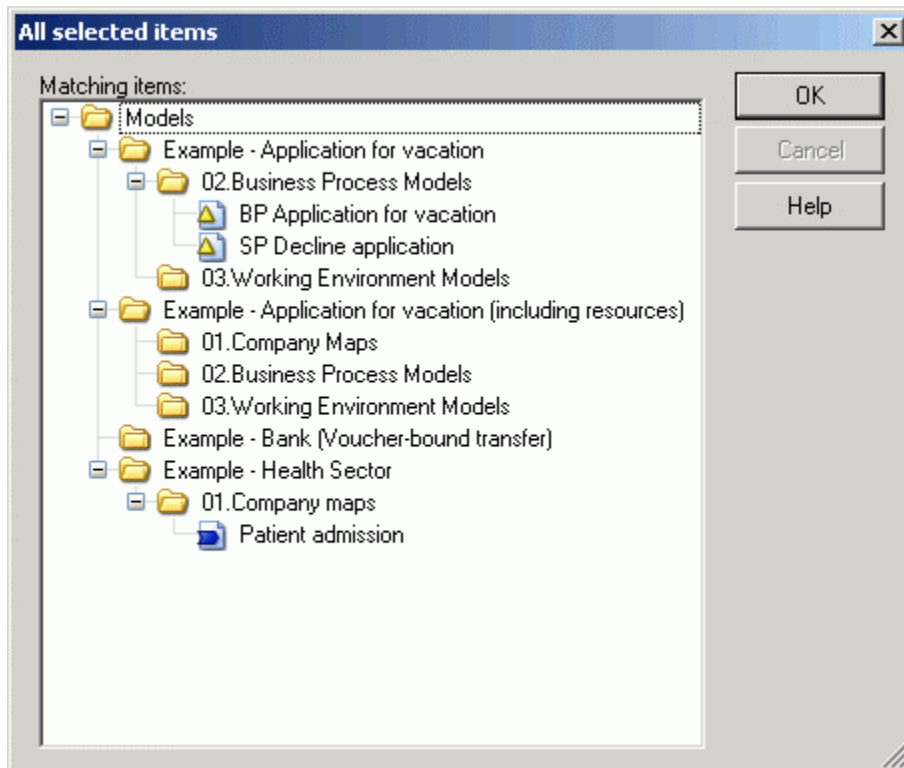
Figure 15: Display selected items

4.1.9   Shrink/Expand Version Threads

Hint: The "shrink/expand version threads" function is only available if a **time-related versioning** (see chap. 6.1.2, p. 69) is defined in the application library.

Expanding version threads is possible by clicking on the icon (right above the model/attribute profile selection list) and causes the displaying of all the versions of a model/attribute profile saved in the ADONIS database (see fig. 16).
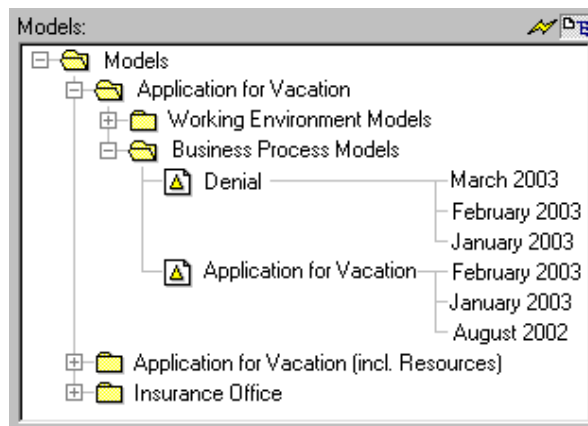
Figure 16: Model selection list with expanded version threads

In case of the hidden version thread, only the latest version of a model/attribute profile will be displayed in the selection list (see fig. 17).
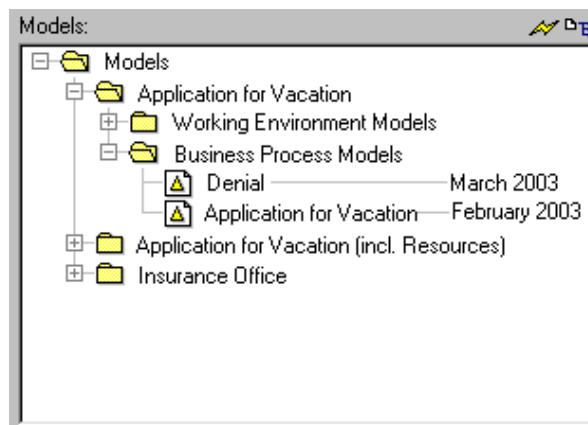
Figure 17: Model selection list with shrunk version threads

4.2   ADONIS Notebook Control Elements

The ADONIS Notebooks within the Administration Toolkit are used to configure the attributes of a library.

Each ADONIS Notebook consists of one or more chapters, which in turn contain one or more pages. Tabs on the right hand side of the Notebook indicate the chapters. If a chapter contains more than one page, then the pages are numbered. By clicking on the tabs, the first page of the selected chapter is displayed. The arrow buttons and in the bottom right hand corner enable you to flip forwards and backwards through each page of the ADONIS Notebook.

The ADONIS Notebooks contain a number of icons, which carry out specific functions and assist in the input of various attribute values. There are also accelerators (underlined letters) in the attribute and chapter names - these in combination with the Alt-key provide another method for navigating through the notebooks.

The ADONIS Notebooks contain the following functions:

☐   "Large text field" icon (see chap. 4.2.1, p. 40)

☐   "Dialog" icon (see chap. 4.2.2, p. 40)

A   Accelerators (see chap. 4.2.4, p. 42)

The "Save" and "Print" icons can be found in the top right hand corner of the ADONIS Notebook.

The "Info" icon can be found on the top right hand corner of the ADONIS Notebook to provide information on the specific class and can also be found over a number of attributes within the Notebook to provide information on the particular attribute.

The "Large text field" icon can be found on the right hand side above attribute fields of this type.

4.2.1   "LargeText Field" Icon

When you click on this icon ☐, a large input field is displayed for the particular attribute. In this way an overview of the input for long attribute values can be taken. The "Large text field" icon is available for all multi-line attributes.

It is also possible to print out the contents of these fields.

4.2.2   "Dialog" Icon

When you click on this icon ☐, a support dialogue for the particular attribute value is displayed. It supports the entering of complex attributes (e.g. performer assignment, statistical distributions for the generation of variables, graphical representation of objects and connectors etc.).

> **Hint:** The "Dialogue" icon is only available in the ADONIS Notebooks of objects and connectors in the Business Process Management Toolkit as well as in the Class Attribute "GraphRep" within Library Configuration in the Administration Toolkit.

4.2.3   Input Window

4.2.3.1   Input Dialogue for Text

In the input dialogue for text (see fig. 18), texts (e.g. attribute values) are displayed in a field changeable in size and further information and functions are available.
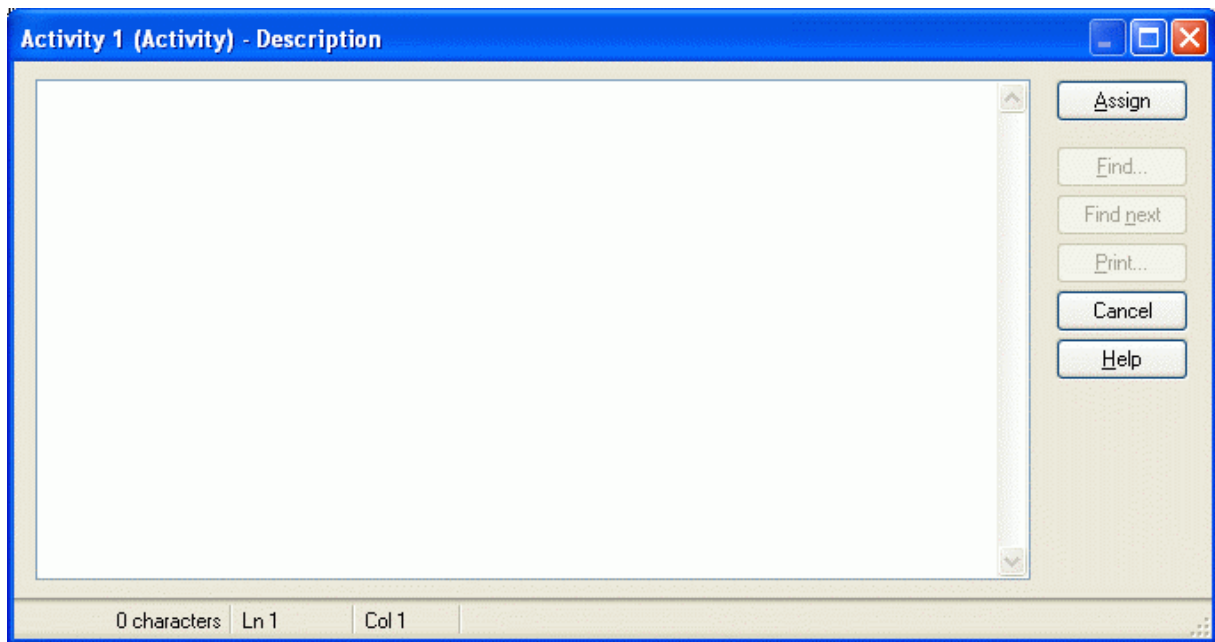
Figure 18: Input dialogue for text attributes

The first field of the status row (at the bottom of the window) shows the number of characters contained in the text. This supports you in entering attribute values, since for example attribute of type "text" (STRING) must not contain more than 3700 characters. In the second field of the status row, the position of the cursor is displayed (row, column).

Once you have edited the text, click **"Assign"** to confirm your entries.

> **Hint:** The "Assign" button is only available if the text displayed can be edited.

By clicking on **"Find"** you can search for entries in the text (see chap. 4.1.3, p. 35). According to the definition of a search text, further entries will be found in the text, if you click **"Find next"**.

You can print (see p. 41) the contents of this input field by clicking on **"Print"**.

Print Text Field / Diagram

Before printing the text, the window "Print text field" is displayed (see fig. 19).
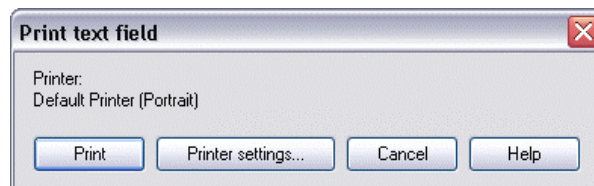
Figure 19: Print text field

Click on the **"Print"** button to send the text to the indicated printer.

Click on **"Printer settings"** to change the settings for the printout.

> **Hint:** If you pront a diagram instead of a text field, the window bears the title "Print diagram". The functionality stays the same.

4.2.3.2 Text Input Field

In the text input field (see fig. 20) a single-line field text input is available.
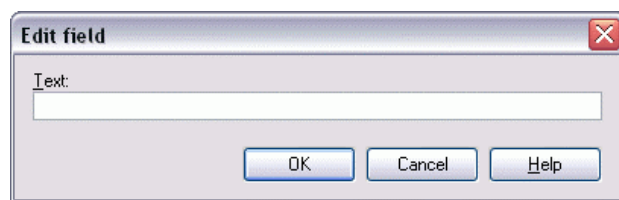
Figure 20: Text input field

Hint: The label of the text input field (window title and field label) as well as some standard text entries are defined in the application library.

4.2.3.3 Input Dialogue for Expressions

A special input dialogue for changeable attributes of the type "Expression" (see chap. 5.4, p. 537) (see fig. 21) is available.

Hint: The changeability status of expression attributes are defined in the library.
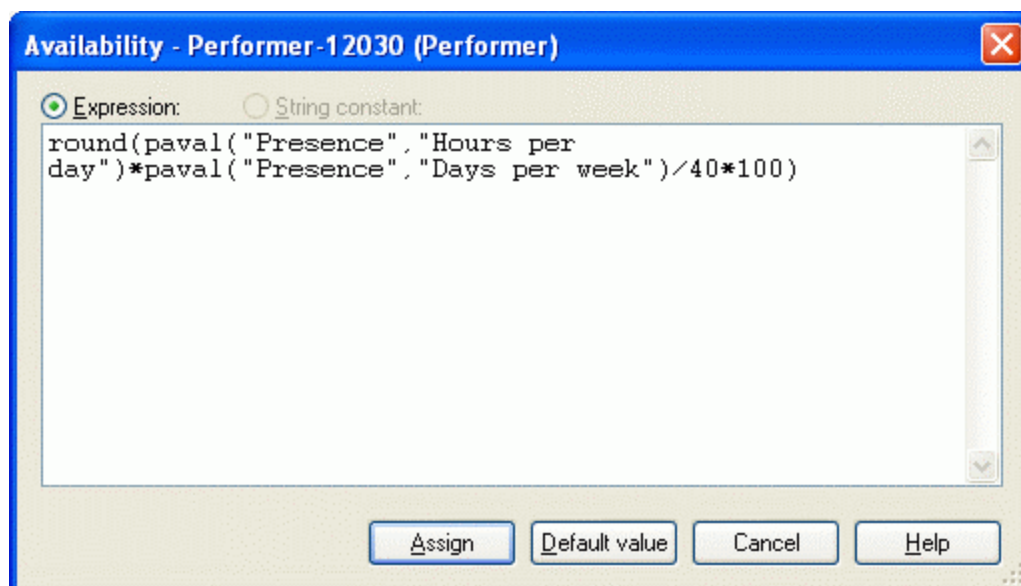
Figure 21: Input dialogue for expressions

You can enter either an expression (see chap. 10., p. 561) or a string. Click on the **"Standard value"** button to enter the standard value defined in the application library.

Click **"Assign"** to confirm your entry.

Hint: The entered value must not contain more than 3600 characters.

4.2.4 Accelerators

The accelerators in the ADONIS Notebooks support you in navigating and entering attribute values.

The accelerators are identified by underlined letters and are contained in all chapter names and attribute names (see fig. 22).
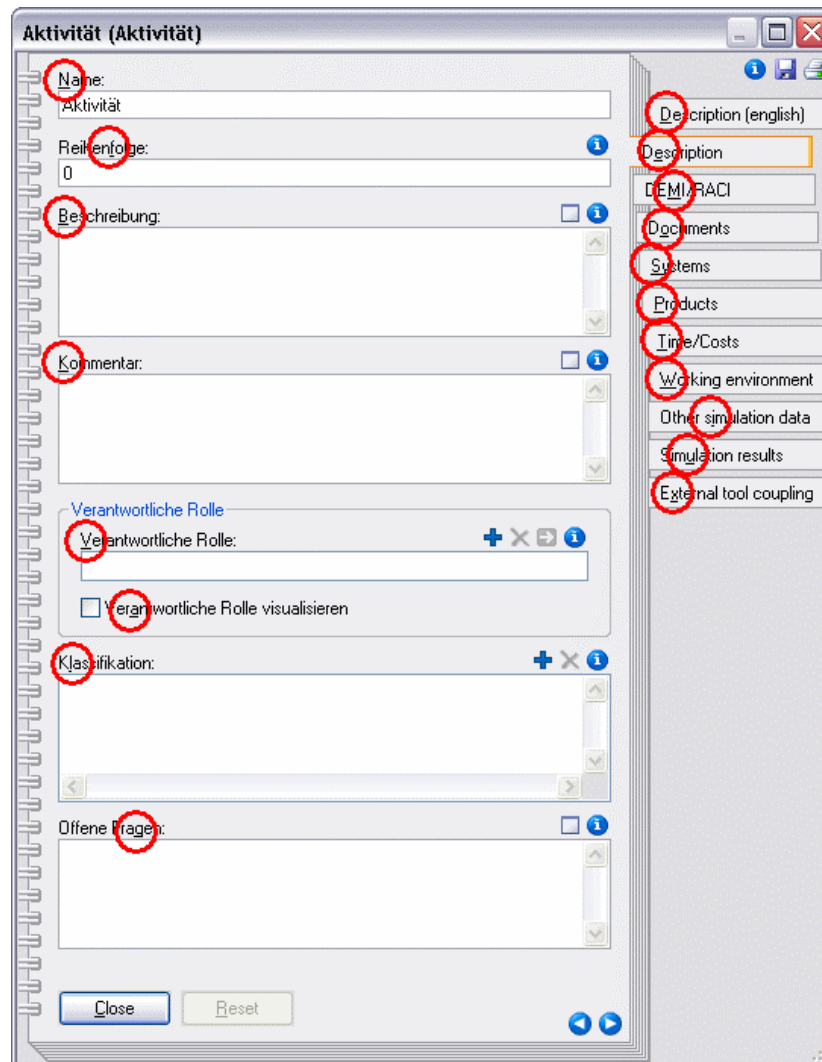
Figure 22: Accelerators in a ADONIS Notebook

Navigation using accelerators is carried out using a hot key combination of **<Alt>** and the desired underlined letter. On entering the accelerator of a chapter, the first page of the chapter is displayed. By entering the accelerator of an attribute name, the cursor is placed in the relevant attribute field so that you can input a value or edit the existing value.

For turning the pages of the Notebook, you can use either the arrow buttons in the bottom right corner of the Notebook or the key combination **<Alt>+<+>** for the next page and **<Alt>+<->** for the previous page.

> **Hint:** In Windows environments it is possible to hide accelerators by default. However, as soon as you press the **<Alt>** key they become visible.

5. ADONIS Browser

A browser included in ADONIS provides an overview, work can be distributed and further work (printing, saving...) can be carried out. Regardless of where you are in ADONIS, the ADONIS browser provides a standard layout.

The ADONIS browser is used in the Library administration (see chap. 2., p. 125) and the Attribute profile administration (see chap. 4., p. 335).

5.1 Classification

In ADONIS different types of browsers are available. They can be divided into:

- **hierarchical** and
- **non hierarchical** browsers

and depending whether the information can be changed into

- **editable** and
- **non editable** browsers.

The information in an **editable browser** (see chap. 5.2.1, p. 48) can be edited and saved while the information in a non editable browser will only be used to show results.

The information in a **hierarchical browser** (see chap. 5.2.2, p. 48) is shown in different levels (e.g. models and sub models) while information in a non hierarchical browser is shown in a one-dimensional structure.

Both groups of browsers (hierarchical and editable) can be combined in the actual ADONIS browser depending on the information shown.

Examples:

- editable/hierarchical:

Figure 23: Example of an editable, hierarchical browser (analysis result)

- editable/non hierarchical:

Figure 24: Example of an editable, non hierarchical browser (tabular model representation)

- non editable/hierarchical:

Figure 25: Example of a non editable, hierarchical browser (simulation result capacity analysis)

- non editable/non hierarchical:

Figure 26: Example of a non editable, non hierarchical browser (simulation result path analysis)

5.2 Structure

The window of the ADONIS browser displays data (results) in tables and offers the functionality to save, print and display the data graphically.
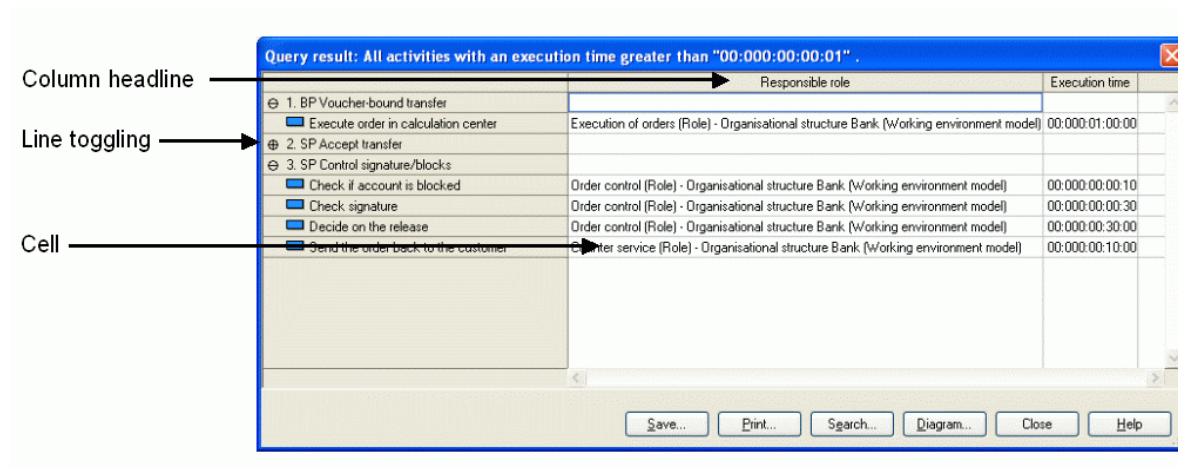
The column headings (first row, grey background) show the type of data in each column (e.g. "attribute"). The first **column** (grey) contains the structure of the rows or the row names for each of the data sections (lines).

The **rows** group the results in individual data sections. Each data section contains one or more cells.

Within the **cells** the corresponding attribute values or calculation results are displayed.

> **Hint:** In hierarchical ADONIS browsers (see chap. 5.2.2, p. 48) empty cells will also be shown.

The picture below (see fig. 27) shows the results of a Capacity Analysis in the ADONIS browser. Some information about the results is also shown.

Figure 27: ADONIS browser structure

In a **hierarchical ADONIS browser** (see chap. 5.2.2, p. 48) a number indicating the level will be shown in the (grey) header of each row indicating the hierarchy level of the current row.

The display area (white background) contains the results.

The structure of the browser in terms of the rows and columns labels and contents varies according to the way results are displayed.

In an **editable ADONIS browser** (see chap. 5.2.1, p. 48) you can change the values displayed in the cells. The availability of this function depends on the one hand on the data displayed and on the other hand on the write access to models from which the data is taken from. For example, you can edit the results of an Analysis query directly in the ADONIS browser, if you have previously opened the models for the query with write access.

> **Hint:** The active (current) cell or the selected area will be indicated by a frame using the colour for the frame as defined in the system settings.

> **Hint:** Write protected attributes in editable ADONIS browsers will be shown in grey font.

The **context menu** (right mouse button) in the ADONIS browser contains among other things the following functions:

- **Expand all (see chap. 5.7, p. 54)**: show the entire contents of the browser by expanding all areas,

- **Shrink all (see chap. 5.7, p. 54)**: only shows the highest level,

- **Save (see chap. 5.13, p. 59)**: save the contents of the browser in a file,

- **Print (see chap. 5.15, p. 62)**: print the contents of the browser as currently displayed (WYSIWYG),

- **Find (see chap. 5.12, p. 58)**: search for specific text within the browser,

- **Copy to clipboard (see chap. 5.14, p. 62)**: copy the contents of the browser to the clipboard,

- **Select attributes/columns (see chap. 5.8, p. 55)**: select the columns, which have to be shown (attributes),

- **Sort (see chap. 5.9.1, p. 56)**: sort the browser contents,

- **Column width (see chap. 5.3, p. 52)**: enter the column width,

- **Adjust column width (see chap. 5.4, p. 53)**: adjust the width of the columns to the optimum size,

- **Row height (see chap. 5.5, p. 53)**: enter the row height.

- **Adjust row height (see chap. 5.6, p. 54)**: adjust the height of the rows to the optimum size.

> **Hint:** The functions "Expand all (see chap. 5.7, p. 54)" and "Shrink all (see chap. 5.7, p. 54)" are only available in the hierarchical ADONIS browser (see chap. 5.2.2, p. 48) .

The **context menu** (right mouse click) **for the column header** in general contains the following functions:

- **align attribute values** (see chap. 5.10, p. 57): align the attribute value of the current column to the left, centre it or to the right,

- **sort ascending/descending** (see chap. 5.10, p. 57): ascending or descending sorting of the attribute values of the current column.

5.2.1 Editable ADONIS Browser

The editing of the displayed value in an editable ADONIS browser is possible by:

● directly entering a value in any cell,

● copying and inserting values inside the browser,

● inserting values which have been copied onto the clipboard, for instance from a table calculation program.

For simple attribute types (figures, text) simply enter an attribute value **directly into the appropriate cell**, by clicking on the cell or selecting the cell using the cursor keys. Type in the new content and confirm with the "Enter" key.

Complex attributes (enumerations, expressions etc.) need an input support dialogue. In this case, double-click the cell or select it and press the "Enter" key to call the support dialogue.

If you want to **edit several (related) values of the same attribute** (i.e. in the same column), hold down the <Shift> key, click on the cell with the first (highest) value (the mouse pointer changes to

) and mark - with another mouse click - the domain of the value to edit. Press the <Enter> key to start the input. If you press the <Enter> key again, you will finish the entry of the current value and start the entry of the next value.

> **Hint:** You can abort the editing of several values at any time by pressing the <Esc> key.

Copy an attribute value, by clicking the cell with the value to copy or select it using the cursor keys and then select the "Copy" menu item in the context menu (right mouse button) or press the key combination <Ctrl>+<C>.

Paste the copied attribute value, by clicking on the cell in which the value should be inserted or select it using the cursor keys and then select the "Paste" menu item in the context menu (right mouse button) or press the key combination <Ctrl>+<V>

> **Hint:** The copy and paste function is also possible for several values (areas) at the same time, by selecting the appropriate areas with the mouse and then carrying out the functions "Copy" and "Paste".

If you want to **copy the value of a cell into several (related) cells** of the same attribute (i.e. in the same column), press and hold the <Ctrl> key, then click on the cell with the value to copy (the mouse

pointer will change to   ) and select the target area - by pressing the mouse button again for each cell while still holding the <Ctrl> key- .

Additionally, it is also possible to insert cell values **from another application** (e.g. Microsoft Excel) into ADONIS browser .

> **Hint:** When copying and pasting values, note that the type of attributes must match with each other, i.e. the type of attributes of the copied value must correspond to the type of attribute in which you want to insert the value.
>
> **Exception:** In attributes of the type "String" or "Longstring", you can insert the values of other types of attributes.

5.2.2 Hierarchical ADONIS Browser

The hierarchical ADONIS browser enables the representation of hierarchical structures (tree structures), making it possible to shrink and expand each tree (see fig. 28). Furthermore a number indicat-

ing the level will be shown in the (grey) header of each row indicating the hierarchy level of the current row.

Figure 28: Hierarchical ADONIS browser (Example simulation result)

Before each section either ⊕ or ⊖ is displayed in the first column. This icon ⊕ indicates that the current row contains a summary of results and can be broken down further by clicking on the symbol. This symbol ⊖ means that all information is displayed. By clicking on this symbol only summary information is displayed.

> **Hint:** Lines without a symbol represent the lowest hierarchy (tree) and therefore have no other subordinated data.

> **Hint:** Due to the hierarchical representation the ADONIS browser will also show empty cells.

5.2.3 Representation of Attribute Values

The attribute values shown in a ADONIS browser are shown as text whereas the representation of the following attribute types differs from the normal representation in the ADONIS Notebook:

Records A record in the tabular modelling is represented by the word "`[Record]`" and in the analysis results represented by the symbol ⊕

Program call The representation of a program call attribute (see chap. 5.2.3.1, p. 50) depends on the definition of your application library.

Reference A reference to a **model** is represented by "<*model name*> (<*model type*>)" whereas multiple references within one attribute are shown using several lines.

A reference to an **object** within a model is represented by "<*object name*> (<*class name*>) - <*model name*> (<*model type*>)" whereas multiple references within one attribute are shown using several lines.

Enumeration list The enumeration values selected from the list will be displayed separated by ";".

Calendar A process or performer calendar is represented by the word "`[Calendar]`".

To **edit the displayed attribute values** click on the respective cell. Depending on the attribute type the following input support dialogues are available:

Edit record attributes (see chap. 4.1.8.1, p. 343)
　　　To edit attributes of type "Record" (RECORD).

Colour definition (see chap. 4.1.8.2, p. 344)
　　　To select a colour (for the graphical representation of an object).

Choose enumeration value (see chap. 4.1.8.3, p. 344)
　　　To choose an attribute value (in the tabular representation).

Selection enumeration value from list (see chap. 4.1.8.4, p. 345)
　　　To select an attribute value from an enumeration list.

Edit external program call (see chap. 4.1.8.5, p. 346)
　　　To edit an external program call (in the tabular representation).

Edit date (see chap. 4.1.8.6, p. 346)
　　　To edit attributes of type "Date" (DATE).

Edit date/time (see chap. 4.1.8.6, p. 346)
　　　To edit attributes of type "Date and time" (DATETIME).

Edit time (see chap. 4.1.8.8, p. 347)
　　　To edit attributes of type "Time" (TIME).

Add references (see chap. 4.1.8.9, p. 348)
　　　To create model and object references of the type "Reference" (INTERREF).

Performer assignment (see chap. 4.1.8.12, p. 353)
　　　To define performer assignments.

Resource assignment (see chap. 4.1.8.13, p. 357)
　　　To define resource assignments.

Definition of a performer calendar (see chap. 4.1.8.14, p. 359)
　　　To define a performer calendar and therefore when the performer will be present.

Definition of a process calendar (see chap. 4.1.8.15, p. 365)
　　　To define a process calendar.

　Hint: For referencing an attribute profile (see chap. 4.1, p. 337) move the mouse pointer to the respective cell, open the context menu (right mouse button) and select the menu entry "attribute profile".

5.2.3.1　　　Representation of Programcall Attribute Values

The attribute values of program calls can be represented differently depending on the definition in your application library. The possible values are as follows:

● "**-> [*Executable*]**"

　Instead of **Executable** the name of the defined program will be shown. The program is started by double clicking on the cell.

　Note: The name of the program and the parameter are defined by the ADONIS administrator and cannot be changed.

● "**-> [<automatically>]**"

　Double clicking on the cell will start the program associated with the parameter.

　Note: The parameter is defined by the ADONIS administrator and cannot be changed.

● "*Executable*"

　Double clicking on the cell will show the input dialogue for the program call (see chap. 4.1.8.5, p. 346) while not displaying the parameter.

Note: The parameter is defined by the ADONIS administrator and cannot be changed.

● "*Parameter*"

Double click on the cell to show the input dialogue for the program call (see chap. 4.1.8.5, p. 346).

● "[*Parameter*]"

Double clicking on the cell will show the input dialogue for the program call (see chap. 4.1.8.5, p. 346) whereas the executable program has been predefined.

Note: The executable program is defined by the ADONIS administrator and cannot be changed.

5.2.3.2 Edit References

By clicking on a cell with references a window showing the reference targets is displayed.

References to models in the list "Reference targets" (see fig. 29) will show the following information:

● Reference status

● Model type icon of the referenced models

● Name of the referenced models
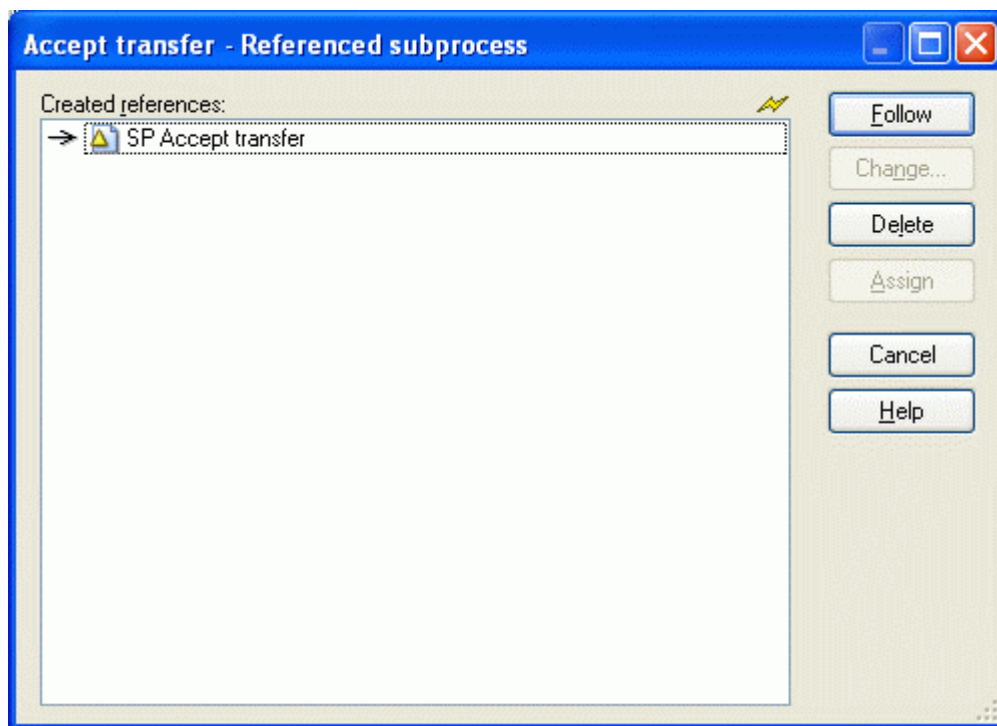
Figure 29: Editing a model reference attribute

References to objects in the list "Created references" (see fig. 30) will show the following information:

● Reference status

● Class symbol of the referenced object

● Name of the referenced object

● Model type icon of the model containing the referenced object

- Name of the model containing the referenced object
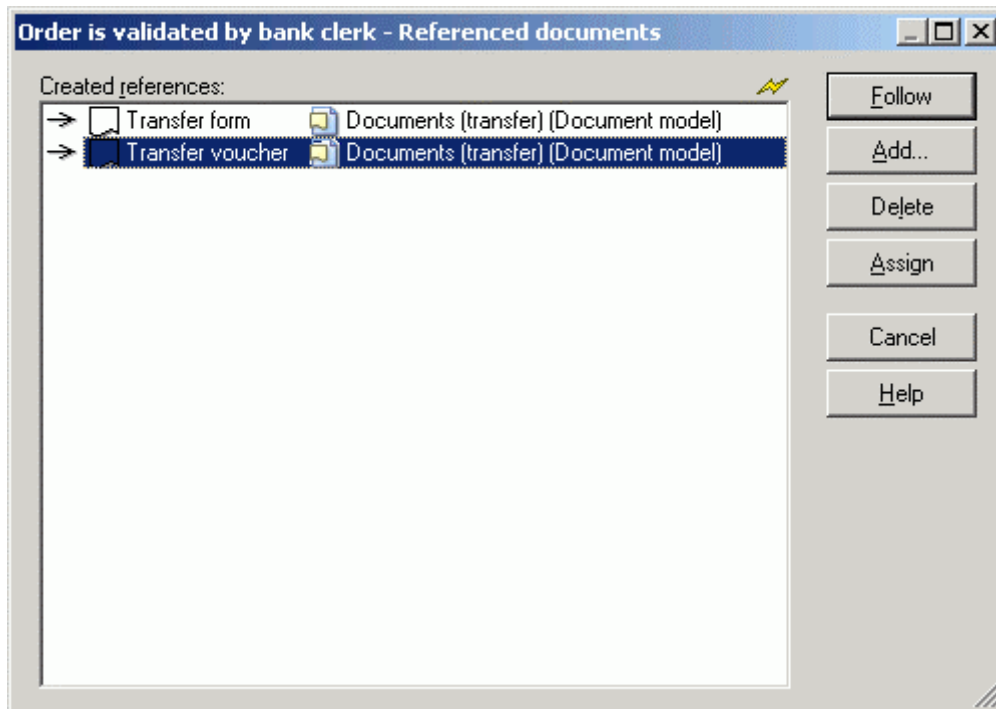
Figure 30: Editing an object reference attribute

The status of the reference at the beginning of each line indicates whether the reference is valid (→) or broken (↯).

By clicking on the button:

"Follow"	you can follow the previously selected reference and display the referenced model or object;
	Note: Following references is only possible in the Business Process Management Toolkit.
"Change"	enables you to edit (see chap. 4.1.8.9, p. 348) the reference;
"Add"	helps you to define (see chap. 4.1.8.9, p. 348) new references;
	Note: Adding references is only possible in the Business Process Management Toolkit.
"Delete"	removes the previous references;
"Assign"	confirms the changes performed.

5.3 Enter Column Width

The ADONIS browser enables the user to define the width of any column. To enter a column width, place the mouse pointer on the appropriate column, open the context menu (right mouse button) and select the "Column width" menu item.

In the "Set column width" (see fig. 31), the current column width is shown in the "width" field.
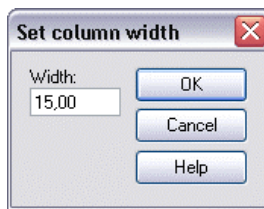
Figure 31: Set column width

Enter the value of the column width and then click on the OK button.

>   **Hint:** Every value between 0 and 200 is allowed for the column width.

>   **Hint:** The column width will be given in so-called norm signs, i.e. the entered value defines the number of characters, which will be displayed in the standard font in the cell.

5.4   Adjust Column Width

Within the ADONIS browser it is possible to dynamically adjust the width of any row. In order to do this, place the mouse pointer at the division point between two columns (see fig. 32) in the header column.
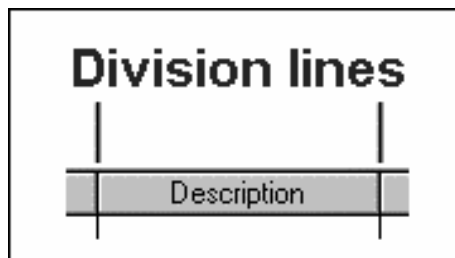
Figure 32: Column separator

The mouse pointer will change to ◀‖▶ when it is placed in the correct place. Once the mouse pointer changes, it is possible to increase or decrease the width of a column by pressing and holding the left mouse button and moving the mouse in the required direction. In this way it is possible to display the results in columns wider than the standard maximum.

It is also possible to adjust the width of a particular column by double-clicking in the header of that column (this will adjust it to the optimal width).

>   **Hint:** By selecting "Adjust columns" in the context menu (right mouse button) all the columns in the ADONIS browser are adjusted to their optimal width (wide enough to display the entire contents of the largest value in the column to a maximum of 80 characters).

5.5   Enter Row Height

The ADONIS browser enables the user to define the height of any rows. To enter a row height, place the mouse pointer on the appropriate row, open the context menu (right mouse button) and select the "Row height" menu item.

In the "Set row height" window (see fig. 33) , the current number of visible rows is shown in the "Row" field.
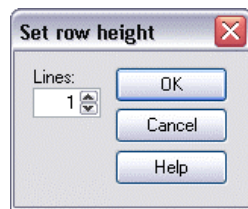
Figure 33: Set row height

Enter the number of rows you want to display and then click on the OK button.

5.6 Adjust Row Height

With the ADONIS browser it is possible to dynamically adjust the width of any column, so that cells that contain several lines can be shown completely.

To adjust a row height, place the mouse pointer on the division line between two rows (see fig. 34) in the first column.
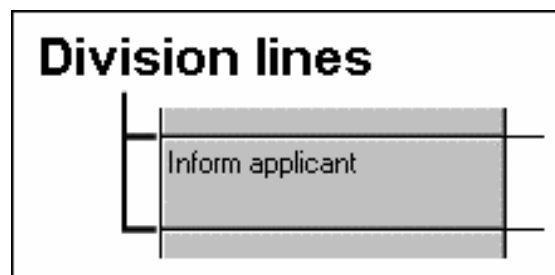
Figure 34: Row separator

The mouse pointer will change to ⇕ when it is placed in the correct place. Once the mouse pointer changes, it is possible to change the height of a row by pressing and holding the mouse button and moving the mouse in the required direction. In this way it is possible to display results containing line breaks.

> **Hint:** By selecting the **"Adjust row height"** menu item in the context menu (right mouse button), the row will be adjusted to it's optimal height.

> **Hint:** By selecting the **"Adjust all rows height"** menu item in the context menu (right mouse button), all rows will be adjusted to their optimal height.

5.7 Expand / Shrink All

The ADONIS browser stores the output of queries, Capacity and Workload Analysis in groups (e.g. models). In order to present an overview to the user and to allow the user to find particular results quickly. These groups can then be expanded (or shrunk) to display more detail.

Unexpanded rows are characterised by the symbol ⊕. By clicking on this symbol, the rows will be expanded and this symbol ⊖ will be shown.

To restore to the original state click on the symbol again and it will change from ⊖ to ⊕.

> **Hint:** By selecting the menu item **"Expand all"** in the context menu (right mouse button) all rows of the ADONIS browser will be displayed.

By selecting the menu item **"Shrink all"** in the context menu (right mouse button) only the highest-level row will be displayed.

5.8   Select Attributes/Columns

With the ADONIS browser it is possible for specific displays to shrink or expand any columns with attribute values or evaluation results

> **Hint:** The function for the selection of attributes/columns to display is not available in all ADONIS browsers.

> **Hint:** If you want to sort the browser contents according to a specific attribute, this attribute must be expanded before you call the "Sort" function (see chap. 5.9.1, p. 56).

Select attributes

To select the attributes to display (i.e. columns with attribute values), open the context menu of the browser (right mouse button) and select the "Select attributes" menu item. A window will be displayed (see fig. 35), in which all the attributes that can be displayed are shown following the notebook structure.
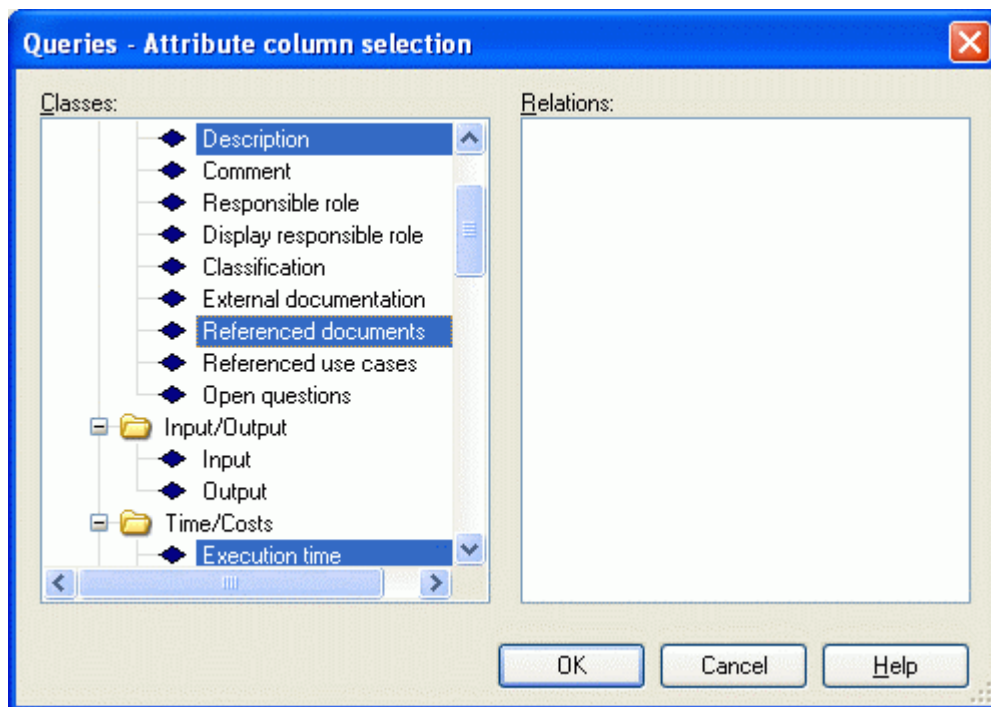
Figure 35: Select attributes (Example from the "Model search")

> **Hint:** In the browser, the editable attributes are marked with the ◆ symbol, and the write-protected attributes with the ◇ symbol.

Select the attributes that the sort should be applied to and then click on the OK button.

Select columns

To select the columns to display, open the context menu of the browser (right mouse button) and select the "Select columns" menu item. All columns, that can be displayed in the browser are shown in the "Select visible columns" window (see fig. 36).
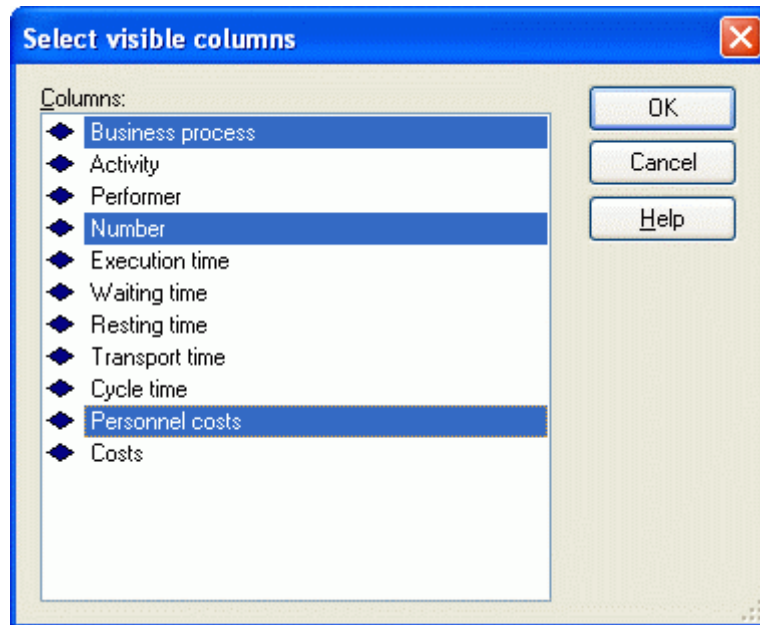
Figure 36: Select columns (Example from the "Analytical evaluation")

Select the columns you want to display and click on the OK button.

5.9   Sort

In the ADONIS browser, it is possible for specific displays to sort the browser contents using the attribute values displayed (within a column).

   Hint: The function to sort is not available in every ADONIS browser.

To sort the browser contents displayed move the mouse to the top of the column and select the menu item "Sort (ascending)" or "Sort (descending)" from the context menu of the browser (right mouse button).

In addition it is possible to select the attribute to use (= column) for the sort using a dialogue for selection (see chap. 5.9.1, p. 56). In this case open the context menu of the browser (right mouse button, the mouse pointer must not be on the header of a column) and choose the menu point "Sort".

5.9.1   Sort by Attribute Columns

After selecting the menu point "Sort" within the context menu of the browser, all attribute columns within the respective notebook structure are displayed (see fig. 37).
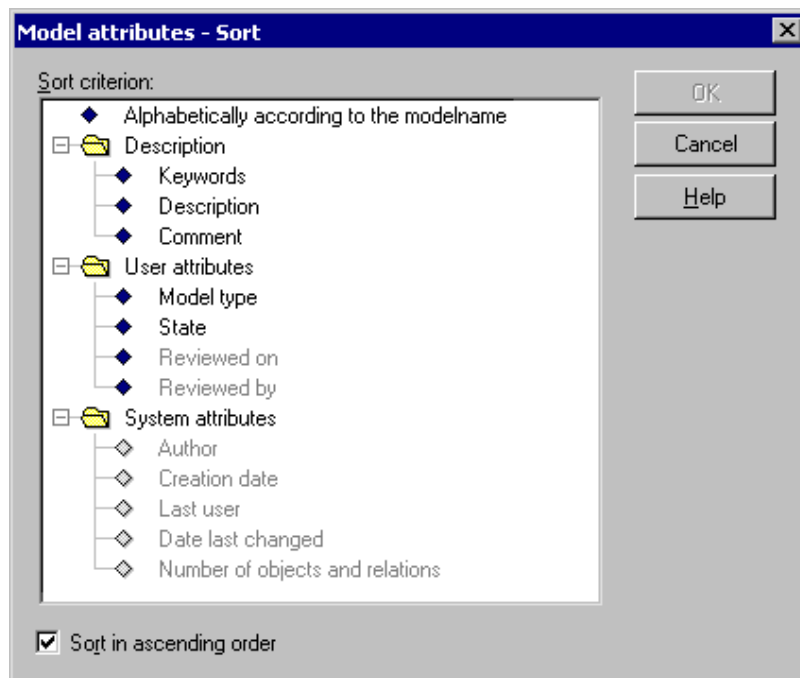
Figure 37: Attribute selection to sort (Example from tabular model display)

> **Hint:** It is only possible to sort with columns being displayed in the browser. Columns/attributes not shown in the browser will be shown in the selection dialogue in grey font. If you want to sort using a column currently not displayed you will have to include the column beforehand (see chap. 5.8, p. 55).

> **Hint:** In editable browsers, changeable attributes are marked with the ◆ symbol, and write protected attributes with the ◇ symbol.

Select the attribute, according to which the sort should be ordered by and then click on the OK button.

5.10   Align Attribute Values

The ADONIS browser offers the possibility to align the displayed attribute values of a column

- left,
- centred or
- right.

Align the values of a column by moving the mouse to the top of the column, open the context menu (right mouse click) and select the alignment ("Left", "centred" or "Right").

> **Hint:** The current setting is indicated by a hook in the context menu.

> **Hint:** The default setting for text values is left, while numbers will be shown right justified.

5.11   Show/Edit (Attribute) Values

For improved readability, for editing or for showing additional information, you can display (attribute) values in a separate window.

> **Hint:** The possibilities for displaying (attribute) values in a separate window depends on the type of the attribute and on the type of the browsers (see chap. 5.1, p. 44).

If you want to show a (attribute) value, double-click on the cell with the (attribute) value to display and the following window will be opened (see fig. 38).
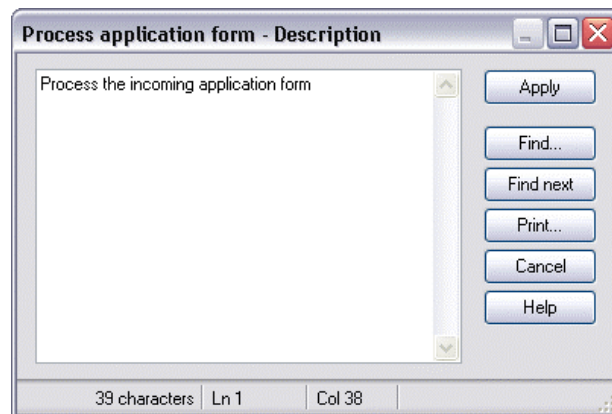
Figure 38: Show (attribute) values

You can save the displayed (attribute) value to a file ("Save" button) or print it out (button "press").

> **Hint:** (Attribute) values can be changed in **editable browsers** (see chap. 5.2.1, p. 48) only.

5.12   Search

Using the button "Search" (or the corresponding item in the popup menu) you can search for a specific piece of text in the browser (in all expanded rows). The window "Search for browser contents" (see fig. 39) is displayed after clicking on this button.
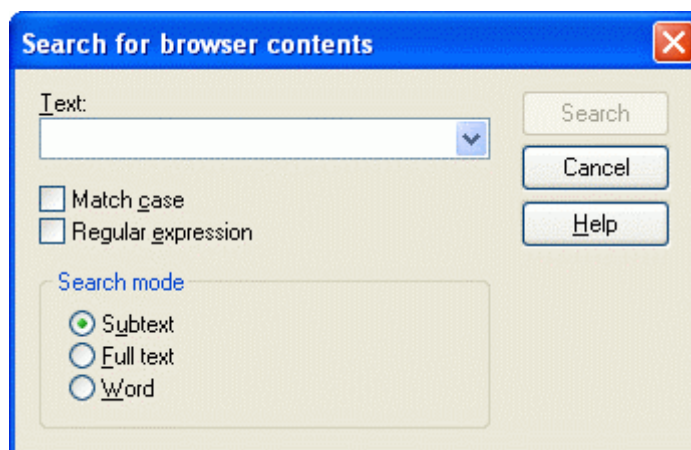
Figure 39: Search for browser content

Enter the text (character or expression) for which you would like to search for, into the input field "Text".

The following options are available for your search:

- **Match case**: when this option is selected the search will differentiate between upper and lower case letters.

- **Regular expression**: the default is to deactivate this option when searching for standard text. However by activating "Regular expression" (see chap. 4., p. 534) you can search the contents with the help of special characters.

  Hint: It is possible to use wildcards in the field "Text" ("*" for any number of characters, "?" for exactly one character).

  In order to use wildcards, the option "Regular expression" <u>must</u> be deactivated.

In addition, the following options are also available when searching for text:

- **Subtext**: means that the text being searched for can be part of a larger word or sentence.
- **Full text**: this is the default option - finds all cells which match exactly the full text being searched for.
- **Word**: finds all texts where the text being searched occurs as a single word (may be bounded by blanks or a full stop).

Clicking on the button "Search" begins the actual search. Each row, which is matched successfully during a search, will be highlighted in red.

5.13 Save

With the "Save" button you can save the displayed results to a file. The window "Save - options" (see fig. 40) is displayed once the button is selected.
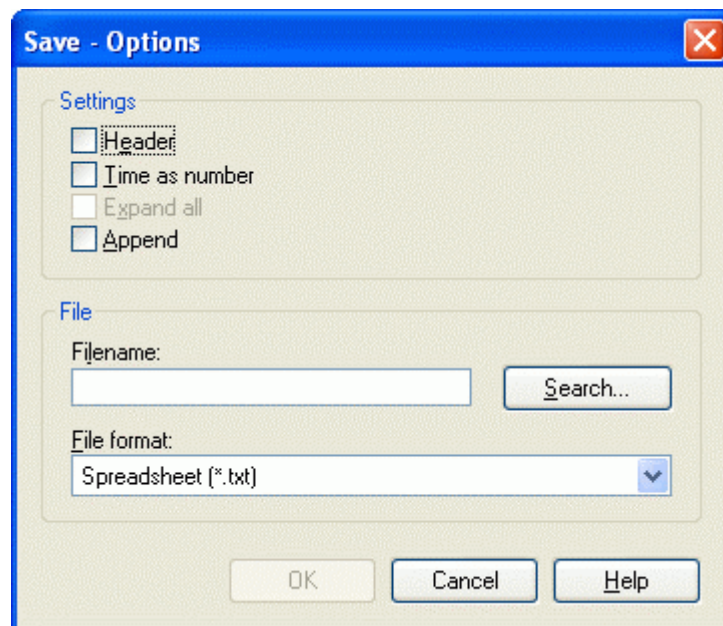
Figure 40: Save - properties

Within the "Save - properties" window you can choose:

- Properties (see chap. 5.13.1, p. 60)
- File formats (see chap. 5.13.2, p. 60) for external data

that have an effect on format and contents of the data.

ATTENTION: The available options depend on the type of results being displayed (e.g. Query results), i.e. not all options are always available.

In addition you must enter the filename and path to which the results should be saved in the input field "Filename".

Click on the OK button to continue.

5.13.1 Properties

The "Properties" allow you to influence the contents of the file being saved. The following properties are available:

Header:
> When this option is selected, a header is added to the beginning of the file being created, which contains general details on the results (including date, time, the query run and model name(s)).

Time as number:
> If this option is selected, then attributes which are currently saved in the ADONIS time format YY:DDD:HH:MM:SS will be converted to seconds in the file (e.g. the expression 00:000:00:30:00 will be converted to 1800 seconds).

Expand all:
> Selecting this option will cause the entire contents of the browser to be saved regardless of how many levels of a hierarchical ADONIS browser (see chap. 5.2.2, p. 48)are actually expanded.
>
> **Note:** This option is only available when the browser tree is not entirely expanded.

Append:
> The contents of the browser will be appended to an existing file.

5.13.2 Formats

By clicking on the file format list, you can select the type of file in which the results should be saved. It is very important to choose the correct format especially when you wish to work further on the results in a spreadsheet program (e.g. Microsoft Excel)

You can save the results

- for Spreadsheet (*.txt),
- for Spreadsheet (*.csv),
- for Word processor (*.txt),
- in Rich Text Format (*.rtf),
- as HTML file (*.htm) or
- as comparing representation (*.acr)

5.13.2.1 Save as spreadsheet (*.txt)

When the "**Spreadsheet (*.txt)**" format has been selected, you can specify user-defined cell separators to enable the resulting file to be easily used in spreadsheet applications.

After selecting the path and filename in the "Save - options" (see fig. 40) window and clicking on the OK button, the window "Separators for spreadsheet" is displayed (see fig. 41).
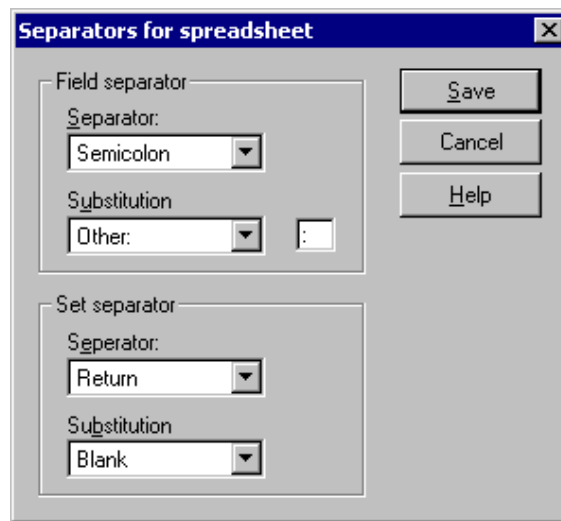
Figure 41: Save as spreadsheet

An entire row in the ADONIS browser - including the row description - is defined as a set. The "Set separator" identifies what character is used to indicate the end of a single row. A single cell is defined as a field and the "Field separator" is used to indicate an appropriate character to separate all cells.

In case the separator characters are already used within the data in the ADONIS browser, a "Substitution" character must also be defined which will be used to replace any separator characters that already exist.

The separator for the decimal point indicates which character should be used to separate decimal numbers (i.e. comma, full stop or other).

> **Hint:** To guarantee that also the cells which contain line breaks will be correctly copied into a spreadsheet program, you can also save the contents of the browser to a CSV-file (see chap. 5.13.2.2, p. 61).

5.13.2.2   Save as comma-separated value (*.csv)

When you have selected the format "**Spreadsheet (*.csv)**", you can edit the contents of the browser window without losses for a later integration into a spreadsheet program.

Like the file format **spreadsheet (*.txt)** (see chap. 5.13.2.1, p. 60), the CSV format enables the definition of a whole line as a record. Additionally the line breaks which may appear in the cells will be correctly interpreted.

5.13.2.3   Save as text (*.txt)

"**Word processor**" saves the contents of the ADONIS browsers in this format as pure text to the selected file (file extension .TXT). The single lines will end in the file with a line break.

5.13.2.4   Save as RTF file (*.rtf)

The format "**Rich Text Format**" saves the content of the ADONIS browser as an RTF file (file extension .RTF). The table format is kept and this RTF file can be opened and further edited in a word processing program.

5.13.2.5 Save as HTML file (*.htm)

By selecting the format "**HTML file**" the contents of the browser are saved in a file in HTML format (file extension .HTM) which can then be viewed through a HTML browser (e.g. Internet explorer).

5.13.2.6 Save as comparable representation(*.acr)

When the format **"Comparable representation"** is selected, the results are written in ACR format (**A**DONIS **C**omparable **R**epresentation).

5.14 Copy to Clipboard

When you select the menu item "Copy to clipboard" in the ADONIS browsers popup menu (right mouse button), the contents currently displayed in the ADONIS browser are copied to the clipboard and can then be pasted into Microsoft Excel for example in order to carry out some further work or calculations.

5.15 Print

In order to print the results displayed in the ADONIS browser, click on the button "Print". This will cause the window "Print" (see fig. 42) to be displayed:
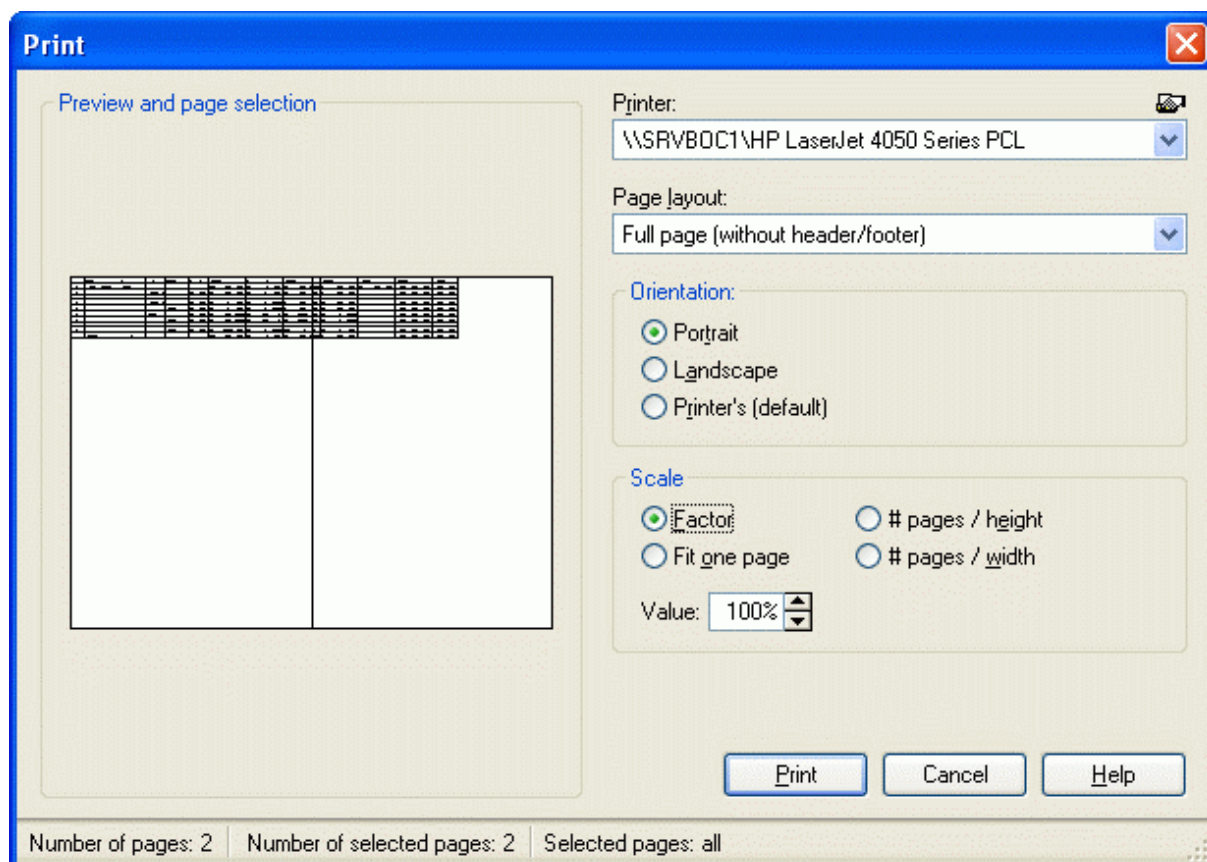
Figure 42: ADONIS-Browser-Print

Print:

The field **"Print"** contains a list of all available printers. Select a printer from the list (default printer is pre-selected). The printer settings can be checked and if necessary changed by clicking on the smart icon 🖶 . If any problem with printers occur contact your system administrator.

Choose your settings for **"Page layout"**, **"Layout parameters"** and **"Scale"**.

The window **"preview and page selection"** is updated with each change. This print preview helps to choose pages for the printout (by allowing the user to mark the pages that aren't to be printed).

In the **status line** you will find information about the printout of several pages depending on the current layout and scale (total number of pages, number of selected pages, numbering of selected pages).

> **Hint:** The numbering of pages is done by the line from the left to the right.

To start printing click **"Print"**. In the status window, you will find information about current printing status. By clicking on the "Cancel" button, the printing process will be cancelled immediately.

5.16 Diagram

In the window "<Browser content> - Diagram settings" (see fig. 43) you can define the settings for the graphical display of results.
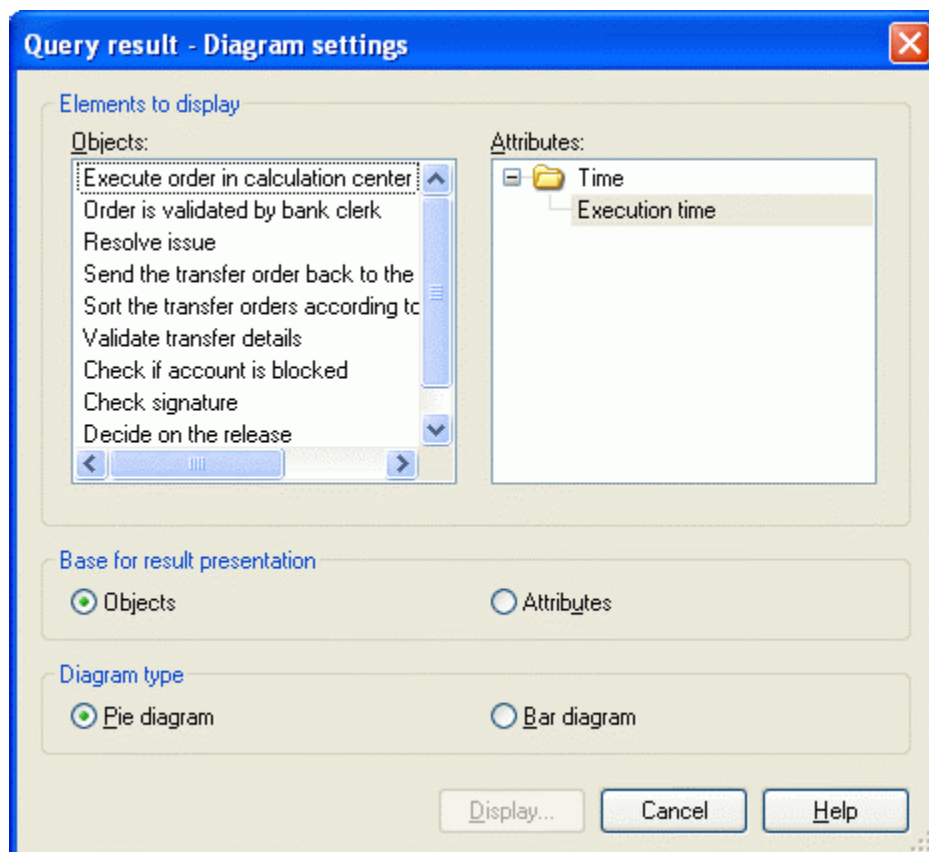
Figure 43: Diagram settings

Select the objects and attributes to be displayed.

> **Hint:** You can only select time or number attributes.

Define either an "object" or an "attribute" **basis for the results display**. The "object" option basis shows the selected attribute per object. The "attribute" option basis gives the selected objects per attribute.

The **type of diagram** sets the type of graphical display. It is possible to display the results either as a bar diagram (see chap. 5.16.1, p. 64) or a Pie chart (see chap. 5.16.2, p. 66). The bar diagram shows a group per value with a bar for each result. The pie diagram shows the part of each result for the selected value.

       ATTENTION: Only one attribute can be displayed as a "Pie chart".

Once you have chosen the selected settings, click on the **"Display" button**. In the "Result comparison - graphical display" window, the results will be shown as graphics. If the graphical representation is bigger than the drawing area in the window, a scroll-bar will also be available to enable you to see the entire diagram.

On the right of the graphical result display you will find options for the following general layout settings, which are available for both types of diagram:

- "**Value**": When this option is selected, the appropriate values - in time or number format - are included in the graphical display.

- "**Legend**": By activating this option, the diagram will be described more precisely using a legend.

- "**Axis labels**": By activating this option, the axis of the diagram will be labelled.

- "**Items per page**": This refers to the number of items shown on one page. Changing this value can give you a better overview of the diagram. By reducing the number of items per page, the undisplayed items will be moved onto the next page.

- "**Page**": The page number of the current page is displayed. Changes to the **field "Items per page"** causes the total number of pages to change. If, for example the number of entries per page is reduced from 15 to 10, then the number of pages may be increased from 1 to 2.

You can copy (see chap. 5.16.3, p. 67) or print (see p. 41) the displayed diagram to the clipboard or to a graphic file.

By clicking on the "cancel" button you will close the graphical result display.

5.16.1   Bar Diagram

Results displayed in a bar diagram (see fig. 44) show several attributes of a type, (time or number) with their attribute values. The attributes will be displayed in a legend on the left upper side of the graphical results window.
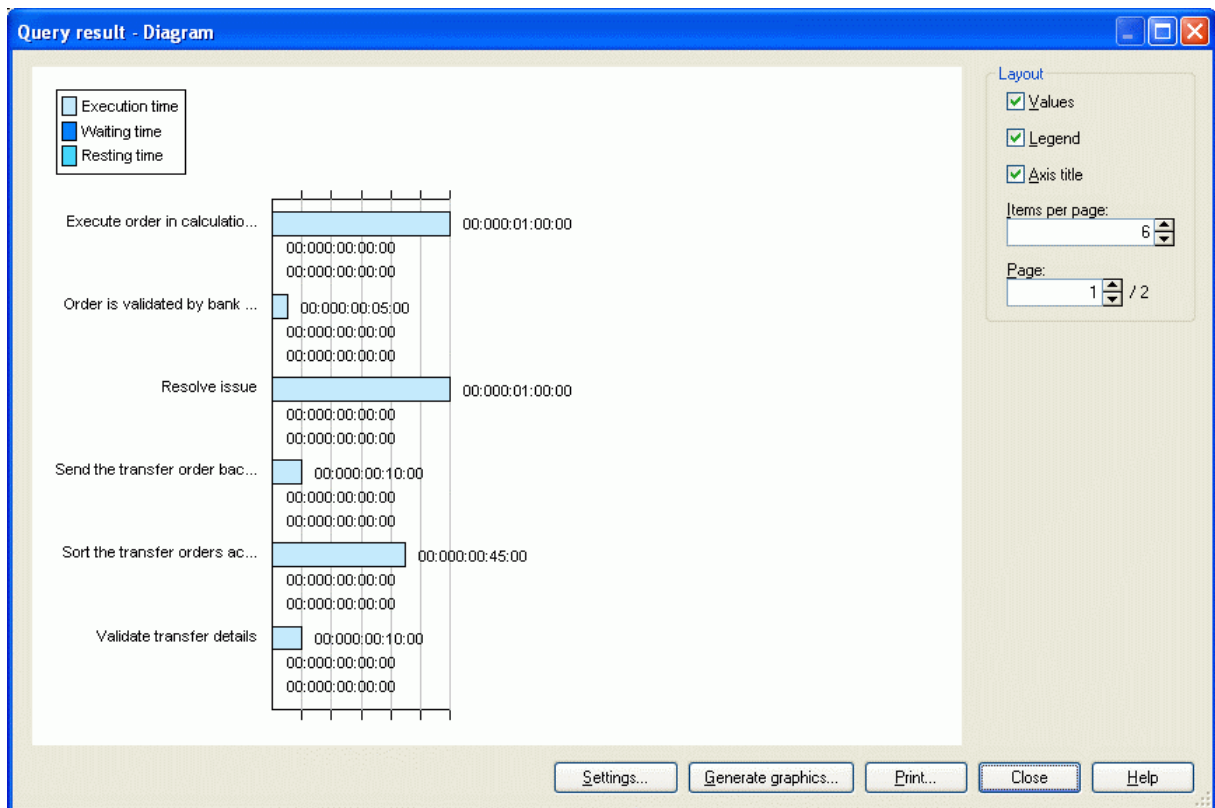
Figure 44: Bar diagram

On the right of the graphical result display you will find options for the following general layout settings:

- "**Value**": When this option is selected, the appropriate values - in time or number format - are in-cluded in the graphical display.

- "**Legend**": By activating this option, the diagram will be described more precisely using a legend.

- "**Axis labels**": By activating this option, the axis of the diagram will be labelled.

- "**Items per page**": This refers to the number of items shown on one page, changing this value can give you a better overview of the diagram. By reducing the number of items per page, the remain-ing items will be moved onto the next page.

- "**Page**": The page number of the current page is displayed. Changes to the **field "Items per page"** causes the total number of pages to change. If, for example, the number of entries per page is reduced from 15 to 10, then the number of pages may be increased from 1 to 2.

Click on the "Settings" button, to change the settings for the bar diagram display (see chap. 5.16.1.1, p. 65).

You can save the displayed diagram to the clipboard or copy (see chap. 5.16.3, p. 67) or print (see p. 41) it to a graphic file.

5.16.1.1   Settings for the Bar Diagram Display

You can change the settings concerning the display of the bar diagram in the "Diagram settings" win-dow.
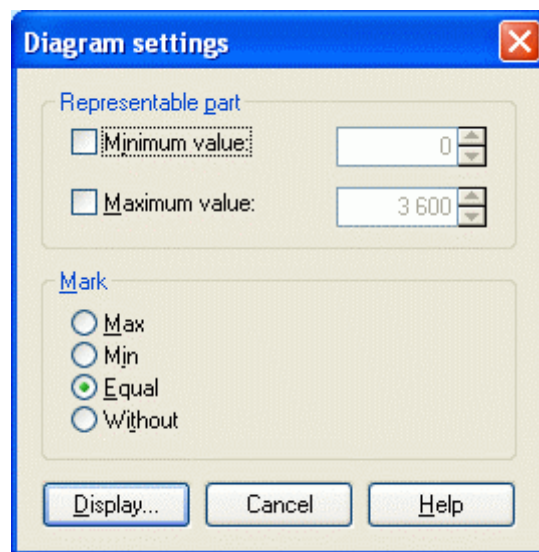
Figure 45: Settings for the bar diagram

In the **"Part to display"** window you can limit the bar diagram to a specific area by entering a lower and/or an upper limit. (For "time" attributes, the lower and the upper limit will be measured in seconds.)

The **"Mark" field** options enables better graphical editing of the files. You can click and select one of the following options for the display of the scale:

- "**Equal**" divides the length of the bar diagram in equal sections.

- "**Min**" marks the smallest attribute value.

- "**Maximum**" marks the largest attribute value.

- "**Without**" shows the diagram without divisions.

Click on the "Display" button to show the bar diagram with the changed settings.

5.16.2   Pie Chart

Only one attribute can be displayed within a pie chart (see fig. 46).

> **Hint:** A graphical representation of "Pie chart" is only available for the results of the "Capacity Analysis" and "Workload Analysis" Simulation algorithms as well as the comparison of results.
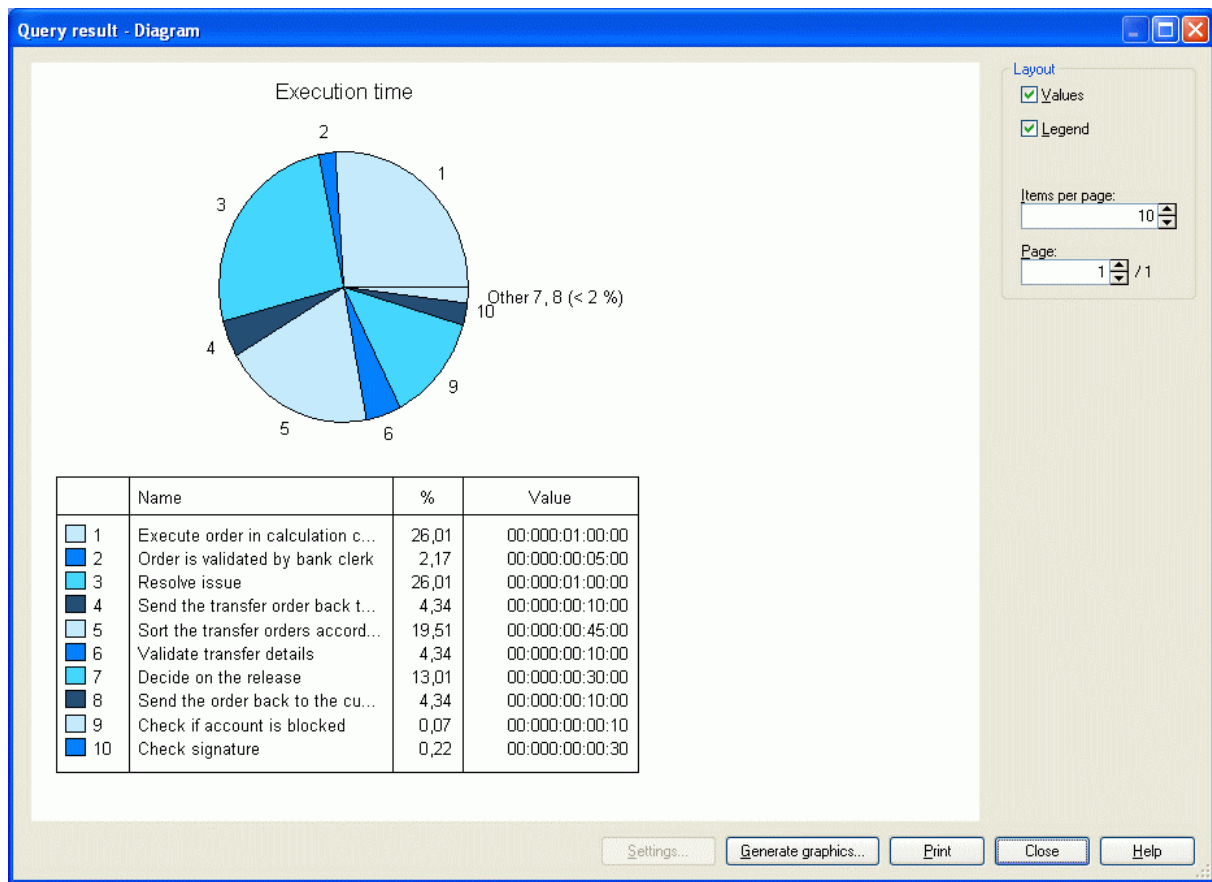
Figure 46: Pie chart

Each section of the pie chart contains a number, which refers to the name of the entry, its value and its percentage of the total sum. They are displayed in a tabular form, below the pie chart.

To the right of the graphical representation the following options are available:

- "**Values**": When this option is selected, the appropriate values - in time or number format - are included in the graphical display.

- "**Legend**": By activating this option, the diagram will be described more precisely using a legend.

- "**Page**": The page number of the current page is displayed. Changes to the **field "Items per page"** causes the total number of pages to change. If, for example, the number of entries per page is reduced from 15 to 10, then the number of pages may be increased from 1 to 2.

5.16.3   Generate Graphics

The button "Generate graphics" allows you to copy the displayed diagram onto the clipboard or to store it as either a BMP file (1-bit or 24-bit), a PCX file(8-bit or 24-bit) or a JPG file (24-bit), PNG file (24-bit) or. EMF file (24-bit).

Clicking on this button causes the window "Generate graphics" (see fig. 47) to be displayed.
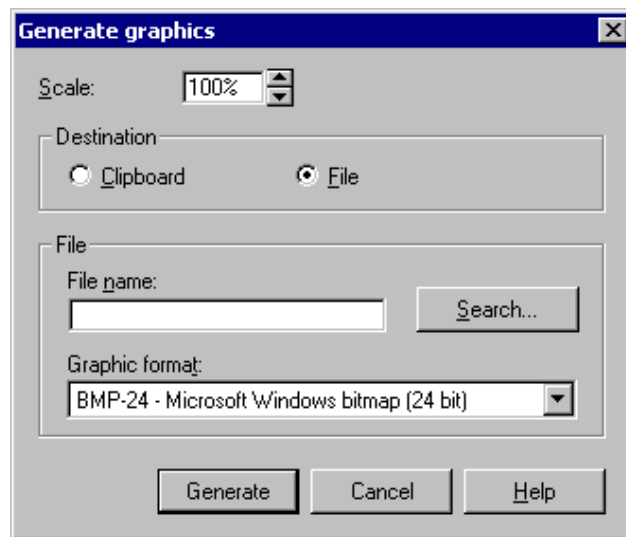
Figure 47: Generate graphic

Select the destination to which the graphic should be copied. If the clipboard is selected, no more options need to be set.

If copying to a file, then you must enter a path and filename in the field **"Filename"**. Additionally, you can select the size of the graphic to be generated in the field **"Scale"** and the type of graphic in the field **"Graphic format"**.

Click on the button "Generate" to actually generate the graphic.

6.   Versioning

6.1   Basics

ADONIS supports the following types of model and attribute profile versioning:

- **model-related versioning** (see chap. 6.1.1, p. 69)

- **time-related versioning** (see chap. 6.1.2, p. 69)

> **Hint:** In the application library, one versioning type is defined and used for all model types available.

6.1.1   Model-related Versioning

In the model-related versioning, a version number text can be assigned to each model. This is the case when creating, saving as another version or renaming the model.

ADONIS neither demands a specific format nor a specific semantic for the version number. According to the use case, the user can define the format and semantics.

IIt is possible to create a new model version by creating a new model and entering the same model name with a different version number.

> **Hint:** In model-related versioning you can only create version numbers for models, not for attribute profiles (see chap. 8., p. 77) .

6.1.2   Time-related Versioning

In time-related versioning, a **date of validity** will be assigned during the creation of every model and every attribute profile. This indicates the date from which the attribute profile version becomes valid. The validity of a version ends with the start of validity of the next version. The youngest version of a model is the valid one.

Dates of validity are of significance especially for the use of inter-model references and attribute profile references : References can only be created on models or on attribute profiles, which are valid at the same time as the referencing model.

The format of the version number can consist of the elements "Day", "Month" and "Year".

Example:

If you do not want to create a version with a day level, the format of the version number can for instance consist only of month and year (see fig. 48).
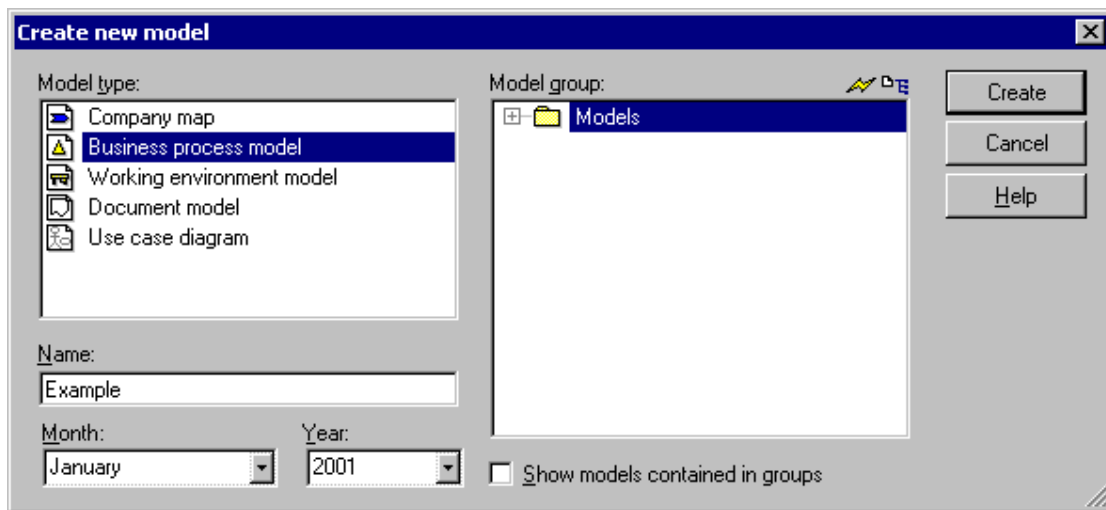
Figure 48: Example of time-related versioning when creating models

6.2 Effects

The versioning of models and attribute profiles has effects on the following functions:

- **ADONIS Business Process Management Toolkit:**

  - Create new model

  - Create new version of an already existing model (only for time-related versioning)

  - Rename models

  - Save model with a new name

  - Define application models (only for time-related versioning)

  - Create and follow inter-model references (only for time-related versioning)

    The order of inter-model references concerns all functions, for which models or attribute profiles have been selected for further editing (e.g. Analysis, Simulation, Evaluation, Import, Export).

  - Create attribute profile references (see chap. 4.1.6, p. 340) (only for time-related versioning)

  - Edit attribute profiles (only for time-related versioning)

- **ADONIS Administration Toolkit:**

  - Edit attribute profiles (see chap. 4.1, p. 337) (only for time-related versioning)

  - Models/Attribute profiles import (see chap. 3.4, p. 302)

  - Models/Attribute profiles export (see chap. 3.5, p. 325) (only for time-related versioning)

7. Inter-model References

With inter-model references, it is possible to reference from an object of one model into:

- other models, or
- objects in those other models.

In ADONIS 3.9 you can classify the inter-model references. This is of importance in all model selection lists, which contain the "Included referenced models" (e.g. "Open model", "Query", "ADL export"), since all referenced models of the models selected in these list will be determined and further edited according to their settings.

A "Reference tree" will be created with all models referenced by the settings. The figure below shows an example of a simple reference tree (see fig. 49).
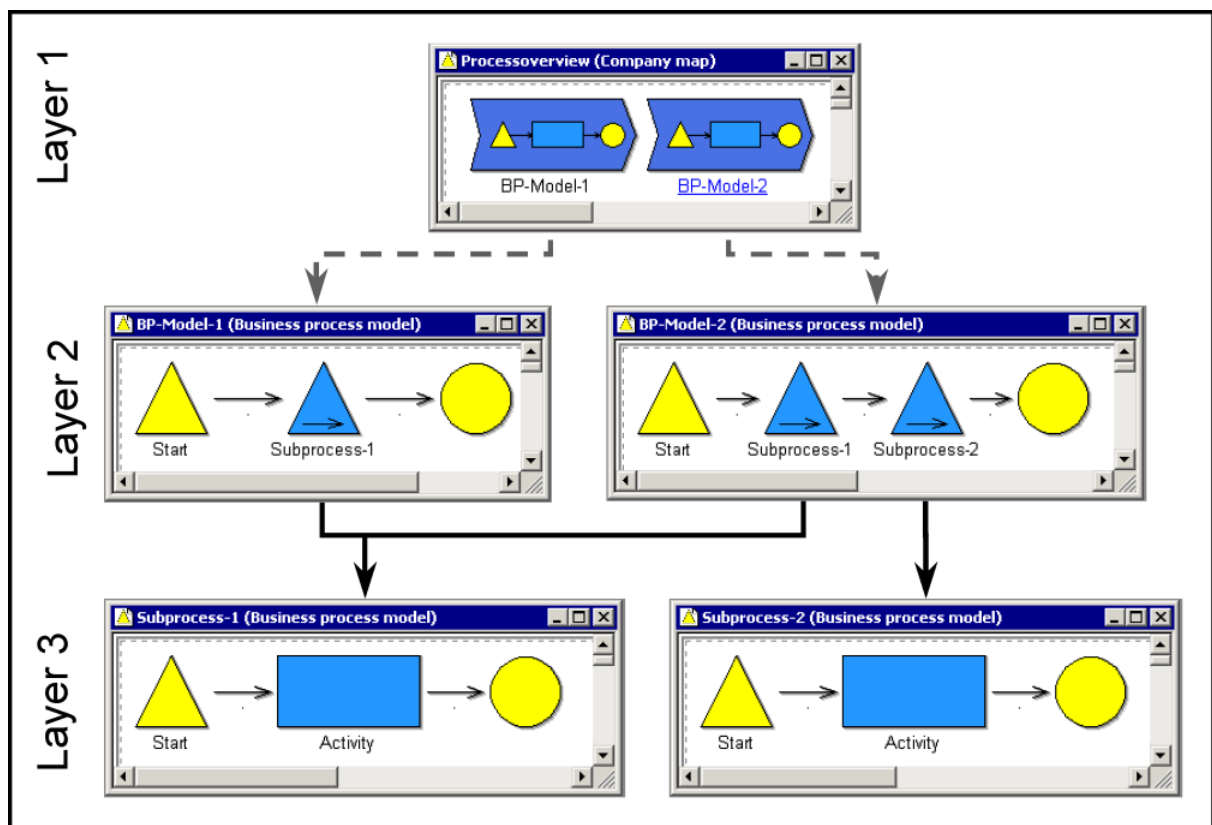
Figure 49: Example of reference tree

In the "main model" (Level 1) the models "Model 1" and "Model 2" (level 2) are referenced, they contain the models "referenced model A" and "referenced model B" (Level 3). Starting from the "main model", you can form the reference tree shown, i.e. by selecting the "main model" and activating the option "Including referenced models", the five models will be loaded for editing.

With the settings (see chap. 7.1, p. 74) for the inter-models referenced you will determine,

- the depth (level) of which the references should be taken and
- whether they are main or side references.

The **depth** indicates of which level (starting from the selected model) the references will be taken and the referenced models taken into account. In the example with the reference tree (see fig. 49) , a depth of 1 starting from the '"main model" would give us as a result only the models "model 1" and "model 2".

The classification **main reference** and **side reference** determines how to proceed after having followed a reference .

Main referenced models will be treated like the (selected) starting model, i.e. all references contained in this model will be followed, if the limit of the depth allows.

For **side referenced models**, the references contained in these models will only be followed if:

1. they start from the same attribute of an object of the same class and

2. the limit of the depth allows it.

In contrast to the main references, the limit of the depth of side references will be evaluated each time the side references are used (see fig. 50). The depth of main references is verified globally, over all levels.
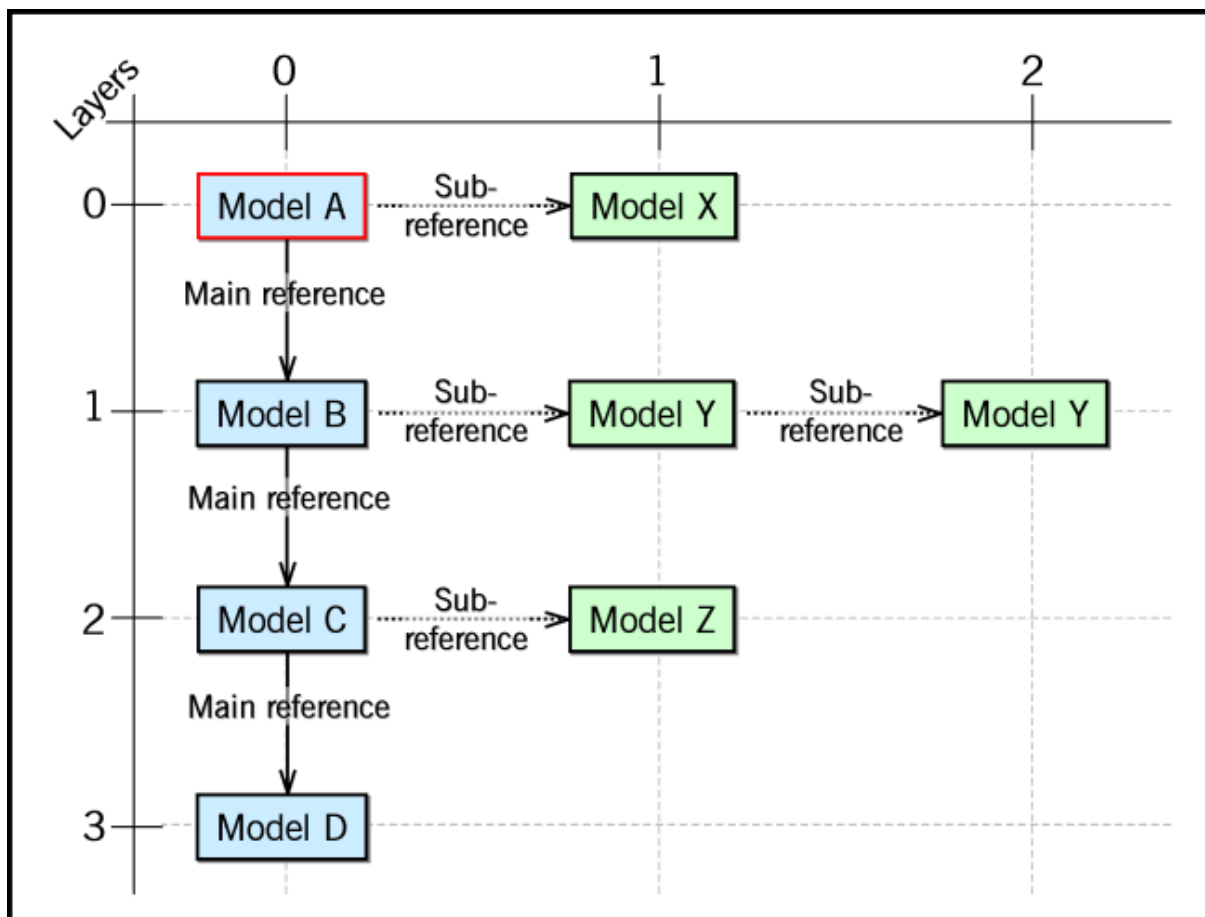
Figure 50: Schematic display of reference evaluation

In the schematic display of reference evaluation (see fig. 50) the "model A" is the original model, i.e. the model from which the reference tree is formed. The models: "model B", "model C" and "model D" will be referenced from the "model A" with a main reference (e.g. sub processes) and form together the reference tree of the main reference (main tree). A depth of "2" for this main reference means, that the "model B" will not be taken into account in the reference tree to be created.

The side references (e.g. documents) will be considered per level, i.e. the depth Evaluation is always done starting from a model of the main tree and the count starts again at 0. A depth by "1" for the side references means - looking at the figure above - that the "model N" is not considered when creating the tree.

Differences between main and side references are explained in the graphic below (see fig. 51).

  Hint: For the references in this example, the depth is set to <unlimited>.
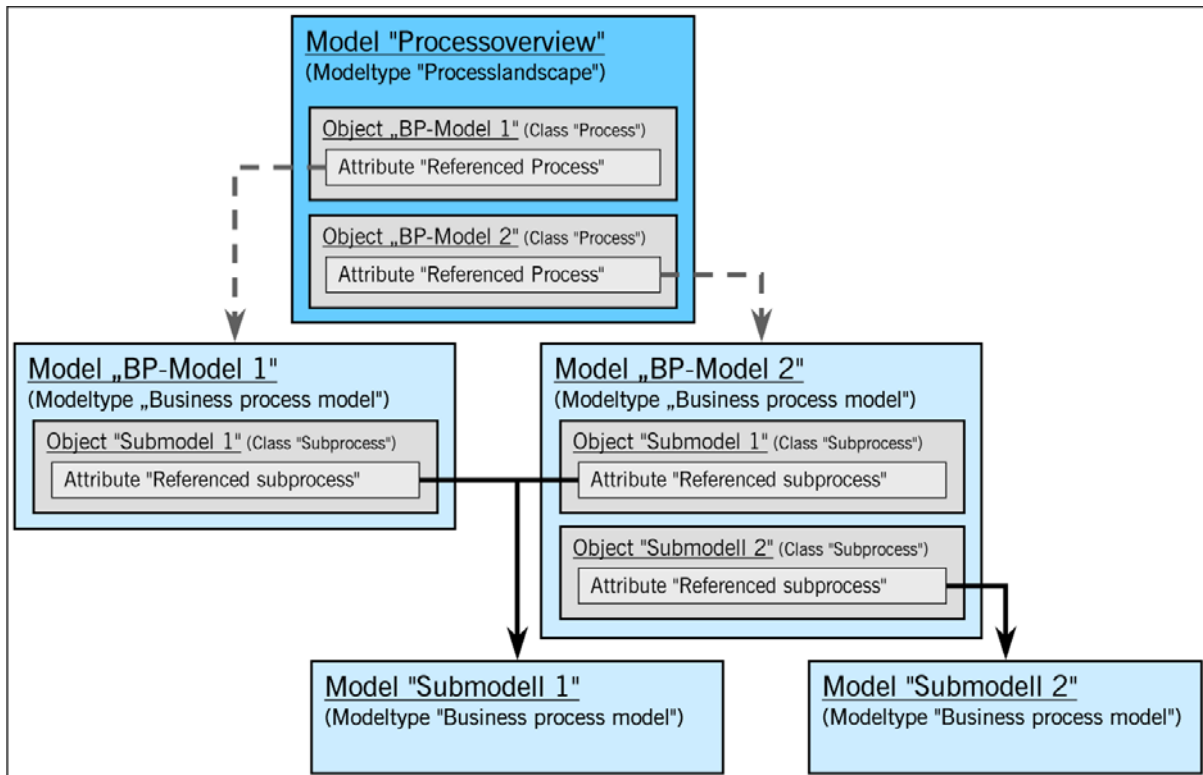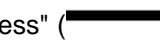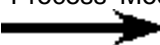
Figure 51: Example of a reference tree

The reference attributes "called process" in the class "process call" (in the models of the type "Business Process Model") and "referenced process" in the class "process" (in the models of the type "Process map") are defined within the ADONIS BPMS Application Library (see chap. 20., p. 693).

The references of the attribute "referenced process" are shown by ▬ ▬ ➤ , and those of the attribute "called process" by ━━━━➤ .

If the references of the attribute "referenced process" (▬ ▬ ➤) are defined as <u>side references</u>, the results are - starting from the "process overview" process map - the Business Process Models "BP model 1" and "BP model 2", irrespective of the definition of the references in the attribute "called process" (━━━━➤), since the results of side references are taken into account, only if the same side references refer to further models.

If the references of the attribute "referenced process" (▬ ▬ ➤) are defined as <u>main references</u>, the results are - starting from the "process overview" process map - all Business Process Model drawn, irrespective of the definition of the references of the attribute "called process" (━━━━➤).

7.1　Reference Settings

In the window "references - settings" (see fig. 52), you can edit the classification (main or side references) and the depth for all inter-model references defined in the application library.

> **Hint:** You can edit the reference settings in all the model selection lists (e.g. "Open model", "Query", "ADL export"), in which the option "including referenced models" is available and activated, by clicking on the "References" button.
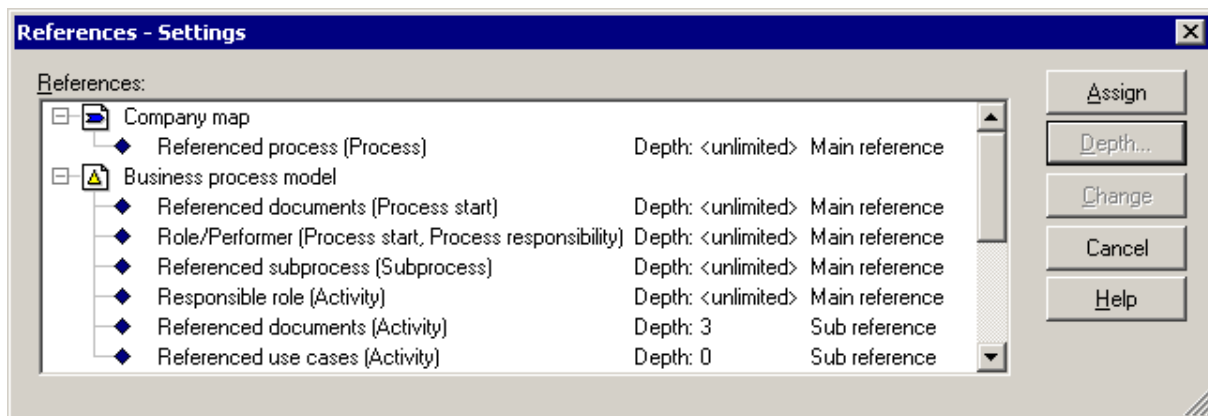
Figure 52: References - Edit settings

Select the reference to edit and adjust the **depth** (see chap. 7.1.1, p. 75) by clicking on the button "Depth".

Change the classification of a selected reference by clicking on the button "Change".

Alternatively, you can edit the selected reference via the context menu (see fig. 53) (right mouse button).
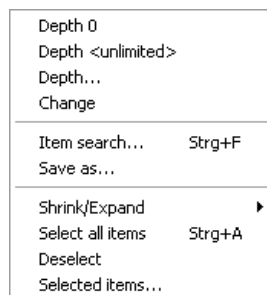
Figure 53: context menu to edit the reference settings

The following menu items are available to change the reference depth:

Depth 0　sets the value to "0" (the menu item will only be displayed, if the value of the selected reference is not 0).

Depth <unlimited>　sets the value by "<unlimited>" (the menu item will only be displayed, if the value of the selected reference is not <unlimited>).

Depth　opens the window "Edit reference depth" (see chap. 7.1.1, p. 75).

By selecting the "Change" menu item, you will change the classification (main reference <-> side reference) of the selected reference.

7.1.1 Edit Reference Depth

The reference depth gives the depth (down to which level) - starting from a selected model - you will follow the references and take the referenced models into account.

Enter the depth for the previously selected inter-model reference by selecting any predefined value (0 - 1 - 2 - 3 - 4 - 5 - <unlimited>) or by entering any value in the list "reference depth" (see fig. 54).
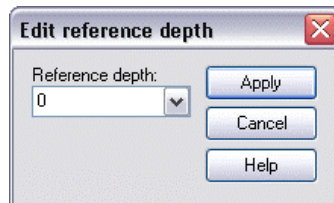
Figure 54: Edit reference depth

Once you have selected the reference depth, click on the "Apply" button to assign the value of the reference.

7.2 Effects of the Versioning on Inter-model References

Inter-model references for model-related versioning

For model-related versioning, inter-model references always point at concrete model versions or at objects in concrete model versions. The version number of the model only has an effect on the inter-model references, in so far as it (together with the model name) clearly identifies the target-model or the target-object.

Inter-model references for time-related versioning

For time-related versioning, inter-model references always point at version threads. The actual goal of a reference will be dynamically determined via the current version number: Inter-model references always point at the version of the target threads, which is valid at the same time as the source model.

Example:

1. Starting from "model A (April 2001)", an inter-model reference points at "model B". "Model B" is to be found only in the versions "January 2001" and "February 2001". In April 2001, the version "February 2001" is therefore still available. Therefore the target of the reference is "model B (February 2001)".
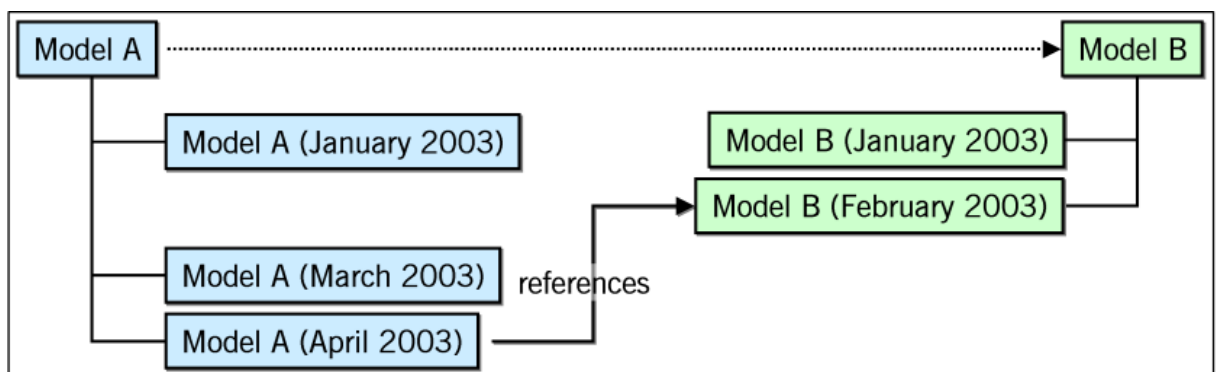
Figure 55: Reference from "model A (April 2001)" to "model B (February 2001)"

2. If a new version for "March 2001" is created from the "model B", it will be valid in April 2001. The inter-model reference points therefore automatically at "model B (March 2001)".
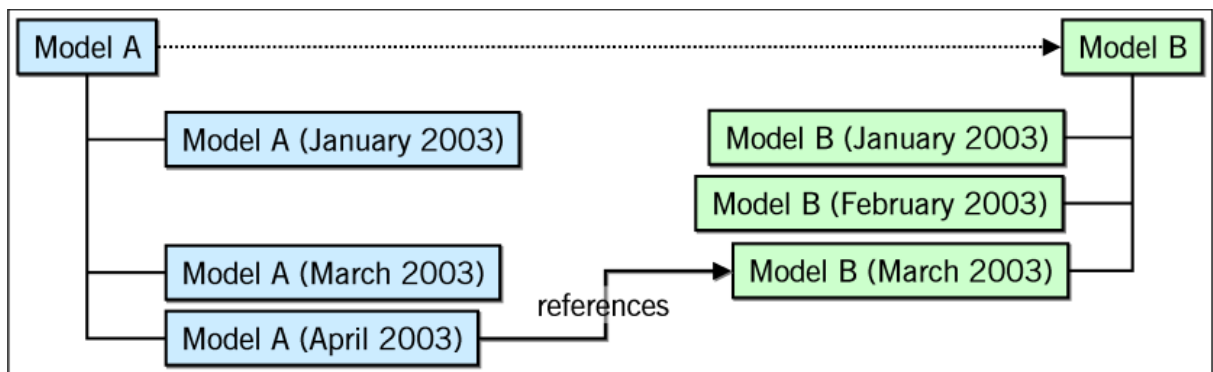
Figure 56: Reference from "model A (April 2001)" to "model B (March 2001)"

3. If moreover a new version for "May 2001" is created from "model B", this has no effect on the inter-model reference. In April 2001 "model B (March 2001)" is still valid. Therefore the inter-model reference still points at this version.
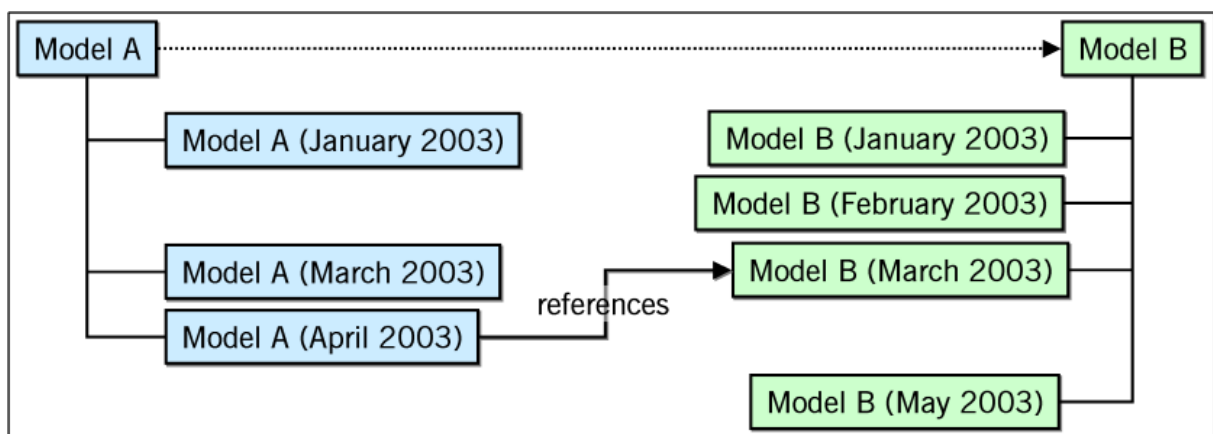
Figure 57: Reference from "model A (April 2001)" to "model B (March 2001)"

4. If a new version for "May 2001" is created from "model A", the intermodel- reference of "model A (May 2001)" points at "model B (May 2001)". The inter-model reference of "model A (April 2001)" still points at "model B (March 2001)".
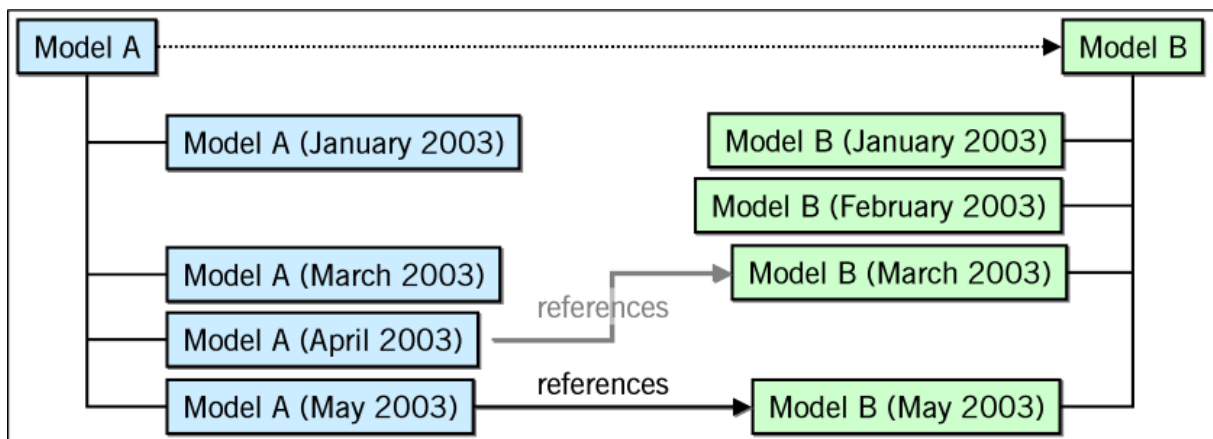
Figure 58: Reference from "model A (Mai 2001)" to "model B (May 2001)"

8. Attribute Profiles

With ADONIS attribute profiles, it is possible to create a repository, i.e. attribute structures which you can reuse in ADONIS.

With attribute profiles, you can define and change recurrent attribute values at a specific place (attribute profile administration). This way, the consistency of the deposited information is guaranteed.

An attribute profile consists of any number of attributes of any type (see chap. 5., p. 536).

> **Hint:** The attribute profile administration is a component of the Administration Toolkit. Additionally, authorised users can administrate attribute profiles in the Business Process Management Toolkit (modelling component).

The definition of the attribute profile class is done during the definition of the library, the creation and editing of attribute profiles is done in the attribute profile administration (see chap. 4., p. 335). The schemes are graphically displayed in the picture below (see fig. 59) and described in the following.
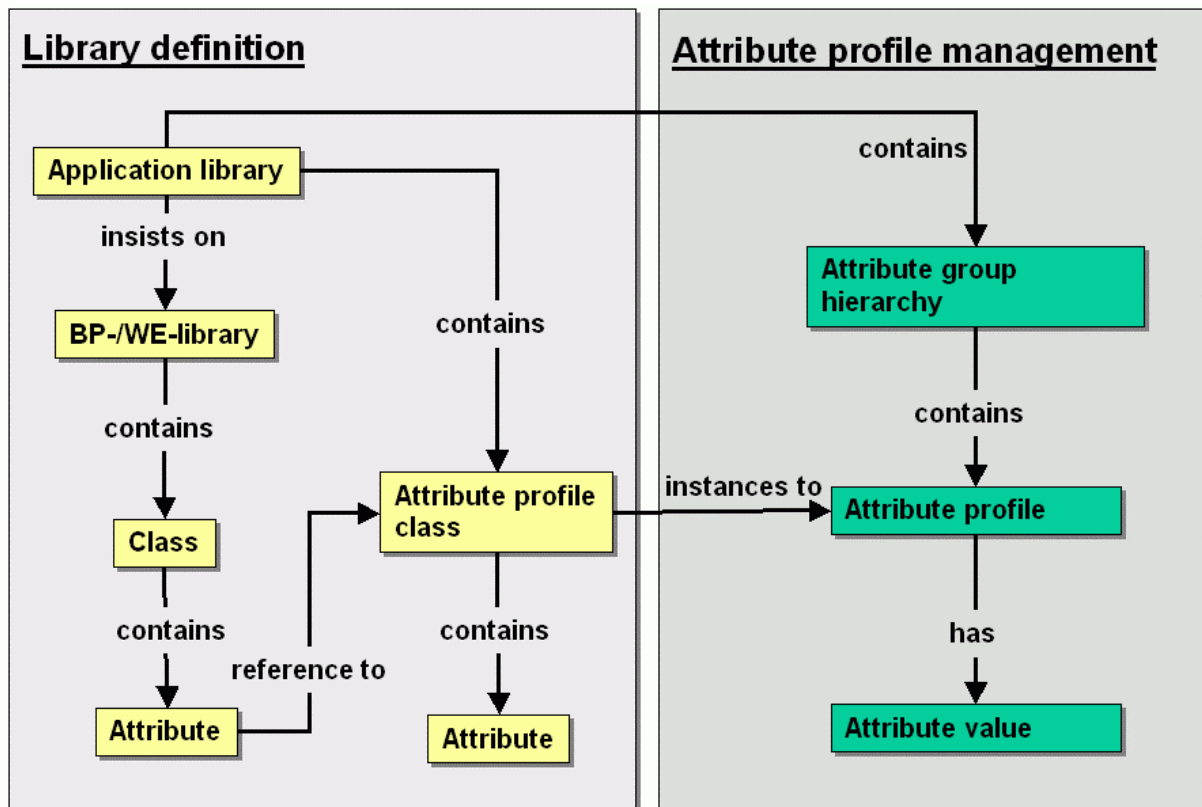
Figure 59: Definition of the library and attribute profile administration

Definition in the application library

Unlike modelling classes or relation classes (see chap. 1., p. 28), attribute profile classes will be defined in an application library and can be used in all model types.

Modelling and relation classes (= classes) will be defined in a Business Process or Working Environment library for model types of this library and thus are available in these model types (for modelling in the Business Process Management Toolkit).

Attribute profile classes and modelling/relation classes also contain definitions of (object) attributes. The connection of an object attribute (of a class) with an attribute profile class is done using a special referencing method.

The reference to an attribute profile class is defined in an attribute of a class, while the attribute type "attribute profile reference" (ATTRPROFREF) (see chap. 5.1, p. 536) is assigned to this attribute.

Editing in the Attribute Profile Management

 Concrete attribute profiles - i.e. attribute sequences determined with values - will be created, edited and deleted in the attribute profile administration (see fig. 59).

The creation of an attribute profile means that an attribute profile class has been instanced to an attribute profile (like instancing of an object of a class in the modelling in Business Process Management Toolkit).

Additionally in the attribute profile administration, it is possible to create so-called (attribute) profile group hierarchy, in order to set up user-specified structures for the attribute profiles.

9.   Record Attributes

Record attributes allow for the realisation of complex attributes. A record attribute is built like a table, i.e. with a lot of rows (0 is possible) and columns. Single rows can be added and deleted accordingly. The rows are numbered starting from 1. The columns display the attributes from which the record attribute is built. All types of files are allowed as possible attribute types inside a record attribute, except a complex attribute and an attribute profile reference. Default values can be defined for the single fields, so that they are already filled when adding a new row. A so-called multiplicity can be defined in each record attribute, which gives the maxiumum number of rows, which can constitute this record attribute.

The arrangement is graphically displayed in the picture below and described as follow (see fig. 60):
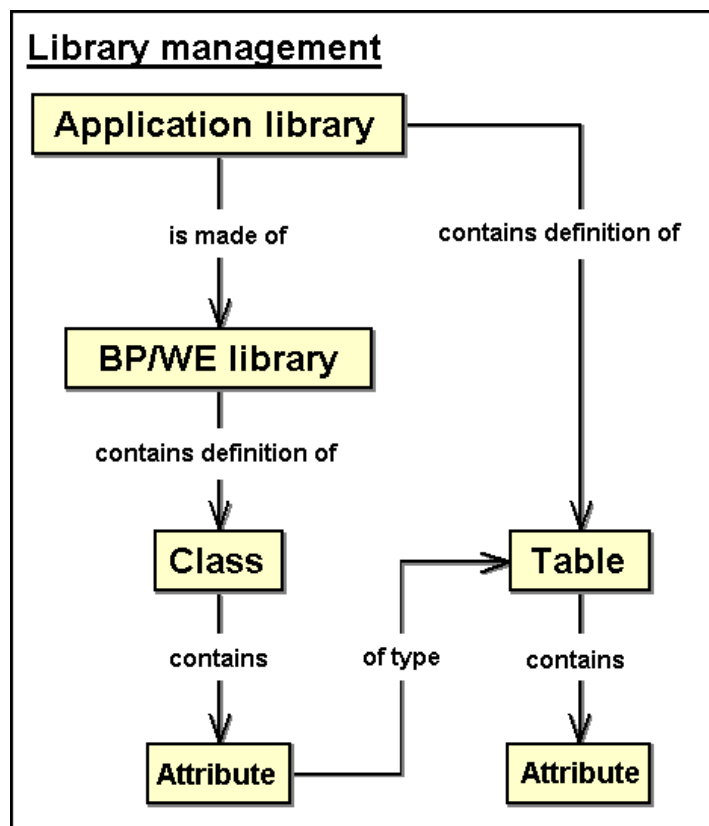
Figure 60: Schematic arrangement of record attributes in ADONIS

Unlike modelling or relation classes (see chap. 1., p. 28), record attribute classes will be defined in an application library and so can be used in all model types.

Modelling and relation classes (= classes) will be defined in a business process or Working Environment library for the model types of this library and are thus available exactly in these model types (for modelling in the Business Process Management Toolkit).

Record attribute classes and modelling/relation classes equally contain definitions of (object) attributes, which enable the connection of an (object) attribute of a class with a record attribute class through the definition of record attributes.

Usage in modelling
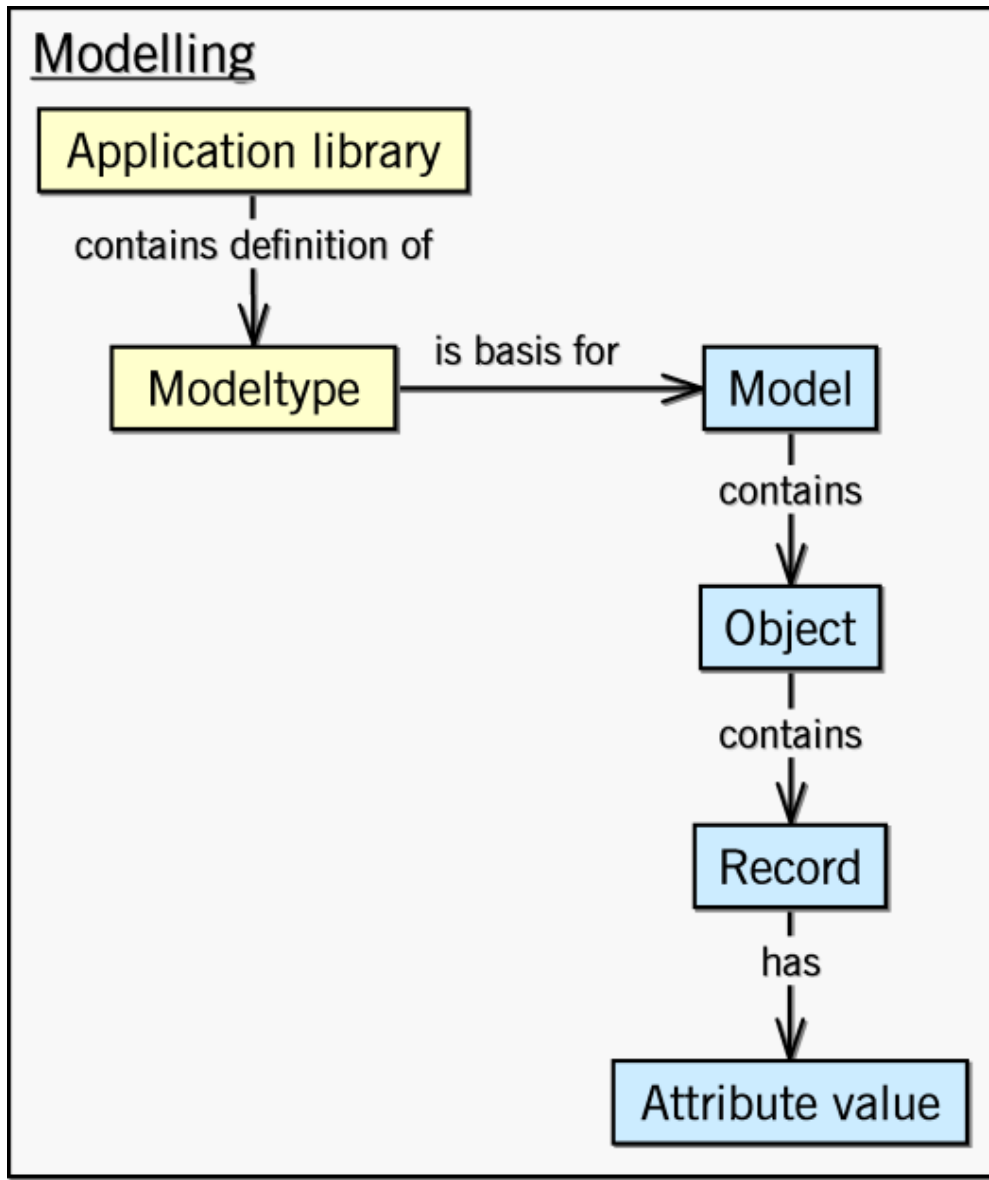
Figure 61: Record attributes in modelling

As previously mentioned in the description of the Terms and context (see chap. 1., p. 28), model types and classes will be defined in the application library.

Models of a specific type will be created in the modelling component. The objects of the available classes will be created again in a model. The (object) attributes will be determined with attribute values in an object.

If an object contains an attribute of the type "table" (see chap. 5.13, p. 539), the default value assignment is done via a record attribute class and the value it contains.

Record attributes are displayed graphically in the ADONIS browser (see fig. 62). You can edit the single fields the same way as for tabular modelling, i.e. support dialogues will be given for special attribute types. The icons ➕ and ✖ are used to add or delete selected rows.

Hint: Both functions are also available in the context menu of the ADONIS browser.
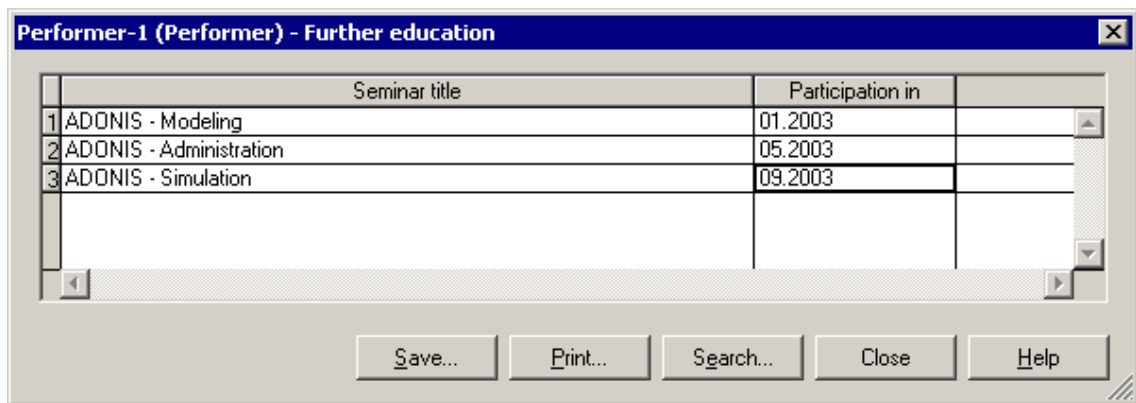
Figure 62: Display of a table attribute (example)

Each addition will cause a row to be added to the end of the table. The fields of the rows will be filled with the default values of the attributes.

To delete one or more rows from the table, select the row(s) by clicking in the numbered headers (first, grey column). You can also select several rows by pressing the control key and the shift key at the same time. Then click on the icon or select the appropriate menu item from the context menu. Confirm the appearing security query to delete the selected rows.

Part II

Part III

Components of the Administration Toolkit

Through the ADONIS Administration Toolkit (see chap. 3., p. 16) you control the administration of the ADONIS users, the ADONIS user groups, the ADONIS libraries, the ADONIS models, the ADONIS model groups and the configuration of the Business Process Management Toolkit.

The components of the Administration Toolkit: "User Management", "Library Management", "Model Management" and "Component Management" are explained in the following sections (see fig. 63).
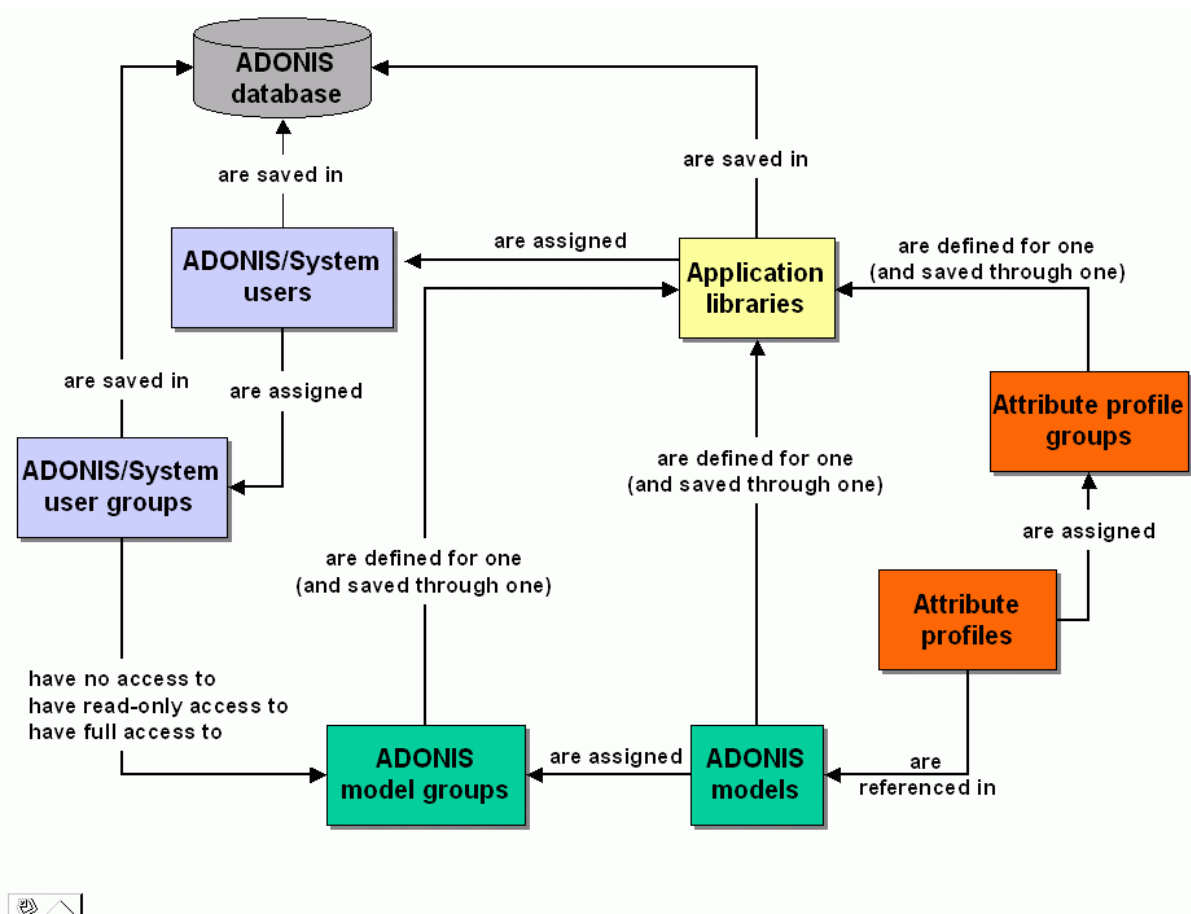
Figure 63: Overview of the components in Administration Toolkit

In the **User Management** (see chap. 1., p. 85) component, the ADONIS users and user groups needed for working with ADONIS (Business Process Management Toolkit and/or Administration Toolkit) are created. ADONIS application library (AL) is assigned to each ADONIS user. In addition, each ADONIS user is assigned to one (or more) user group(s). The ADONIS user's access rights to the different components of the Business Process Management Toolkit can also be defined here.

In the **Library Management** (see chap. 2., p. 125) component, the ADONIS application libraries - which consist of a business process library (BP library) and a working environment library (WE library) - can be imported, exported or deleted. By configuring the libraries, you can edit and check library and class attributes.

ADONIS users can create and edit business process and working environment models in the Modelling Component of the Business Process Management Toolkit. The BP and WE models are based on a BP or WE library respectively.

In the **Model Management** (see chap. 3., p. 292) component, model groups are created. Furthermore, it is possible to define which of the existing user groups have read-only access, read-write access or no access to these model groups. In addition, models can be imported, exported and deleted and application models can be imported and exported.

In **Attribute Profile Management** (see chap. 4., p. 335) attribute profiles can be created, edited or deleted. Attribute profiles are referenced to the Modelling Component of the Business Process Management Toolkit, and therefore belong to repository concept used in ADONIS.

Within the **Component Management** (see chap. 5., p. 379) it is possible to check which components of the ADONIS Business Process Management Toolkit (see chap. 4., p. 18) ("Acquisition", "Modelling", "Analysis", "Simulation", "Evaluation" and "Import/Export") are available to the ADONIS user. (The Modelling Component is always available. Additional components, which may have been purchased later, may be made available to the ADONIS user here too).

1.   User Management

ADONIS can only be accessed by authorised ADONIS users. Users will be authorised by the person responsible for the system (the ADONIS Administrator, e.g. the project leader), whose tasks are to create ADONIS users, to assign ADONIS application libraries to these users, to provide the ADONIS users with rights concerning the use of the ADONIS components and to assign the ADONIS users to ADONIS user groups.

Within this chapter a general description of the ADONIS user management component can be found:

● User management - overview (see chap. 1.1, p. 85)

● User management - Functionality (see chap. 1.2, p. 87)

● User management - Single-Sign-on functionality (see chap. 1.3, p. 88)

If you wish to use the services of the user administration, proceed according to the steps below:

1. Start the ADONIS Administration Toolkit (see Part IV., p. 527) . Once the toolkit is started, the window of the ADONIS Administration Toolkit will appear.

2. Activate "User Management" by clicking on the appropriate smart-icon ![icon] in the component panel.

Alternatively, you can activate the User Management by opening the pop-up menu in the component panel. Just click with the right mouse button on the component panel (on the right side of the smart-icons for the components) and select the option "User Management". Alternatively, you can open the pop-up menu using the function key <F9> and activate the User Management component by pressing key <e>.

As soon as you have activated the User Management component, the quick-access panel will show the smart-icons for the user list, the user group list, the user import and the user export administration (see fig. 64).
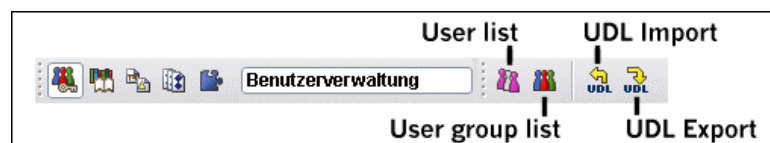
Figure 64: User Management - Components and quick-access bar

1.1   User Management - overview

The following diagram (see fig. 65) gives you an overview how the user management in the ADONIS Administration Toolkit is structured.
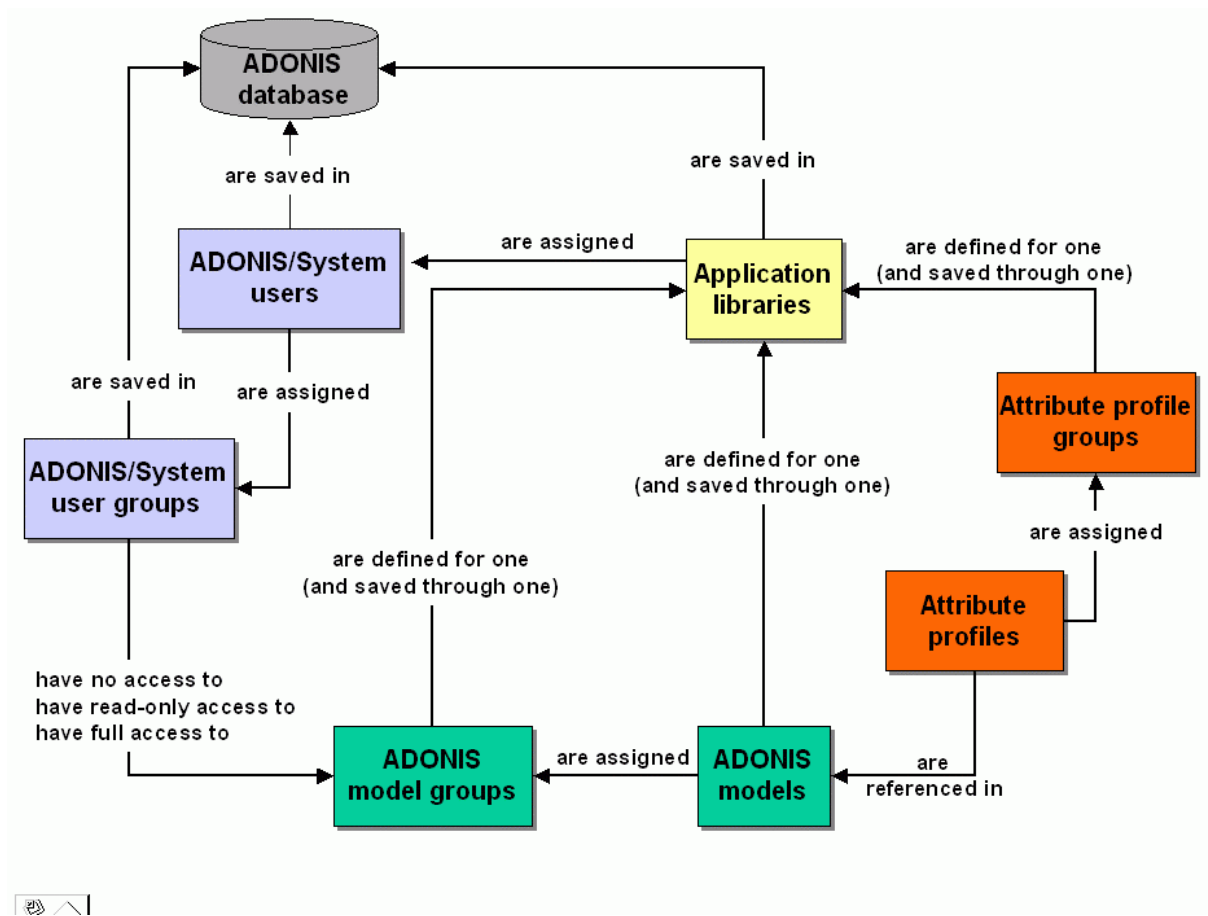
Figure 65: Overview of the components in the Administration Toolkit

In the centre of the User Management are the ADONIS users, created by an ADONIS administrator and thus stored in the ADONIS database. The ADONIS administrator both assigns an application library to the ADONIS user and defines his access rights to the ADONIS Business Process Management Toolkit and/or the ADONIS Administration Toolkit. All ADONIS users are shown in the User list (see chap. 1.4, p. 89).

When using the Single Sign-on functionality **system users** can be imported by the ADONIS administrator and stored within the ADONIS database. The ADONIS administrator assigns the system user to an application library and defines the access rights for the Business Process Management and/or Administration Toolkit. All system users are shown together with the ADONIS users in the **User group list** (see chap. 1.4, p. 89).

The ADONIS users are assigned to **ADONIS user groups** and the system users are assigned to **system user groups**. The ADONIS user groups and the system user groups are also created by the ADONIS administrator and stored in the ADONIS database. All ADONIS and system user groups are shown in the User group list (see chap. 1.5, p. 106).

> **Hint:** The ADONIS user groups establish a connection to the ADONIS models via the ADONIS model groups, i.e. an ADONIS user can only create or edit ADONIS models after he has been assigned to an ADONIS user group and this ADONIS user group has been equipped with read/write access to an ADONIS model group.

> **Hint:** The system user groups together with the ADONIS model groups determine access to the ADONIS models, i.e. a system user can edit and create ADONIS models only when assigned to a system user group and the system user group possesses write access to an ADONIS model group.

The ADONIS (system) users and/or ADONIS (system) user groups stored in an ADONIS database can be exported into an UDL (UDL=User Definition Language) file for save/backup or imported into any other ADONIS database.

The **standard user "Admin"** and the **standard user group "ADONIS"**, as well as the **standard system user group "ADONIS"** occupy special positions in the user management.

The **standard user "Admin"** can be neither deleted, exported nor imported, the settings of the standard user may not be changed except the password (see chap. 7., p. 547), which can be changed by the Admin user. This ensures that access to the Administration (see chap. 3., p. 16) and Business Process Management Toolkits (see chap. 4., p. 18) is possible at all times.

The **standard user group "ADONIS"** can be neither deleted nor renamed. When importing the standard user group only the ADONIS users (excluding the standard user) are imported.

The standard user "Admin" and the standard user group "ADONIS" are automatically created when the ADONIS database is created. The ADONIS BPMS Application Library (see chap. 20., p. 693) is assigned to the standard user "Admin", he has access rights to the Business Process Management Toolkit (see chap. 4., p. 18) and Administration Toolkit (see chap. 3., p. 16) and he is assigned to the standard user group "ADONIS".

The **standard system user group "ADONIS"** is automatically created when the Single-Sign-on functionality has been installed or activated. The default system user group "ADONIS" cannot be deleted or renamed. When importing the standard user group only the system users contained can be taken over.

1.2 User management - Functionality
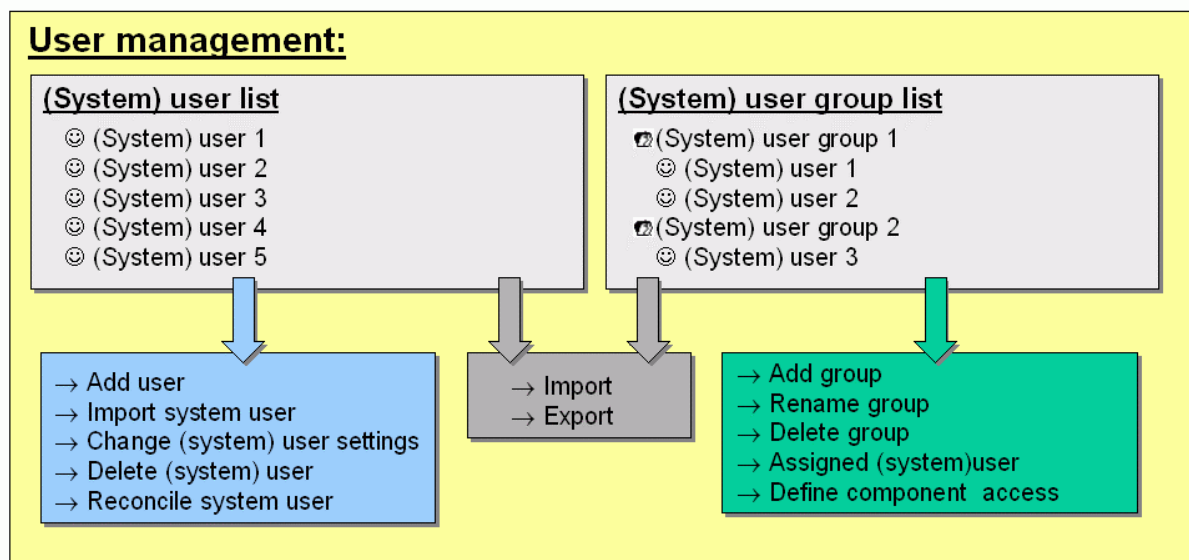
Figure 66: Functionality in the user management

The following functions are available to the ADONIS administrator within user management:

- via the <u>**User list**</u> (see chap. 1.4, p. 89)

  - **Add ADONIS users** (see chap. 1.4.3, p. 91)

  - **Edit user list** (see chap. 1.4.6, p. 100)

  - **Delete ADONIS users** (see chap. 1.4.9, p. 103)

- via the <u>**User group list**</u> (see chap. 1.5, p. 106):
  - **Add user groups** (see chap. 1.5.2, p. 107)
  - **Rename user groups** (see chap. 1.5.5, p. 112)
  - **Assign ADONIS users to user groups** (see chap. 1.5.8, p. 113)
  - **Define component access for user groups** (see chap. 1.5.10, p. 115)
  - **Delete user groups** (see chap. 1.5.7, p. 112)
- **Import ADONIS users** (see chap. 1.6, p. 116)
- **Export ADONIS users** (see chap. 1.7, p. 122)

1.3   User management - Single-Sign-on functionality

The **Single-Sign-on functionality in the ADONIS Administration Toolkit** enables that system users of the Windows operating system to be imported into the ADONIS User Management, that these users can be assigned to ADONIS application libraries, system users can be assigned access rights to use the ADONIS components and also that these system users can be assigned to (ADONIS) system user groups.

Furthermore the **Single-Sign-on functionality in ADONIS** (Business Process Management Toolkit and Administration Toolkit) supports for those users imported into the ADONIS user management that they can use their system user account to log into ADONIS without providing their user name and password.

> **Hint:** For reasons of easier reading the **term "system user"** will relate within this document always to **operating system users which have been imported into the ADONIS user management** (i.e. the so-called "ADONIS system user").
>
> The **term "system user group"** within this document will always relate to an **an ADONIS user group for (ADONIS-) system users** (i.e. to a "ADONIS system user group").

> **ATTENTION:**   The system users are represented in the ADONIS user management, i.e. changes in the ADONIS user management (e.g. deleting an ADONIS system user) will only have an impact with regard to the use of ADONIS. Changes within the operating system administration will not lead to an automatic update in ADONIS, i.e. when moving an operating system user into a different operating system user group, or when deleting or renaming an operating system user the ADONIS administrator will have to perform these changes for system users (see chap. 1.4.5, p. 100) or for system user groups (see chap. 1.5.4, p. 108) in ADONIS.

The following Single-Sign-on functionality is offered in the user management:

- via the <u>**User list**</u> (see chap. 1.4, p. 89):
  - **Define system user reference** (see chap. 1.4.4, p. 96)
  - **Reconcile system user names** (see chap. 1.4.5, p. 100)
  - **Change system user settings** (see chap. 1.4.7, p. 101)
  - **Delete system users** (see chap. 1.4.9, p. 103)
- via the <u>**User group list**</u> (see chap. 1.5, p. 106):
  - **Add system user groups** (see chap. 1.5.3, p. 108)
  - **Reconcile system user groups** (see chap. 1.5.4, p. 108)

- - **Rename system user groups** (see chap. 1.5.6, p. 112)
  - **Assign system users to system user groups** (see chap. 1.5.9, p. 114)
  - **Define component access for system user groups** (see chap. 1.5.10, p. 115)
  - **Delete system user groups** (see chap. 1.5.7, p. 112)
- **Import system users** (see chap. 1.6, p. 116)
- **Export system users** (see chap. 1.7, p. 122)

1.4 User list

The user list is the starting point to administer ADONIS users (see fig. 66). Within the user list all ADONIS users in the ADONIS database will be shown.

Select the option "User list" from the menu "User" or click on the respective smart-icon 👥 in the quick-access panel.

The window "User Management - User list" (see fig. 67) will appear on your screen, showing all ADONIS users sorted by the criteria as defined.
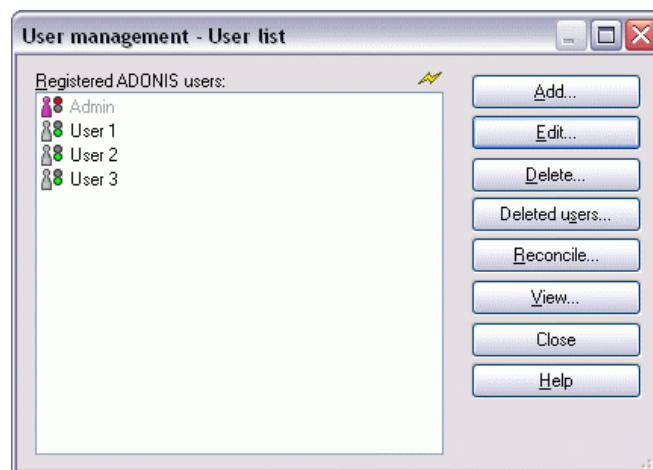
Figure 67: User list (with system users)

In the user list an icon infront of a user's name shows which ADONIS users are logged in at the moment 👥 ("red traffic light") and which ADONIS users are currently not working with ADONIS 👥 ("green traffic light").

When using Single-Sign-on functionality for imported system users the icon infront of a system user name indicates, which system users are currently logged in 👥 ("red traffic light") and which system users are currently not working with ADONIS 👥 ("green traffic light").

By clicking on the button:

"Add" you may add new ADONIS users (see chap. 1.4.3, p. 91) or system users (see chap. 1.4.7, p. 101);

"Edit" you may change (see chap. 1.4.6, p. 100) the user settings of ADONIS users (see chap. 1.4.6, p. 100) or system users (see chap. 1.4.7, p. 101) stored in the ADONIS database;

"Delete" you may remove (see chap. 1.4.9, p. 103) ADONIS users or system users stored in the ADONIS database from this database;

"Deleted users"   will list (see chap. 1.4.9.1, p. 104) ADONIS or system users which have been re-
moved from the database;

"Reconcile"   will reconcile (see chap. 1.4.5, p. 100) the system user names in the ADONIS user
management with those in the operating system's user administration;

"View"   will show (see chap. 1.4.1, p. 90) additional user information or let you choose
(see chap. 1.4.2, p. 91) the sort criteria for the listed users;

"Close"   you may close the window "user list".

Opening the **context menu** (right mouse click) will provide you with general functionality
(see chap. 4.1, p. 34), the functionality for the user list listed above and the **function "User pool"** to
show all users currently not assigned (see chap. 1.4.10, p. 105).

> **Hint:** The standard user "Admin" cannot be selected, since you can neither change its set-
> tings nor delete its user.

1.4.1 Information to show in user list

When pressing the button "View" a menu (upper part) is displayed to select the information to be
shown in the user list (see fig. 68).
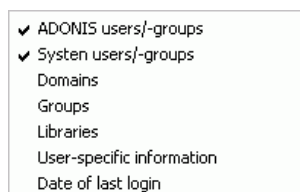
Figure 68: List of information to show in user list

The information currently being shown is marked.

Click on one of the menu points to include the selected information in the user list.

The following information can be shown:

ADONIS user/groups
ADONIS user in the user list or ADONIS user groups in the user group list.

System user/groups
System user in the user list or user groups in the user group list.

Domains
Domain names of the system users.

Is only available in the user list when system users are shown.

Is only available in the user group list when system user groups and users are shown.

User groups / Users
User groups to which the users from the list are assigned to or users which are assigned to user
groups in the user group list.

Libraries
Names of application libraries which are assigned to the users shown.

Is only available in the user group list when users are shown.

User specific information
User specific information for the displayed users in the user list will be shown.

Is only available in the user group list when users are shown.

Date of last login
> Date of last ADONIS login for the respective user.

> Is only available in the user list.

1.4.2 Sorting criteria for the user list

When pressing the button "View" a menu (lower part) is displayed to select the sort criterion for the user list (see fig. 69).

Figure 69: Selecting the sort criterion

The current sort criterion is marked.

Click on the required sort criteria to sort the user list accordingly.

> **Hint:** The available sort criteria will depend on the information shown in the user list, i.e. you can sort the user list by all the information shown.

1.4.3 Add new ADONIS users

New ADONIS users should be added directly after installing the system. During installation a standard user is created (user name "Admin", password "password"). This standard user is equipped with all administration rights. Thus, we advise you to create some or all ADONIS users without administration rights directly following the installation.

> **ATTENTION:** After ADONIS and at least one ADONIS database has been installed, you can only assign the ADONIS standard application library (see chap. 20., p. 693) to the ADONIS users. If you wish to assign a user-defined ADONIS application library (= ADONIS-AL) (see chap. 2., p. 125) to the ADONIS users, this must first be imported (see chap. 2.3.1, p. 282), **before** you create the ADONIS users. By changing the user settings (see chap. 1.4.6, p. 100) you can assign a different ADONIS-AL to an ADONIS user at any time. Afterwards, however, this ADONIS user can no longer access the models, which he created with the ADONIS-AL formerly assigned to him

> **ATTENTION:** After ADONIS and at least one ADONIS database has been installed, you can only assign the ADONIS user to the **standard user group "ADONIS"**. If you wish to assign the ADONIS user to a different ADONIS user group, you have to create this user group (see chap. 1.5.2, p. 107), **before** you add the ADONIS user. By changing the user settings (see chap. 1.4.6, p. 100) you may assign an ADONIS user to one or more other ADONIS user groups at any time.

You can add ADONIS users by opening the window "user list" (see fig. 67) and clicking on the button "Add". On your screen the window "Add new user - Settings" (see fig. 70) appears.

Figure 70: Add new user - Settings

Before adding an ADONIS (system) user you must create or assign the following settings:

- user name and password (see chap. 1.4.3.1, p. 93) (obligatory)

- application library (see chap. 1.4.3.2, p. 93) (obligatory)

- rights (see chap. 1.4.3.3, p. 93) (optional)

- user groups (see chap. 1.4.3.4, p. 93) (obligatory)

- access to components (see chap. 1.4.3.5, p. 94) (optional)

- user-specific information (see chap. 1.4.3.6, p. 95) (optional)

After having entered the necessary information, click on the button "Add" in order to create the ADONIS user. A window will then appear, informing you that the user has been stored in the ADONIS database (see fig. 71).
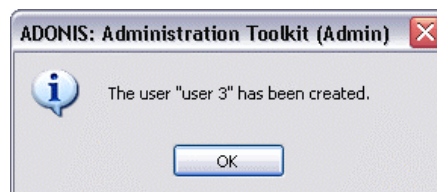
Figure 71: Successfully add user

Click on the OK button or press the "Enter" key. This will close the information window and show the window "Add new users - settings". The entries in the fields for the user name and the password will be deleted, the user settings (see chap. 1.4.6, p. 100) (access rights, application library assigned, ADONIS user groups assigned) of the user just created will be left so that you can easily create another ADONIS user with the same settings.

Finish adding new ADONIS users by clicking on the "Cancel" button. The window "Add new users - settings" will be closed and the current user list (see chap. 1.4, p. 89) will be shown.

1.4.3.1   Enter user name and password

Enter the user name into the field "user name" and the password into the fields "password" and "password (confirmation)".

When you enter the password, asterisks ("*") will be shown instead of the characters entered. You must enter the password a second time to confirm it.

> **Hint:** The user name and the password must consist of at least three characters. The system distinguishes between capital and small letters. You may not enter quotation marks. (").

> **ATTENTION:**   The characters permitted for the password are only the numbers from **0 to 9**, the letters from **a to z** and **A to Z**, **Blanks** and the symbols **! # $ % & ( ) * + , - . / : ; < = > ? @ [ ] ^ _ { | } ~**. All other characters are strictly prohibited.

1.4.3.2   Selecting application libraries

You must assign exactly one application library to each ADONIS user, selecting the application library chosen by a mouse-click from the list "application libraries".

1.4.3.3   Defining rights

Define a user's access rights by clicking on the ADONIS component (Administration Toolkit (see chap. 3., p. 16) and/or Business Process Management Toolkit (see chap. 4., p. 18)).

> **Hint:** You may restrict the access rights to the Business Process Management Toolkit by defining the component access to the modules accordingly.

1.4.3.4   Assigning user groups

You can assign each ADONIS user to one or more ADONIS user groups. To do this, click on the button "User group". The window "<User name> - User groups" (see fig. 72) will appear.
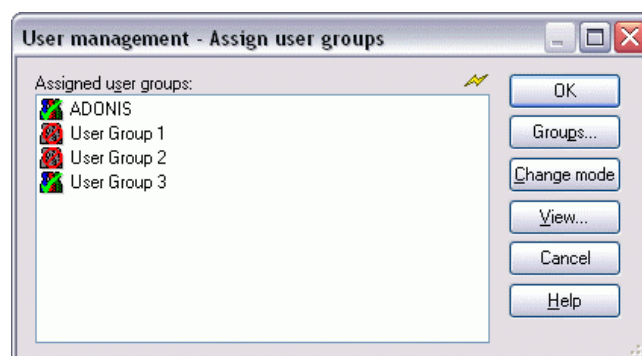
Figure 72: Assign user groups

The list "User groups assigned" contains all ADONIS user groups already defined. The icon preceding the name of the ADONIS user group shows, whether the current ADONIS user is assigned to this ADONIS user group 🡆 or not 🡆.

Assign the ADONIS user by double-clicking the user group or clicking on the icon to the user group you wish to assign the user.

Alternatively, you can assign user to several user groups in one step by selecting the user groups you wish and clicking on the **button "Change"** which then changes the assignment.

If you click on the **button "Groups"**, the user administration window "User group list" appears (see fig. 86) and you can Add user groups (see chap. 1.5.2, p. 107).

By clicking on the **button "View"** you can sort the user groups shown alphabetically ("Sort by user group name") or by their status of assignment, i.e. firstly all groups (alphabetically) () the user is not assigned to and subsequently all groups (alphabetically) () the user has been assigned to ("Sort by assignment").

<u>**Change user group assignment of several users**</u>

If you want to change the settings of several ADONIS users click in the window "<*User names*> - Change user settings " (see fig. 83) on the button "User groups", the window "<*User names*> - User group assignment" (see fig. 73) will appear.
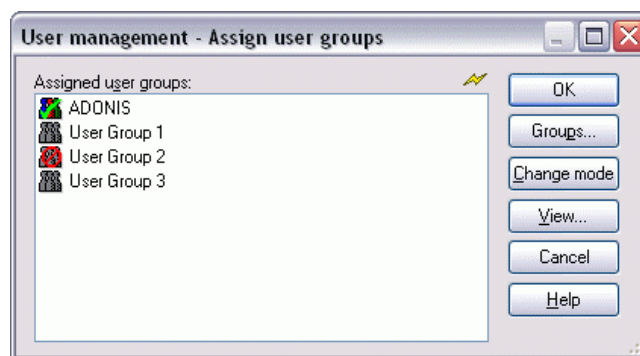
Figure 73: Assign several users to user groups

The list "Assigned user groups" contains all the ADONIS user groups already defined. The icon infront of the name of the ADONIS user group indicates whether the selected ADONIS users of this ADONIS user group are assigned or not or whether the assignment of the selected users is different .

By clicking on the **button "View"** you can sort the user groups shown alphabetically ("Sort by user group name") or by their status of assignment, i.e. firstly all groups (alphabetically) () with different assignment, then all groups (alphabetically) (), with the users that are not assigned, and finally all groups (alphabetically) (), containing users that have been assigned ("Sort by assignment").

The assignment of the ADONIS user groups is done by double-clicking on the appropriate user group or by clicking on the icon or through selecting several user groups and clicking on the button "Change".

1.4.3.5   Define component access

You may restrict a user's access to components so that not all functions of the Business Process Management Toolkit are available. Defining component access is done in the window "<*user name*> - Component Access" (see fig. 74).
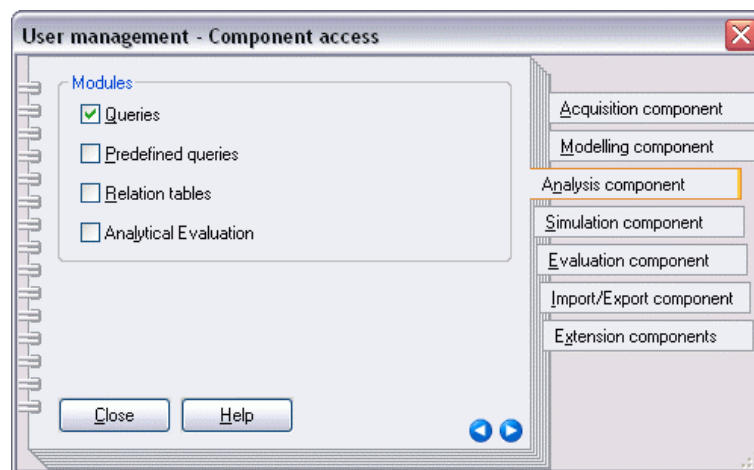
Figure 74: Define component access (Example analysis)

Define the component access by selecting the particular module (see chap. 5.2.2, p. 383) within a chapter. You could for instance provide the ADONIS user "User 2" with only the module "Query" in the Analysis Component (see fig. 74) .

> **Hint:** The access to components can also be defined for user groups. In this case the access rights of the ADONIS user will be taken over by this user group.

> **Hint:** Defining the component access is only possible, if the ADONIS user has access to the Business Process Management Toolkit.

1.4.3.6 Store user-specific information

In the field "User-specific information" you may enter information about the ADONIS user, if necessary.

1.4.3.7 Check list

Check list for adding new ADONIS users:

Settings	Entry	Comment
User name	mandatory	At least three characters; quotation marks (**"**) are forbidden.
Password	mandatory	At least three characters; permitted only are the numbers **0 to 9**, the letters **a to z** and **A to Z**, **blanks** and the symbols **! # $ % & ( ) * + , - . / : ; < = > ? @ [ ] ^ _ {	} ~**.
Password (Confirmation)	mandatory	see above
Application library	optional	If you do not select an application library, the pre-set "ADONIS BPMS Application Library" will be assigned.
Rights	optional	If you do not define the rights, the new ADONIS user will only have access to the Business Process Management Toolkit.
User groups	mandatory	The new ADONIS user must be assigned to at least one ADONIS user group.

Settings	Entry	Comment
Component access	optional	Access to the various components available can be restricted.
User-specific information	optional	Arbitrary text (free text) up to 250 characters.

Table 1: Check list for adding new ADONIS users

ATTENTION: You have just created an ADONIS user and assigned him to an ADONIS user group. Now you must provide read or write access to a model group which either already exists or must also be defined. You can do this in the Model Management component of the ADONIS user group to which you have assigned the new ADONIS user. Otherwise the ADONIS user cannot create or edit ADONIS models.

1.4.4 Import system user

Import new system users by clicking on the button "Add" in the window "User list" (see fig. 67) and select from the menu "Import system user" and proceed with the following steps:

1. Select system domain (see chap. 1.4.4.1, p. 96)

2. Select system user group (see chap. 1.4.4.2, p. 97)

3. Select system user (see chap. 1.4.4.3, p. 97)

4. Import system user (see chap. 1.4.4.4, p. 98)

   Hint: To import several system users with system user groups you have to use the function "Reconcile system user groups" (see chap. 1.5.4, p. 108).

1.4.4.1 Select system domain

Within the window "User management - Select system domain" (see fig. 75) all registered system domains are listed.

Figure 75: Import system users - Select system domain

Select the domain you want to import the system users from. Additionally you can choose one of the following options for the selection of the system user groups:

Selected user groups only (default setting)
Only the existing system user groups are identified and you can select certain user groups (see chap. 1.4.4.2, p. 97).

All user groups and users

All user groups and the users within these groups are identified and you can select the system users (see chap. 1.4.4.3, p. 97) for importing.

Hint: Identifying **all** system user groups and users can lead to a long wait when there exists a large number of groups and users.

Click on the button "Next" to continue.

1.4.4.2 Select system user groups

In the window "Select system user group" (see fig. 76) all system user groups of the selected domains are listed.
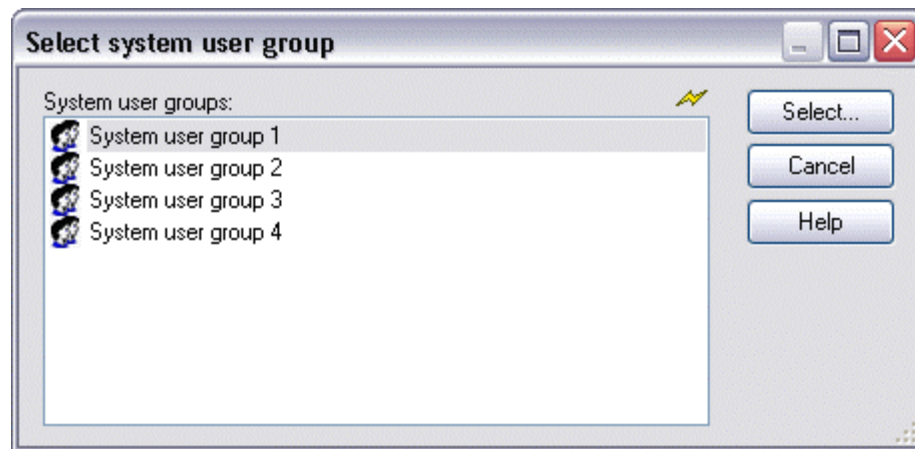
Figure 76: Import system users - select system user group

Select those system user groups from which you want to import the system users.

Click "Next" to select system users (see chap. 1.4.4.3, p. 97) from the chosen groups.

1.4.4.3 Select system user

Within the window "User management - Select system user" (see fig. 77) all available system users (in the system user group structure) in the current domain are listed.
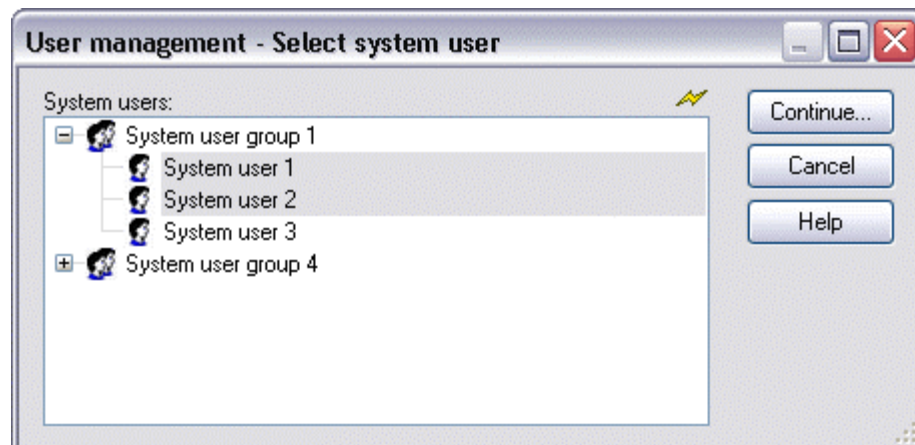
Figure 77: Import system users - Select system user

Select the users you want import and click the button "Next" to import system users (see chap. 1.4.4.4, p. 98).

1.4.4.4 Import system user

Within the window "User management - Import system user" (see fig. 78) all system users to be imported are listed.
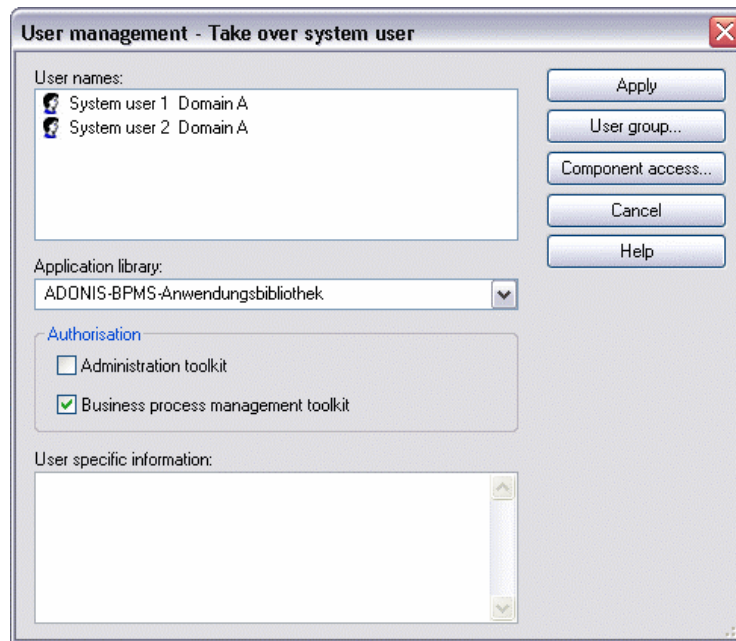
Figure 78: Import system user

To import system users into the ADONIS user management you will have to make the following ADONIS specific settings:

- Application library (see chap. 1.4.3.2, p. 93) (obligatory)
- Access rights (see chap. 1.4.3.3, p. 93) (optional)
- User group (see p. 98) (obligatory)
- Component access (see chap. 1.4.3.5, p. 94) (optional)
- User-specific information (see chap. 1.4.3.6, p. 95) (optional)

After having provided the information click on the button "Assign" to import the system user. After the successful import of the user into the ADONIS database an information message will be displayed and the user list (see fig. 67) will be updated accordingly.

Assign system user groups

You can assign a system user to one or several ADONIS system user groups. This is done in the window "User management - User group assignment" (see fig. 79).
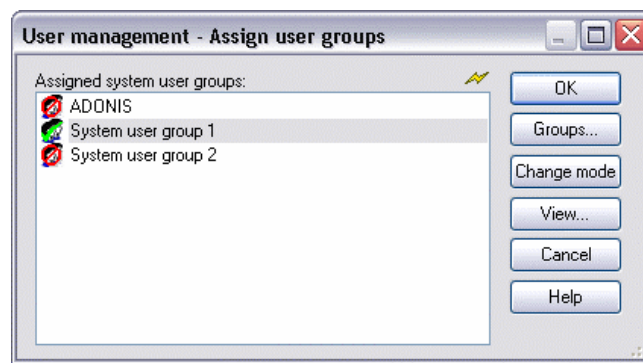
Figure 79: Assign system user groups

The list of "Assigned user groups" contains all system user groups currently defined. The icon infront of the name of the system user group indicates whether the current system user is assigned to this system user group   or not   .

You can assign user groups to the selected system users by double-clicking on the user group or clicking on the icon.

> **Hint:** System users can only be assigned to system user groups. They cannot be assigned to ADONIS user groups.

Alternatively, you can assign a user to several user groups in one step by selecting the user groups you wish and clicking on the **button "Change"** which then changes the assignment.

If you click on the **button "Groups"**, the user administration window "User group list" appears (see fig. 86) and you can Add system user groups (see chap. 1.5.3, p. 108).

By clicking on the **button "View"** you can sort the user groups shown alphabetically ("Sort by user group name") or by their status of assignment, i.e. firstly all groups (alphabetically) ( ) the user is not assigned to and subsequently all groups (alphabetically) ( ) the user has been assigned to ("Sort by assignment").

Change user group assignment of several system users

If you want to change the settings of several system users click in the window "User management - Change user settings " (see fig. 83) on the button "User groups", the window "User management - User group assignment" (see fig. 73) will appear.
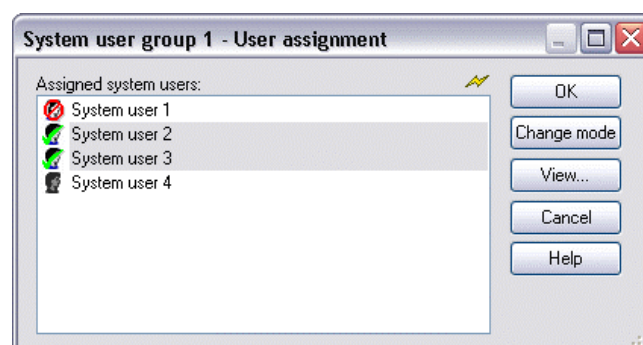
Figure 80: Assign several system users to user system groups

The list "Assigned user groups" contains all the ADONIS user groups already defined. The icon infront of the name of the ADONIS user group indicates whether the selected ADONIS users of this ADONIS user group are assigned ![icon] or not ![icon] or whether the assignment of the selected users is different ![icon].

By clicking on the **button "View"** you can sort the user groups shown alphabetically ("Sort by user group name") or by their status of assignment, i.e. firstly all groups (alphabetically) (![icon]) with different assignment, then all groups (alphabetically) (![icon]), with users that are not assigned, and finally all groups (alphabetically) (![icon]), containing users that are assigned ("Sort by assignment").

The assignment of the system user groups is done by double-clicking on the appropriate user group or by clicking on the icon or through selecting several user groups and clicking on the button "Change".

1.4.5   Reconcile system users

By reconciling system users the names of system users imported into ADONIS are compared with those of the operating system. If the name of a users (login name) has been changed by the operating system administrator this change is also done in the ADONIS user management ensuring that the login name in the operating system equals the ADONIS system user name.

> **Hint:** The login of a system user to ADONIS is still possible when the login name in the operating system has been changed, as the identification of the user takes place using his GUID (=Global Unique Identifier; which is a world-wide unique name used to register system users in domains) and not the login/user name.

Reconcile system users by clicking on the button "Reconcile" in the window "User list" (see fig. 67).

The reconciliation will be performed and a corresponding message will be displayed.

1.4.6   Change ADONIS user settings

Open the "User list" (see chap. 1.4, p. 89) (Menu "User" - option "User list" or ![icon]), to change the settings of ADONIS users stored in the ADONIS database.

> **ATTENTION:**   Should the ADONIS user whose settings you wish to change be logged in (icon with "red traffic light"), the changes will be effective only when the ADONIS user logs out and logs in again.
>
> For the standard user "Admin" you can only change the password (see chap. 7., p. 547). To do this, you must be logged in as user "Admin" in ADONIS.

Select the user whose settings you wish to change in the window "User list" (see chap. 1.4, p. 89) by mouse-clicking on the user's name. Then click on the button "Edit" or double-click on the user's name.

The window "User management - Change user settings" (see fig. 81) will appear on your screen, showing the user name, the user rights for the ADONIS components (Administration Toolkit (see chap. 3., p. 16) and/or Business Process Management Toolkit (see chap. 4., p. 18)), the ADONIS application library assigned to the user selected and possibly entered user-specific information.

> **Hint:** The password of the ADONIS user is shown encrypted and remains unchanged, as long as no new password has been entered.

Figure 81: ADONIS - Change user settings

You may now change the following user settings:

- Password (see chap. 1.4.3.1, p. 93)

- Application library (see chap. 1.4.3.2, p. 93)

- Rights (see chap. 1.4.3.3, p. 93)

- User group (see chap. 1.4.3.4, p. 93)

- Component access (see chap. 1.4.3.5, p. 94)

- User-specific information (see chap. 1.4.3.6, p. 95)

Having made the changes, click on the button "Change" in order to save the user settings after a security query in the ADONIS database.

1.4.7   Change system user settings

Open the "user list" (see chap. 1.4, p. 89) (Menu "User" - "User list" or click on 👥) in order to change the settings of system users stored in the ADONIS database.

> **ATTENTION:**   Should the ADONIS user whose settings you wish to change be logged in (icon 👥 with "red traffic light"), the changes will be effective only when the ADONIS user logs out and logs in again.
>
> For the standard user "Admin" you can only change the password (see chap. 7., p. 547). To do this, you must be logged in as user "Admin" in ADONIS.

Select the user whose settings you wish to change in the window "User list" (see chap. 1.4, p. 89) (see fig. 67)by mouse-clicking on the user's name. Then click on the button "Edit" or double-click on the user's name.

The window "User management - Change user settings" (see fig. 82) will appear showing the system user name (with the domain), the user rights to access the ADONIS components (Administration Toolkit (see chap. 3., p. 16) and/or Business Process Management Toolkit (see chap. 4., p. 18)), the

ADONIS application library assigned to the user selected and optionally entered user-specific informa-
tion.
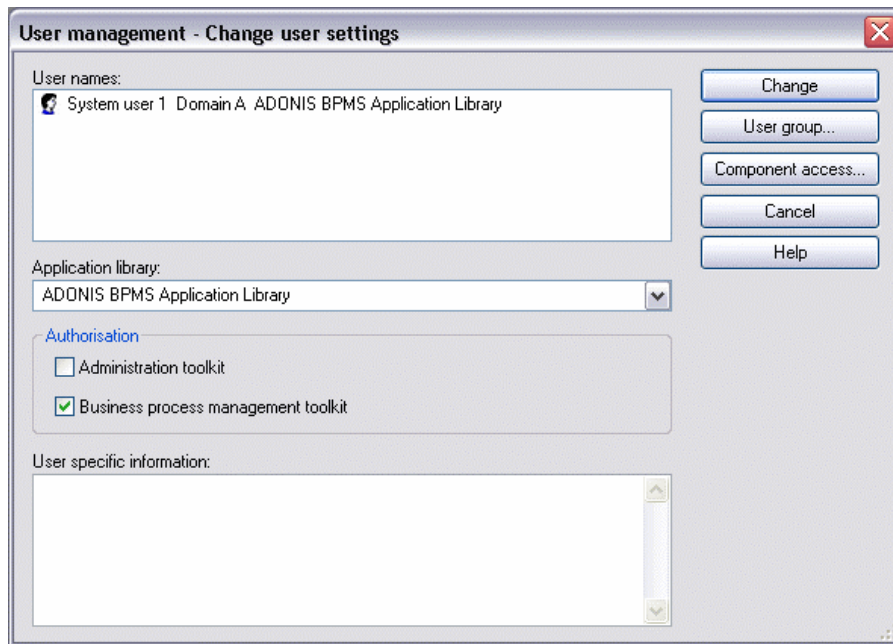
Figure 82: System - change user settings

You may now change the following user settings:

● Application library (see chap. 1.4.3.2, p. 93)

● Rights (see chap. 1.4.3.3, p. 93)

● User group (see chap. 1.4.3.4, p. 93)

● Component access (see chap. 1.4.3.5, p. 94)

● User-specific information (see chap. 1.4.3.6, p. 95)

Having made the changes, click on the button "Change" in order to save the user settings after a secu-
rity query in the ADONIS database.

1.4.8   Simultaneously editing several users' settings

You may change the settings for several users simultaneously by selecting the ADONIS and/or sys-
tem users you require in the User list (see chap. 1.4, p. 89) and then click on the button "Edit". The
window "User management - Change user settings" (see fig. 83) appears.

> **Hint:** If one or more ADONIS users from the group chosen is logged in, a security message
> will appear informing you that the changes will be effective only when the ADONIS user
> logs out and logs in again
>
> Clicking on the "Yes" button causes the specified ADONIS user to be updated with the
> changes. Clicking on "All" does the same for all of the users logged in and selected. A
> click on the "No" button removes the particular ADONIS user from the group
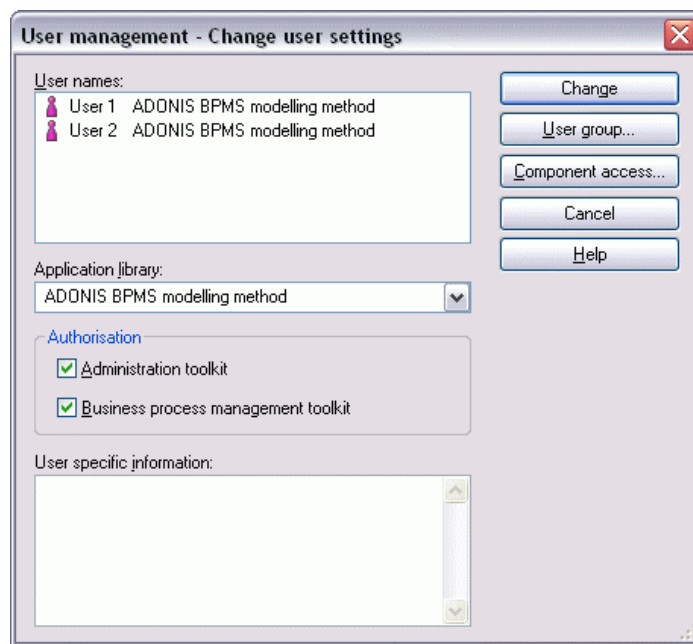
Figure 83: Change users' settings of several users at the same time

The list "User names" shows all the previously selected users with the application libraries assigned to them. For system users their domain will be shown additionally.

> **Hint:** Here you can change all user settings - except the passwords - for several ADONIS and/or system users.
>
> When changing **settings of ADONIS users and system users at the same time** it is not possible to **change the user group assignment** as ADONIS users can only be assigned to ADONIS user groups and system users can only be assigned to system user groups.

■ or the text "<Keep assigned library>" (in the field "Application libraries") or "<Keep assigned text>" (in the field "User-specific information") indicates that previously selected users have different settings.

You may change the following user settings:

- Application library (see chap. 1.4.3.2, p. 93)
- Rights (see chap. 1.4.3.3, p. 93)
- ADONIS user group (see chap. 1.4.3.4, p. 93) or system user group (see p. 98)
- Component access (see chap. 1.4.3.5, p. 94)
- User-specific information (see chap. 1.4.3.6, p. 95)

Having made the changes, click on the button "Change", and confirm the security query to save the change.

1.4.9   Delete an ADONIS user

As administrator you may at any time delete one or more of the ADONIS and/or system users from the ADONIS database.

> **Hint:** You can only delete an ADONIS and/or system user, if he is not currently logged in. The icon 🔼 respectively 🔼 preceding the user name shows, that this ADONIS user is not logged in at the moment ("green traffic light").

The standard user "Admin" **cannot** be deleted.

To delete one or more users open the user list (menu "User" - option "user list") and mark the user or users to be deleted. Then click on the button "Delete".

A security message for each selected user will appear, asking whether you really want to delete this user.

Delete the user by clicking on the "Yes" button or pressing the "Enter" key. If you want to stop the security queries in case of multiple selections for each single user, click on the "Yes, all" button.

If you do not want to delete the user, click on the "No" button. If you want to answer the security query with "Yes" for all selected users, click on the "All" button. Close the window which appears by clicking on the OK button or by pressing the "Enter" key.

The window "User list" (see fig. 67) will be shown with an updated list of users.

> **Hint:** The users will only be removed from the user list, i.e. the users will still be stored in the ADONIS database. To delete users permanently from the ADONIS database you have to remove them from the list of deleted users (see chap. 1.4.9.1, p. 104).

1.4.9.1 List of deleted users

The list of deleted users contains all users which have been deleted from the user list and therefore are no longer valid (i.e. these users can no longer log into ADONIS).

Open this list by opening the window "User list" (see fig. 67) and clicking on the button "Deleted users".
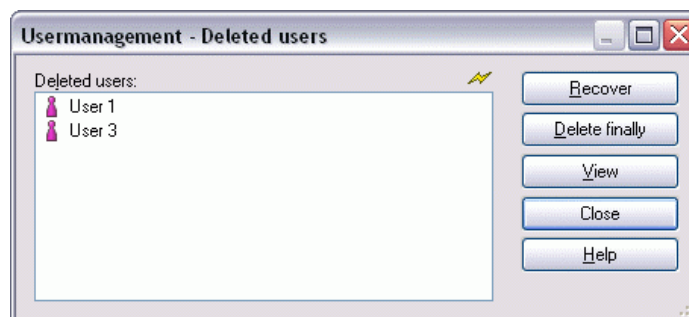
Figure 84: List of deleted users

The ADONIS users deleted from the user list are identified by the icon 👤 .

When using the Single-Sign-on functionality deleted system users will be identified by the icon 👤 .

By clicking on the **button "Restore"** the previously selected users can be restored to the user list.

For each selected user a security message will be displayed.

Restore the user by clicking on the "Yes" button or pressing the "Enter" key. If you want to stop the security queries in case of multiple selections for each single user, click on the "Yes, all" button. If you do not want to restore the user, click on the "No" button.

> **Hint:** During UDL import (see chap. 1.6, p. 116) - when applying the import option "**Update existing user**" or "**Overwrite existing user**" - each ADONIS or system user within the list of deleted users will be restored to the user list (see chap. 1.4, p. 89) automatically.

Clicking on the **button "Delete permanently"** you can permanently delete all previously selected users from the ADONIS database.

For each selected user a security message will be displayed.

Delete the user permanently from the ADONIS database by clicking on the "Yes" button or pressing the "Enter" key. If you want to stop the security queries in case of multiple selections for each single user, click on the "Yes, all" button. If you do not want to delete the user, click on the "No" button.

By clicking the **button "View"** you can view (see chap. 1.4.1, p. 90) additional information for the user or select (see chap. 1.4.2, p. 91) the sort criterion for the users displayed.

1.4.10 Show users not assigned to user groups (user pool)

These type of users are not assigned to any user group and therefore have no access to ADONIS models.

To determine which users have not been assigned to a group activate the "User pool" by selecting **"Non-assigned users"** from the context menu (right mouse click). Within the window "User management - User list" (see fig. 85) the field "Non-assigned users" will be shown.
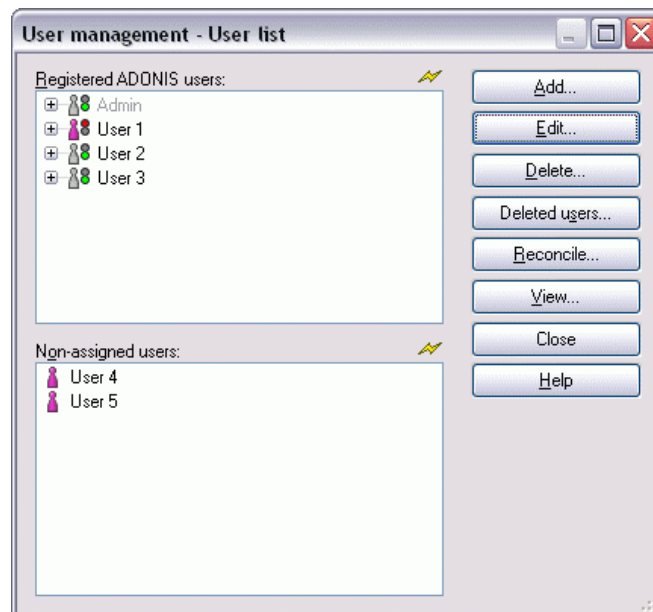
Figure 85: Show users not assigned (user pool)

The ADONIS users not assigned can be re-assigned to an ADONIS user group by editing the ADONIS user settings (see chap. 1.4.6, p. 100).

The system users not assigned can be re-assigned to a system user group by editing the system user settings (see chap. 1.4.7, p. 101).

By selecting "Delete users" (see chap. 1.4.9, p. 103) the selected and unassigned users will be deleted from the ADONIS user list.

1.5 User group list

The user group list is the starting point for the administration of the ADONIS user groups (see fig. 66). The user group list contains all ADONIS user groups defined as well as all ADONIS users assigned to these user groups.

> **Hint:** Both the standard user "Admin" and the standard ADONIS user group "ADONIS" in which he is assigned are automatically created whenever an ADONIS database is created. This assignment cannot be changed.

> **Hint:** The standard system user group "ADONIS" is automatically created when a ADONIS database WITH Single-Sign-on functionality is created.

Select the option "User group list" in the "User" menu or click on the corresponding smart-icon 👥 in the quick-access panel to display all the ADONIS user groups stored in the ADONIS database.

The window "User management - User group list" (see fig. 86) appears on your screen. Here, the ADONIS user groups 👥 are shown in alphabetical order.

When using Single-Sign-on functionality the system user groups 🐾 are shown in alphabetical order.
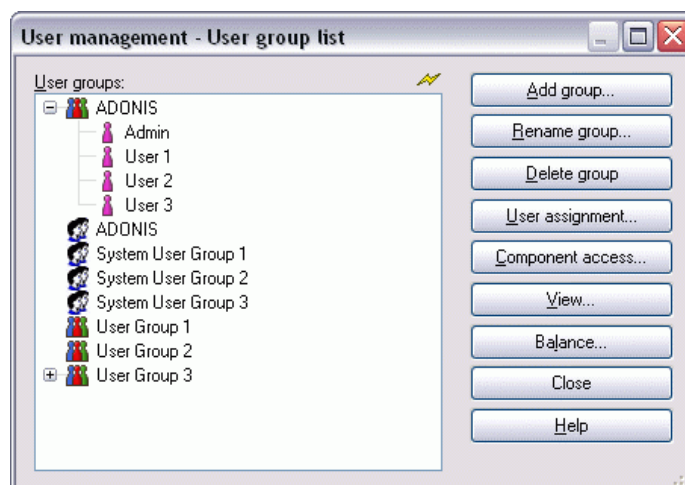
Figure 86: User group list (with system user groups)

The following buttons are available when viewing the user group list:

"Add group" — add a new ADONIS user group (see chap. 1.5.2, p. 107) or a new system user group (see chap. 1.5.3, p. 108) ;

"Rename group" — change the name of the ADONIS user group (see chap. 1.5.5, p. 112) or system user group (see chap. 1.5.6, p. 112) previously selected;

"Delete group" — delete (see chap. 1.5.7, p. 112) a previously selected ADONIS or system user group from the user group list ;

"User assignment" — change the user assignment for the previously selected ADONIS user group (see chap. 1.5.8, p. 113) or system user group (see chap. 1.5.9, p. 114)

"Component access" — restrict the access of ADONIS or system user groups to certain components (see chap. 1.5.10, p. 115)

"View"	show (see chap. 1.4.1, p. 90) additional information for the users or select (see chap. 1.5.1, p. 107) the sort criteria for the listed user groups ;
"Reconcile"	reconcile (see chap. 1.5.4, p. 108)system user groups and their system users in the ADONIS user management with those from the operating system's user administration ;
"Close"	close the window "User list".

1.5.1   Sorting criteria for the user group list

On clicking the button "View", a menu (lower part) for the selection of the user group list sorting criterion is displayed (see fig. 87).
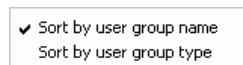
Figure 87: Selecting the sorting criterion

The active sorting criterion is marked with a check mark.

Click on the desired criterion to sort the user list

- by user group name or

- user group type.

When **sorting by user group name** all user groups - ADONIS and system user groups together - are shown in alphabetical order.

When **sorting by user group type** first the ADONIS user groups appear in alphabetical order followed by the system user groups (again in alphabetical order).

1.5.2   Add user group

Add a new ADONIS user group in the window "Add user group" (see fig. 88).

Figure 88: Create user group

Enter in the field "New user group name" a name that has not been used for any of the existing ADONIS user groups. Then click on the OK button or press the "Enter" key to create the new ADONIS user group.

> **Hint:** The name for the new ADONIS user group must be unique, i.e.it must not yet have been used for any other existing ADONIS user group.

Once the new ADONIS user group has been added, the updated user group list will be displayed.

1.5.3   Add system user group

Add a new system user group in the window "Add system user group" (see fig. 89).

Figure 89: Create system user group

Enter into the field "New system user group name" a name that has not been used for any of the existing system user groups. Then click on the OK button or press the "Enter" key to create the new system user group.

> **Hint:** The name for the new system user group must be unique, i.e.it must not yet have been used for any other existing system user group.

Once the new system user group has been added, the updated user group list will be displayed.

1.5.4   Reconcile system user groups

By reconciling system user groups the system user groups and the system users assigned to them in the ADONIS user management are compared with the operating system user groups.

As a result the differences are listed and actions are suggested to represent the operating system user groups and their system users within the ADONIS user administration.

The reconciliation therefore allows to take over system user groups with their system users into the ADONIS user management.

Reconcile system user groups by clicking on the button "Reconcile" in the window "User group list" (see fig. 86) and taking the steps described below:

1. Select system domain (see chap. 1.5.4.1, p. 108)

2. Select system user groups (see chap. 1.5.4.2, p. 109)

3. Reconcile system user groups (see chap. 1.5.4.3, p. 110)

4. Perform reconciliation (see chap. 1.5.4.4, p. 110)

5. Adopt settings for new system users (see chap. 1.5.4.5, p. 111)

> **Hint:** In order to import single users into the ADONIS user management use the function "Import system users" (see chap. 1.4.4, p. 96)

1.5.4.1   Select system domain

Within the window "User management - Select system domain" (see fig. 90) all registered system domains are listed.
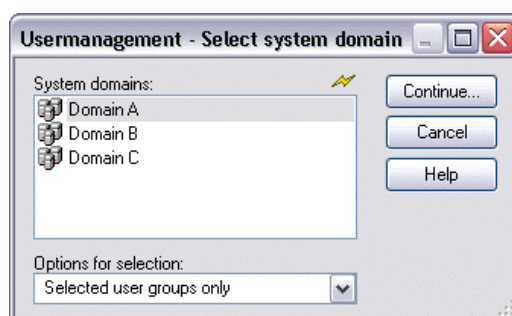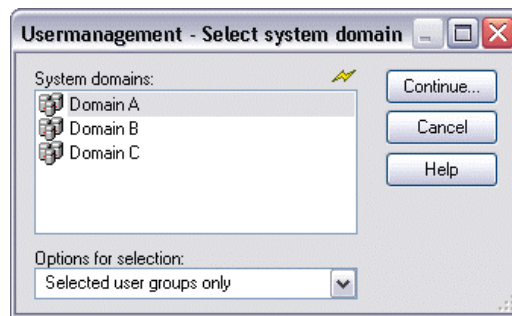
Figure 90: Reconcile system user groups - Select system domain

Select the domain you want to import the system users from. Additionally you can choose one of the following options for the selection of the system user groups:

Selected user groups only (default setting)
> Only the existing system user groups are identified and you can select certain user groups (see chap. 1.4.4.2, p. 97) for reconciliation.

All user groups and users
> All user groups and the users within these groups are identified and you can select the system users (see chap. 1.4.4.3, p. 97) for reconciliation.

> **Hint:** Identifying **all** system user groups and users can lead to a long wait when there exists a large number of groups and users.

Click on the button "Next" to continue.

1.5.4.2 Select system user groups

In the window "Select system user group" (see fig. 91) all system user groups of the selected domains are listed.
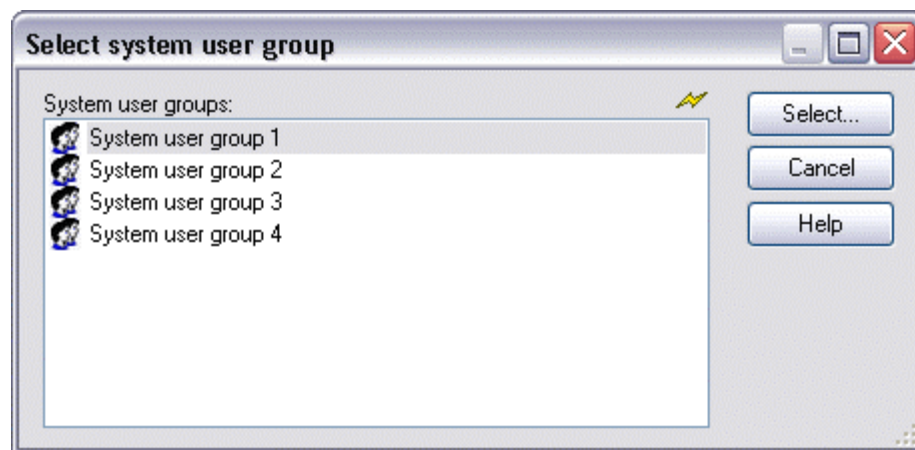
Figure 91: Reconcile system user groups - select system user group

Select those system user groups from which you want to import the system users.

Click on the button "Next" to reconcile the selected system user groups (see chap. 1.4.4.3, p. 97).

1.5.4.3 Reconcile system user groups

After loading the operating system's user groups the window "Reconcile user groups" will be shown (see fig. 92).
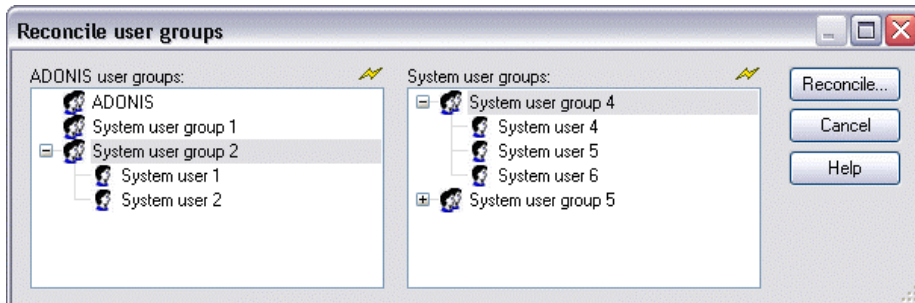
Figure 92: Reconcile system user groups - Select user group

Within the list "ADONIS user groups" all system user groups of the ADONIS user management are listed while in the list "System user groups" all user groups of the operating system are shown.

Select the user groups to reconcile and click on the button "Reconcile" to start and obtain possible actions (see chap. 1.5.4.4, p. 110).

1.5.4.4 Perform reconciliation

In the window "Reconcile user groups - planned actions" the result of the user group comparison is shown (see fig. 93).
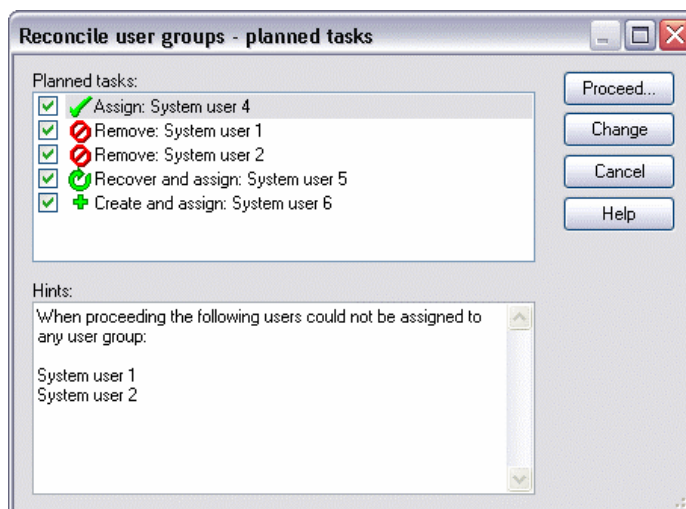
Figure 93: Reconcile system user groups - Select action

Within the list **"Planned actions"** the system users from the user groups compared are shown. In addition you will receive information on which additional action has to be performed in order to fully reconcile the operating system user group with the ADONIS user management. The following action is possible:

✔ **Assign** if the system user already exists in the ADONIS user management, but has not been assigned to the previously selected ADONIS system user group.

⊘ **Remove** when system users only exist in the previously selected ADONIS system user group but not within the selected operating system user group.

↻   **Restore and assign** if the system users exist in the ADONIS user management, but have been deleted from the user list (and therefore is only contained in the list of deleted users (see chap. 1.4.9.1, p. 104)).

➕   **Create and assign** if the system user does not exist in the ADONIS user management. The user settings for new users (see chap. 1.5.4.5, p. 111) must be defined after performing the reconciliation.

The actions are performed when they are selected (☑). Activate or de-activate an action by double-clicking on the action or selecting several actions and clicking on the button "Change".

Within the **field "Hints"** information for the different actions will be shown.

To run the selected actions click on the **Button "Assign"**. After the reconciliation has finished an up-dated User group list will be shown (see fig. 86, p. 106).

1.5.4.5   User settings for new system users

Before creating new system users during the reconciliation the window "Reconcile user groups - user settings for new users" will be shown (see fig. 94).
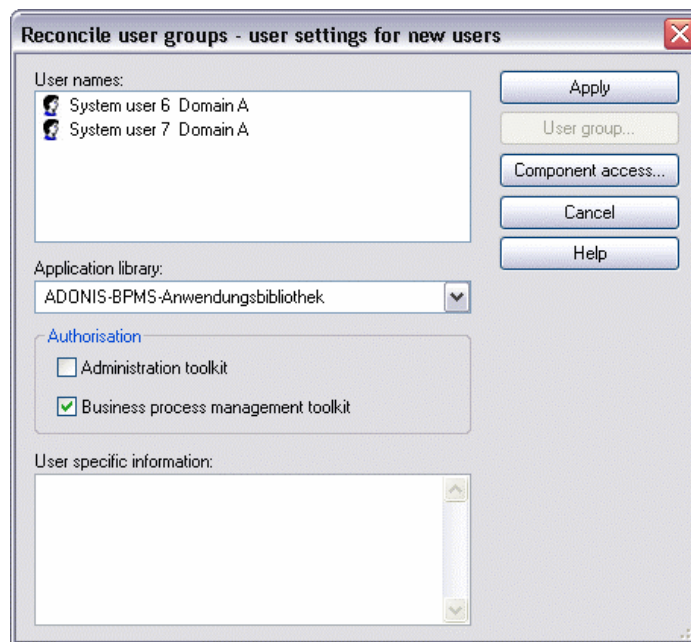
Figure 94: Reconcile system user groups - User settings for new system users

To create system users into the ADONIS user management you will have to make the following ADONIS specific settings:

- Application library (see chap. 1.4.3.2, p. 93) (obligatory)

- Access rights (see chap. 1.4.3.3, p. 93) (optional)

- Component access (see chap. 1.4.3.5, p. 94) (optional)

- User-specific information (see chap. 1.4.3.6, p. 95) (optional)

After having provided the information click on the button "Apply" to import the system user. After the successful import of the user into the ADONIS database an updated User group list will be shown (see fig. 86, p. 106).

1.5.5 Rename user group

Assign a new name to an ADONIS user group in the window "Rename user group" (see fig. 95).

Figure 95: Rename user group

Enter the name of your choice into the field "New user group name", then click on the "OK" button or press the "Enter" key.

> **Hint:** The name of the ADONIS user group has to be unique, i.e. the name must not be assigned to an already existing ADONIS user group.

> **Hint:** The standard user group "ADONIS" cannot be renamed.

1.5.6 Rename system user group

Rename a system user group in the window "Rename system user group" (see fig. 96).

Figure 96: Rename system user group

Enter the name for the system user group in the window "New system user group name", then click on the "OK" button or press the "Enter" key.

> **Hint:** The name of the system user group has to be unique, i.e. the name must not be assigned to an already existing system user group.

> **Hint:** The standard system user group "ADONIS" cannot be renamed.

1.5.7 Delete user group

Delete one (or more) user groups by selecting the groups to be deleted in the window "User group list" (menu "User" - option "User group list") and then clicking on the button "Delete group".

> **ATTENTION:** Note that when you delete a user group, the users assigned to this group will lose their access rights to ADONIS model groups and ADONIS models assigned to this user group.

> **Hint:** The **standard ADONIS user group"ADONIS"** and the **standard system user group "ADONIS"** cannot be deleted.

If a user group has been deleted and a user is no longer assigned to a user group, a security query will be displayed offering to assign this user to the standard user group "ADONIS" (by clicking on the Yes button). By clicking on the "No" button the user will not be assigned to any user group.

Users not assigned can be viewed from the "User list" in the User pool (see chap. 1.4.10, p. 105).

1.5.8   User assignment

Change the user assignment of an ADONIS user group in the window"<*User group name*> - User assignment" (see fig. 97) which shows a list of all ADONIS users stored in the database. The icon preceding the user name informs you, whether this ADONIS user is assigned to the current ADONIS user group or not .
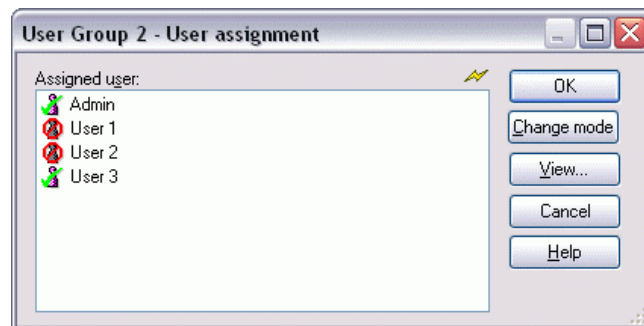
Figure 97: ADONIS - User assignment

Assign the ADONIS users you wish to the ADONIS user group by clicking on the icon.

Alternatively, you can assign several ADONIS users to the ADONIS user group in one step by selecting the users you wish and changing the assignment by clicking on the **button "Change"**.

By clicking on the **button "View"** you can view the application libraries assigned to the users and/or their user-specific information. In addition you can sort the users alphabetically ("Sort by user name") or by their state of assignment ("Sort by assignment"):

1.   all users (alphabetically) (), which are not assigned to the selected groups.

2.   all users (alphabetically) (), which are already assigned to the selected groups.

Change user assignment of several user groups

Edit the user assignment of several ADONIS user groups in the window "<*user group names*> - user assignment" (see fig. 98) with a list of all ADONIS users stored in the database. The icon preceding the user name indicates whether this ADONIS user is assigned to the current ADONIS user group or not or whether the assignment to the selected user groups is different inside the selected user groups .
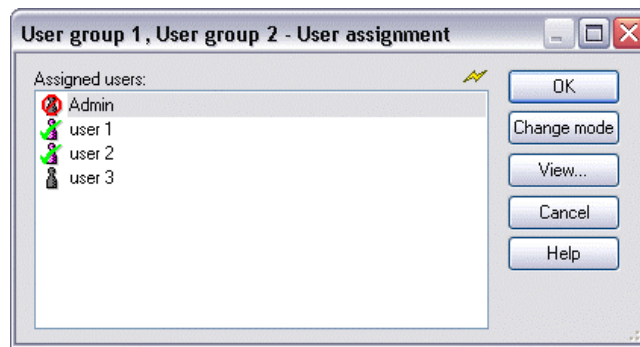
Figure 98: User assignment of several user groups

The assignment of ADONIS users is carried out by double-clicking on the appropriate ADONIS user or by selecting several users and clicking on the **button "Change"**.

By clicking on the **button "View"** you can view the application libraries assigned to the users and/or their user-specific information. In addition you can sort the users alphabetically ("Sort by user name") or by their state of assignment ("Sort by assignment"):

1. all users (alphabetically) ( ) with different assignment.

2. all users (alphabetically) ( ), which are not assigned to the selected groups.

3. all users (alphabetically) ( ), which are already assigned to the selected groups.

> **Hint:** Whenever an ADONIS database is created, the standard user "Admin" will be automatically assigned to the standard user group "ADONIS", from which it cannot be removed.

Finish the user assignment by clicking on the "OK" button.

1.5.9   System user assignment

Change the user assignment of a system user group in the window"*<System user group name>* - User assignment" (see fig. 97) which shows a list of all system users stored in the database. The icon preceding the user name informs you, whether this system user is assigned to the current system user group or not .
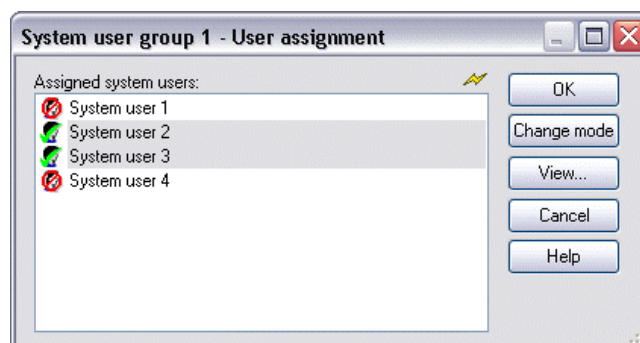
Figure 99: System user assignment

Assign the system users you wish to the system user group by double-clicking or clicking on the icon.

Alternatively, you can assign several system users to the system user group in one step by selecting the users you wish to assign and changing the assignment by clicking on the **button "Change"**.

By clicking on the **button "View"** you can view the application libraries assigned to the users and/or their user-specific information. In addition you can sort the users alphabetically ("Sort by user name") or by their state of assignment ("Sort by assignment"):

1. all users (alphabetically) ( ), which are not assigned to the selected groups.

2. all users (alphabetically) ( ), which are already assigned to the selected groups.

Change user assignment of several system user groups

Edit the user assignment of several system user groups in the window "*<System user group names>* - User assignment" (see fig. 100) with a list of all system users stored in the database. The icon preceding the user name indicates whether this ADONIS user is assigned to the current ADONIS user group or not , respectively if the assignment within the selected user groups is different .
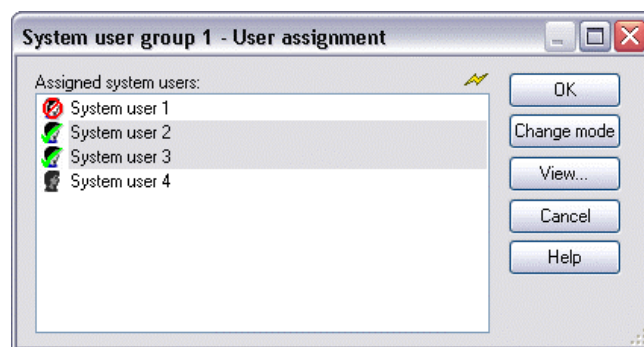
Figure 100: System user assignment of several user groups

The assignment of system users is carried out by double-clicking on the appropriate system user or by selecting several users and clicking on the **button "Change"**.

By clicking on the **button "View"** you can view the application libraries assigned to the users and/or their user-specific information. In addition you can sort the users alphabetically ("Sort by user name") or by their state of assignment ("Sort by assignment") as follows:

1. all users (alphabetically) ( ) with different assignment.

2. all users (alphabetically) ( ), which are not assigned to the selected groups.

3. all users (alphabetically) ( ), which are already assigned to the selected groups.

Finish the user assignment by clicking on the "OK" button.

1.5.10   Define component access

You can define the component access of all ADONIS users of an ADONIS user group/several ADONIS user groups, so that only specific functions of the Business Process Management Toolkit are available. the definition of the component access is carried out in the window "*<User group name>* - Component access" (see fig. 101).
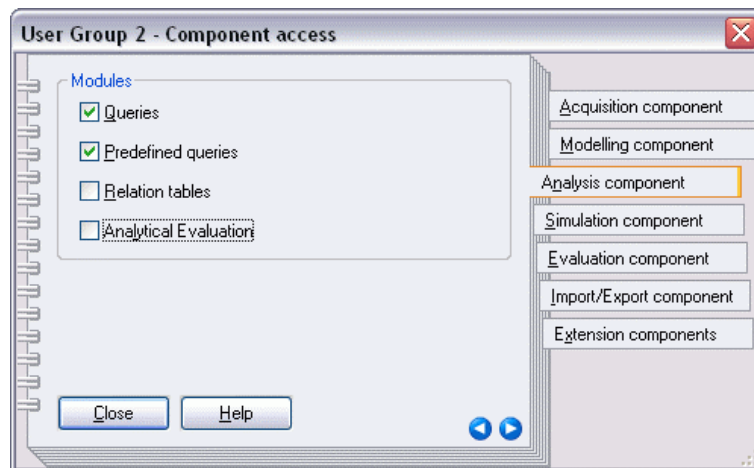
Figure 101: Define component access

Determine the component access by selecting the appropriate module (see chap. 5.2.2, p. 383) of a component (tab).

> **Hint:** The definition of the component access is only possible, if the ADONIS user has an access right to the Business Process Management Toolkit.

> **Hint:** If you want to define the component access for several ADONIS user groups at the same time, various settings will be displayed within this selected user groups through 🔲.

1.6 UDL Import

You can import ADONIS users and/or ADONIS user groups using the option "Import users" in menu "Users" or by clicking on the smart-icon 🔶 in the quick-access bar.

> **Hint:** When you are importing user groups, these user groups' access rights to model groups will also be imported, if you select the option "Rename user group" (see chap. 1.6.12, p. 121) and if the model groups for the respective ADONIS application library are stored in the ADONIS database.

> **Hint:** When using Single-Sign-on functionality you can additionally import system users and/or system user groups from a UDL file.
>
> These system users are users which already exist in the ADONIS user management and which have been exported previously into a UDL file from ADONIS using the UDL export. To import operating system users into the ADONIS user management use the function "Import system user" (see chap. 1.4.4, p. 96) or "Reconcile system user" (see chap. 1.5.4, p. 108).

On calling the function "Import User", the window "User management - User Import" (see fig. 102) appears.
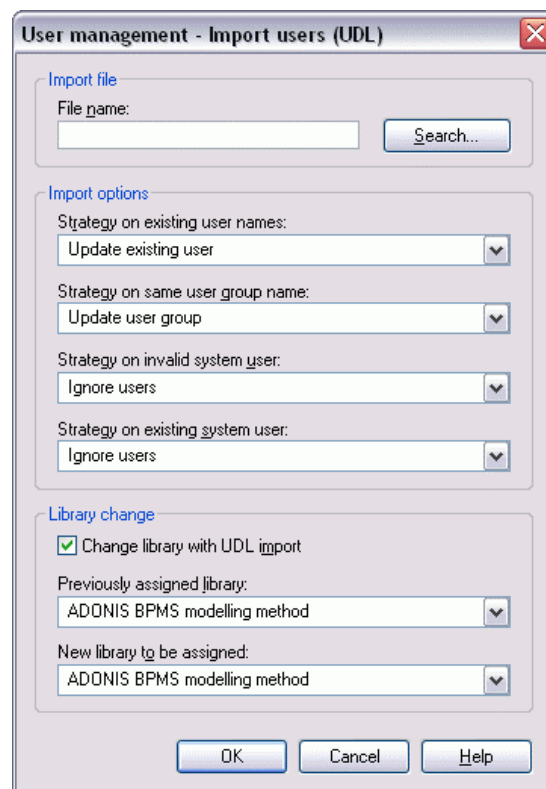
Figure 102: Import user(s) (groups)

Enter the path and the name of the UDL file to be imported into the field *"File name"*.

Should you wish to import users which have the same name as those stored in the ADONIS database you can choose one of the following options from *"Strategy on existing user names"*:

- "**Rename user**" (see chap. 1.6.2, p. 119)

- "**Update existing user**" (see chap. 1.6.3, p. 119)

- "**Overwrite existing users**" (see chap. 1.6.5, p. 120)

- "**Ignore users**" (see chap. 1.6.7, p. 120)

Should you wish to import user groups which have the same name as those stored in the ADONIS database you can choose one of the following options from *"Strategy on same user group name"*:

- "**Rename user group**" (see chap. 1.6.12, p. 121)

- "**Update user group**" (see chap. 1.6.13, p. 121)

- "**Overwrite user group**" (see chap. 1.6.11, p. 121)

- "**Ignore user group**" (see chap. 1.6.10, p. 121)

- "**Adopt users into existing group**" (see chap. 1.6.9, p. 121)

For the import of system users which are not known in the domains or when importing system users into an ADONIS database that doesn't support Single-Sign-on, you can choose from one of the options from *"Strategy on invalid system user"*:

- "**Create as internal user**" (see chap. 1.6.14, p. 122)

- "**Ignore users**" (see chap. 1.6.7, p. 120)

For the <u>import of system users that have the same name as already existing system users in the</u> <u>ADONIS database</u> you can choose from one of the following options from **"Strategy on existing system user"**):

- "**Update existing user**" (see chap. 1.6.4, p. 120)

- "**Overwrite existing users**" (see chap. 1.6.6, p. 120)

- "**Ignore users**" (see chap. 1.6.8, p. 120)

    Hint: The options for **"Strategy on existing system user"** are only available when using Single-Sign-on.

To automatically change users library assignments during the UDL import you can activate the option "**Change library with UDL import"**. Additionally select from the list "Previously assigned library" the library currently assigned to the users to be imported and select from the list "New library to be assigned" the library which will be assigned to the users after the UDL import. This way you can carry out the migration of the users of an application library to a new application library during the UDL import.

    Hint: To change user library assignments, the new assigned library must have previously been imported (see chap. 2.3.1, p. 282).

Start the UDL import by clicking on the "OK" button.

During the user import a status window will inform you on the current state of the import process.

    Hint: If an ADONIS user to be imported is assigned during the UDL import to an application library which is not stored in the ADONIS database, you will be requested to assign an existing application library (see chap. 1.6.15, p. 122).

After the successful UDL import, the result will be displayed (see chap. 1.6.1, p. 118).

    Hint: When you import an ADONIS user, his default password will be the user name.

    An ADONIS user with the user name "Admin" cannot be imported, since this name is reserved for the standard user "Admin".

1.6.1   Result

When the users have been successfully imported, a message window informs you that the import process is finished   (see fig. 103). In addition this window list the names of the imported ADONIS/system users and ADONIS/system user groups stored in the ADONIS database.
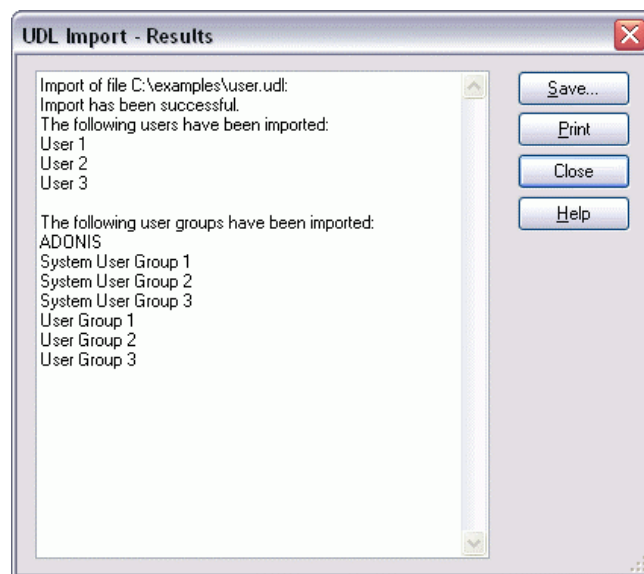
Figure 103: UDL import - Result

> **Hint:** In the result list after the names of the imported system users - separated by ";" - the name of the domain of the respective user will be shown.

1.6.2 Rename ADONIS user

The ADONIS user to be imported will be renamed if the user name already exists in the ADONIS database.

If there is a naming conflict, a query window appears. Click on the "Yes" button, if you wish to assign a different name to the user to be imported. The window "UDL Import - Change Names" (see fig. 104) will appear on your screen.
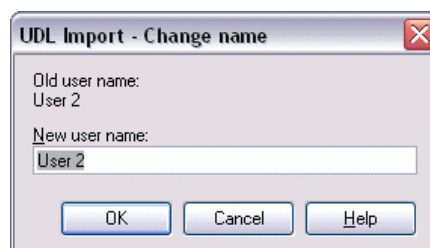
Figure 104: Rename user

Enter a new user name for the ADONIS user to be imported, then click on the "OK" button or press the "Enter" key to continue the UDL import.

1.6.3 Update existing ADONIS user

An ADONIS user stored in the ADONIS database will be updated by the ADONIS user to be imported, i.e. the user settings contained in the UDL file (application library, rights, user group assignment component access, user-specific information) overwrite the settings stored in the ADONIS database.

> **Hint:** If the ADONIS user to be updated is in the list of deleted users (see chap. 1.4.9.1, p. 104) this user will be restored to the user list (see chap. 1.4, p. 89) automatically.

1.6.4   Update existing system user

A system user stored in the ADONIS database will be updated by the system user to be imported, i.e. the user settings contained in the UDL file (application library, rights, user group assignment component access, user-specific information) overwrite the settings stored in the ADONIS database.

> **Hint:** If the system user to be updated is in the list of deleted users (see chap. 1.4.9.1, p. 104) this user will be restored to the user list (see chap. 1.4, p. 89) automatically.

1.6.5   Overwrite existing ADONIS user

An ADONIS user already stored in the ADONIS database will be overwritten by the ADONIS user to be imported. That means all the user settings (application library, rights, user group assignment, component access, user-specific information) will be taken from the UDL file and replace the settings stored in the ADONIS database.

> **Hint:** If the ADONIS user to be overwritten is in the list of deleted users (see chap. 1.4.9.1, p. 104) this user will be restored to the user list (see chap. 1.4, p. 89) automatically.

1.6.6   Overwrite existing system user

A system user already stored in the ADONIS database will be overwritten by the system user to be imported. That means all the user settings (application library, rights, user group assignment, component access, user-specific information) will be taken from the UDL file and replace the settings stored in the ADONIS database.

> **Hint:** If the system user to be overwritten is in the list of deleted users (see chap. 1.4.9.1, p. 104) this user will be restored to the user list (see chap. 1.4, p. 89) automatically.

1.6.7   Ignore ADONIS users

Selecting this option causes ADONIS to ignore users with an already existing user name, i.e. not to import them.

Should you have chosen then options "Ignore user/s" and "Ignore User Group/s" (see chap. 1.6.10, p. 121) and the UDL file contains exclusive ADONIS users and user groups with names identical to those already in the ADONIS database, an appropriate window will appear (see fig. 105).
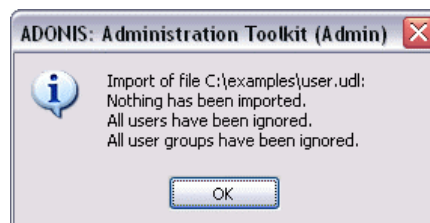
Figure 105: Ignore user

1.6.8   Ignore system users

Selecting this option causes ADONIS to ignore users with an already existing user name, i.e. not to import them.

Should you have chosen then options "Ignore user/s" and "Ignore User Group/s" (see chap. 1.6.10, p. 121) and the UDL file contains exclusive ADONIS users and user groups with names identical to those already in the ADONIS database, an appropriate window will appear (see fig. 105, p. 120).

1.6.9   Adopt users into existing group

Should a user group of the same name already exist in the ADONIS database, users contained in the UDL file and assigned to this user group will be adopted into the existing user group.

1.6.10   Ignore user group

Selecting this option causes user groups with identical names to be ignored, i.e. not to be imported.

Should you have chosen the options "Ignore user group" and "Ignore Users" (see chap. 1.6.7, p. 120) and the UDL file exclusively contains user groups and ADONIS users with names identical to those already in the ADONIS database, an appropriate message window will appear (see fig. 105).

1.6.11   Overwrite user group

A user group stored in the ADONIS database will be overwritten by the user group to be imported. That means the settings will be taken from the UDL file and the existing settings will be deleted.

1.6.12   Rename user group

The user group to be imported will be renamed, should a user group with an identical name already exist in the ADONIS database.

If there is a naming conflict, a query window appears. Click on the "Yes" button, if you wish to assign a different name to the user group to be imported. The window "UDL Import - Change Names" (see fig. 106) will appear on your screen.

Figure 106: Rename user group

Enter a new user name for the user group to be imported, then click on the "OK" button or press the "Enter" key to continue the UDL import.

1.6.13   Update user group

A user group stored in the ADONIS database will be updated by the user group to be imported, i.e. the user group settings contained in the UDL file (application library, rights, user group assignment, component access, user-specific information) overwrite the settings stored in the ADONIS database.

1.6.14 Create as internal user

A system user which is not known in the available domains can be created as an internal user (i.e. as ADONIS user). For the user name and password the *LastDisplayName* from the UDL file will be used. If this user name already exists in the database it will be proceeded according to the settings for **"Strategy on existing user names"** (see chap. 1.6.2, p. 119).

1.6.15 Assign library

The window "UDL Import - Assign Library" (see fig. 107) shows all the application libraries stored in the ADONIS database.
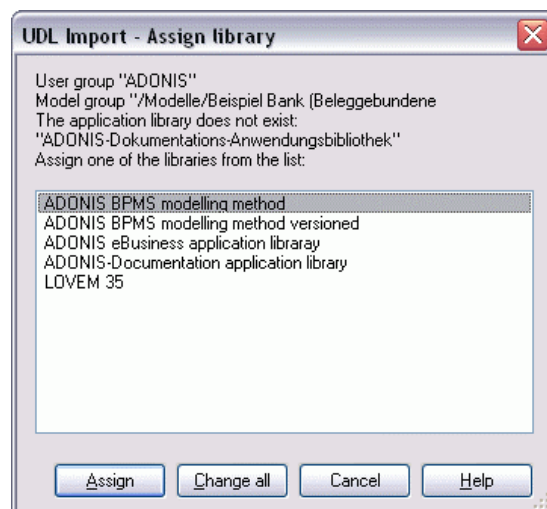
Figure 107: Assign library

Select the library you wish to assign. Then click on the button "Assign".

If you wish to replace all occurrences of the unknown library in the UDL file by the same library, click on the button "Change all".

Clicking on the "Cancel" button causes the import process to be cancelled (due to the unknown library).

1.7 UDL Export

UDL export can be applied for the purpose of data backup as well as for transferring existing ADONIS and system users to another ADONIS database.

When exporting ADONIS and system users, a so-called UDL file will be generated (UDL = **U**ser **D**efinition **L**anguage). In this - as in the ADL files for the model export - the user settings are saved in ASCII text format.

You can export ADONIS users by selecting in the menu "users" the option ""Export users" or by clicking on the icon 🔽. The window "User management - export users" (see fig. 108) will be displayed.
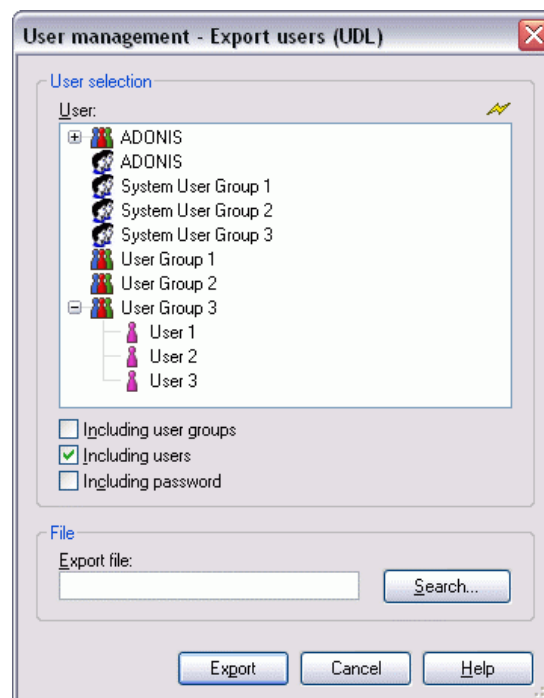
Figure 108: Export users/user groups

Select from the list of ADONIS/system user groups and users the ADONIS users and/or ADONIS user groups to be exported.

By opening the **context menu** (right mouse button) additional information for the user groups can be viewed (see chap. 1.4.1, p. 90) or the sort criterion for the listed user groups can be selected (see chap. 1.5.1, p. 107) (by default the alphabetically sorted user groups and users are shown). Further general functionality (see chap. 4.1, p. 34) is available.

The **option "Including User Groups"** exports - after you selected one or more ADONIS user/s - the ADONIS user groups to which the ADONIS users selected are assigned. In addition, these user groups' access rights to model groups are exported.

The **option "Including users"** exports - after one or more ADONIS user groups have been selected - all the ADONIS users contained in those groups.

The **option "Including password"** exports - after you selected one or more ADONIS user/s - also the coded passwords which are assigned to the ADONIS users.

> **Hint:** When exporting system users the passwords will not be exported, as these passwords are administered by the domain administration.

> **Hint:** Should you, for example, wish to export all the user groups and ADONIS users stored in the ADONIS database, select all user groups, activate the options "Including Users" and "Including User Groups" and then start the export process.

Please enter the path and the name of the UDL-file to which the users/user groups selected are to be exported into the field "Export File".

After selecting the users/user groups to be exported and entering the file name for the UDL file, click on the "Export" button to start the export process.

During the user export a status window will inform you about the current state of the export process. When the users have been exported successfully, a message window informs you that the export process is finished.

Hint: If the option "Including password" has not been selected, the current password is not exported during the export of an ADONIS user . When the user is re-imported, his password will be the user name instead of the former password

The standard user "Admin" cannot be exported.

2.   Library Management

Each ADONIS model is based on an ADONIS application library (ADONIS AL) which defines a customer-specific configuration of the modelling, analysis, simulation, evaluation and documentation/transformation component as well as any additional components such as process costing, personnel and resource planning and case/4/0 interface of ADONIS Business Process Management Toolkit (see chap. 4., p. 18). Every ADONIS user is assigned to his own application library (see fig. 109).
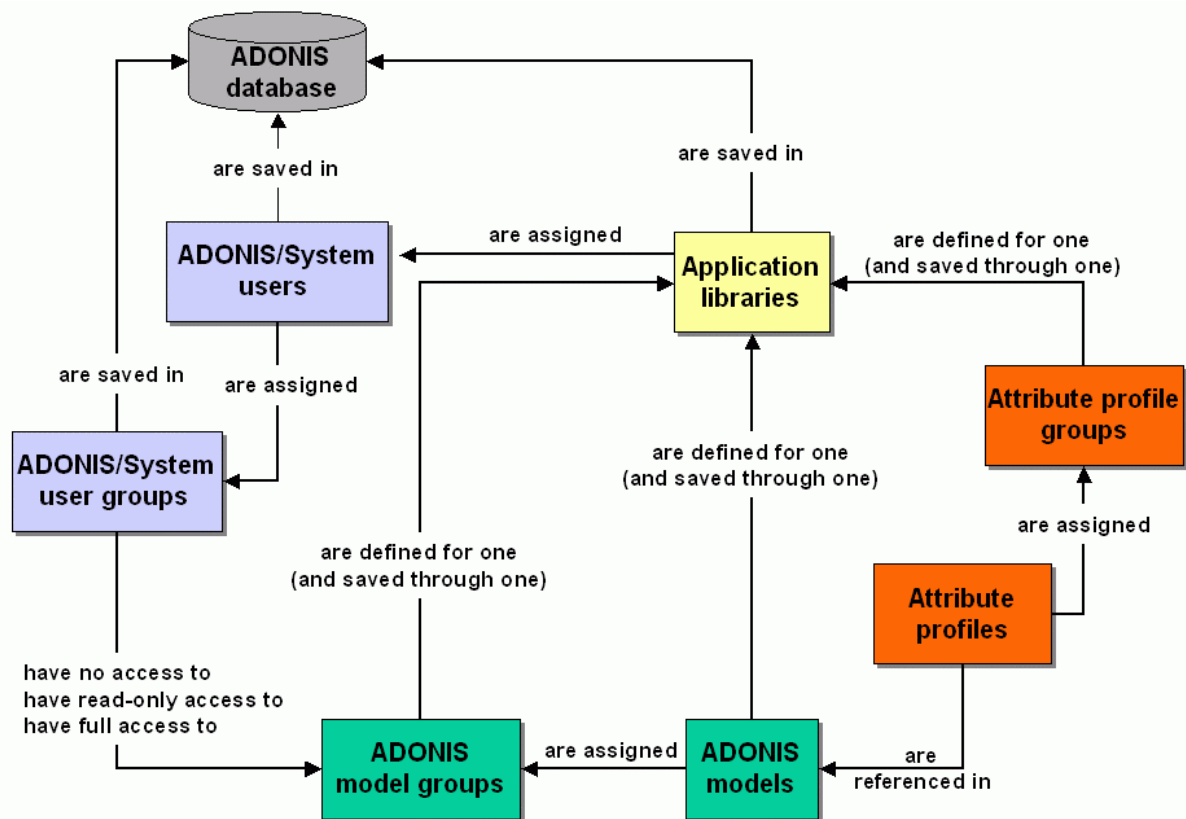
Figure 109: Overview of Administration Toolkit components

Every ADONIS application library is characterised both by the definition of the modelling method and by the definition of the evaluation mechanisms.

An ADONIS application library consists of a business process library (BP library) containing the classes and the relations for the business process models and a working environment library (WE library) containing the classes and the relations for the working environment models. These classes and relations can be combined in various model types (for example, in the ADONIS BPMS Application Library while in the ADONIS BPMS WE Library the model type "Working Environmentmodel" is defined).

In the ADONIS application the attribute profile classes (according to the repository concept in ADONIS) and table classes are also defined.

The classes and relations are defined in the business process or working environment libraries respectively (see fig. 110). The definition includes specifying the attributes available (modelling defini-

tions) and the graphical representation (layout definition). The definitions contained in the application library affect work on the models in the Business Process Management Toolkit.
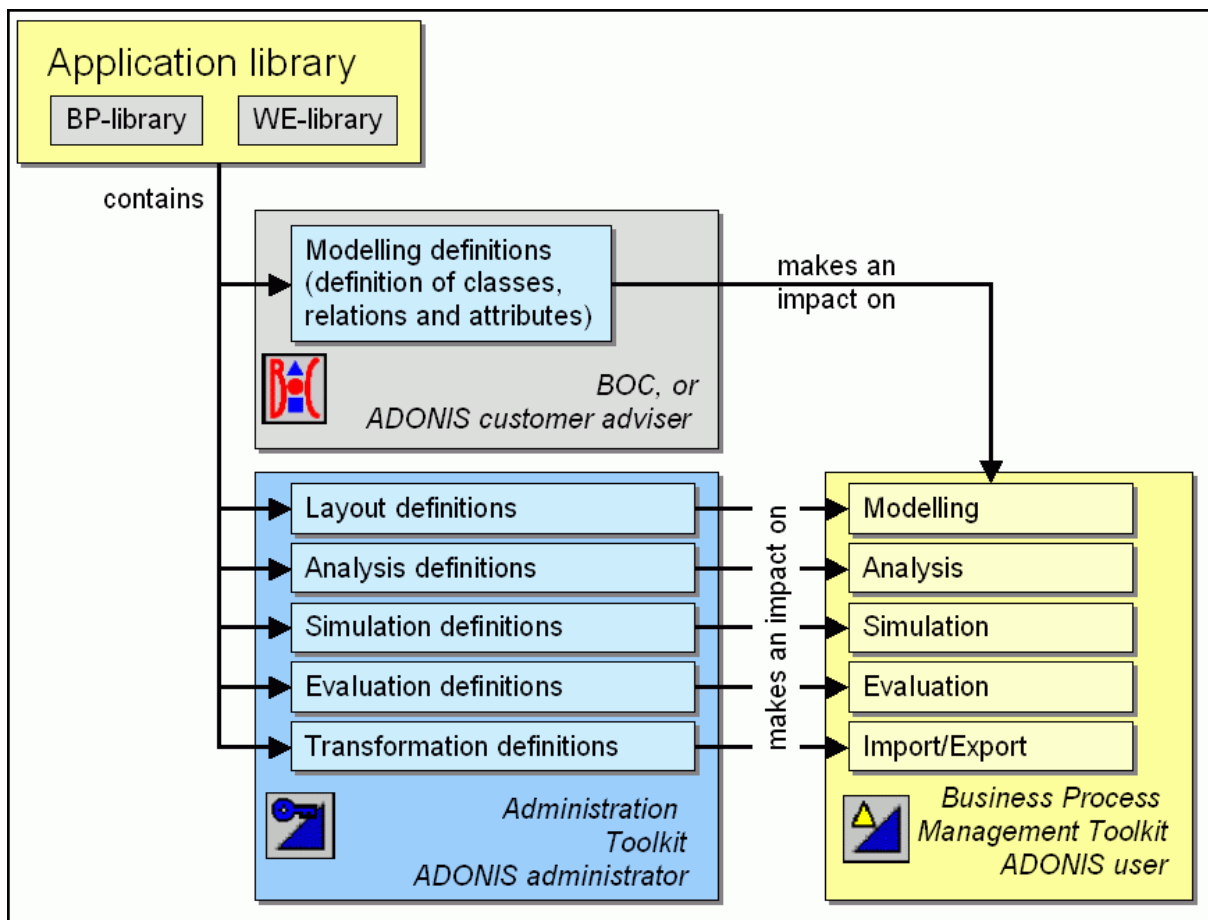
Figure 110: ADONIS application library

Each application library is defined by your ADONIS consultant or by BOC. The Layout-, Analysis-, Simulation-, Evaluation- and Transformation definitions can be changed by the ADONIS administrator in the library management component of the Administration Toolkit.

> **Hint:** The modelling definitions cannot be changed by the ADONIS administrator.

You can look at the ADONIS application libraries stored in your ADONIS database via the options "Settings", "Checks", or "Administration" in the menu "Libraries".

> **Hint:** The **ADONIS BPMS Application Library 3.9** (see chap. 20., p. 693) and its components - the **ADONIS BPMS BP Library 3.9** and the **ADONIS BPMS WE Library 3.9** - are created automatically and assigned to the standard user "Admin", whenever an ADONIS database is created.
>
> The ADONIS BPMS Application Library **cannot** be deleted (neither can the standard User "Admin").
>
> The class and library attributes of the ADONIS BPMS Application Library are described in the appendix of the ADONIS Administration Toolkit's documentation.
>
> The modelling classes and relations of the ADONIS BPMS Application Library are described in the appendix to the ADONIS Business Process Management Toolkit's documentation.

It is possible to define user-specific application libraries in addition to the ADONIS BPMS Application Library. These user-defined application libraries are defined by BOC Group and delivered as ABL

Files (see chap. 9.1, p. 557) (ABL=**A**DONIS **B**inary **L**anguage) These ABL files can then be imported into the ADONIS Administration Toolkit.

When you add a New ADONIS User (see chap. 1.4.3, p. 91), you have - among other things - to assign an ADONIS Application Library to this user so that he can create models. If you wish to assign a company-specific application library to him, you must first import (see chap. 2.3.1, p. 282) it.
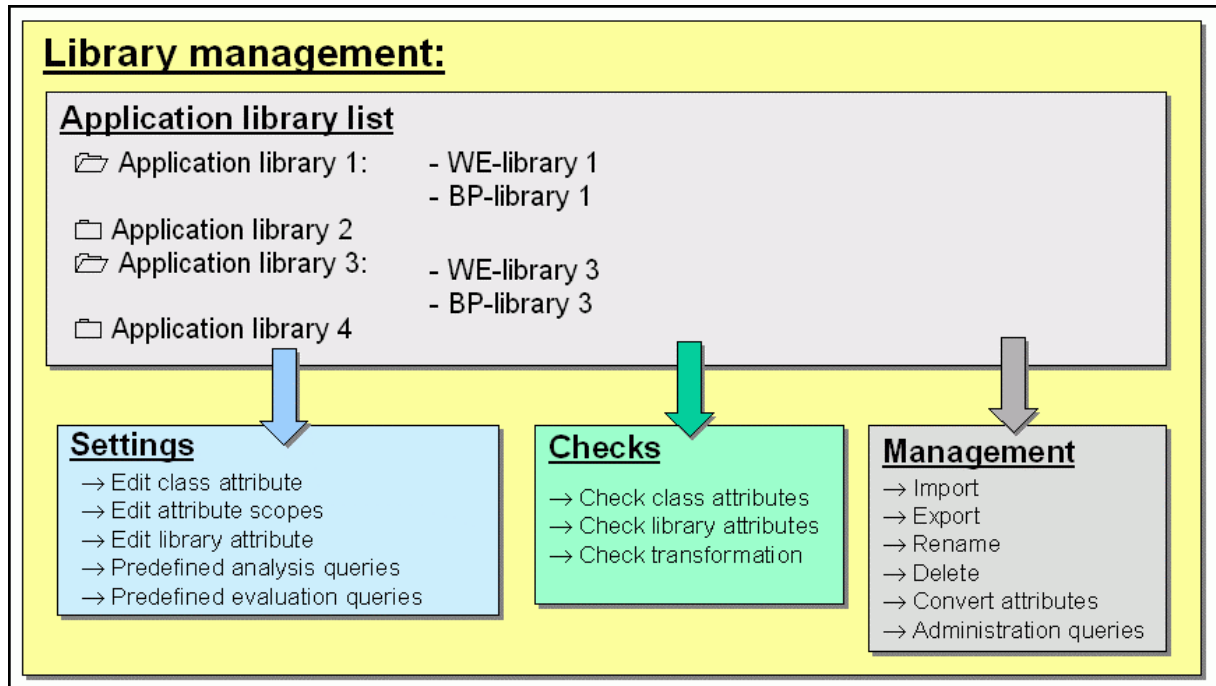
Figure 111: Functions of the library management

With support of the library management (see fig. 111), you can perform (through the selection of the apropriate sub menu item from the menu "Libraries"):

Option settings (see chap. 2.1, p. 128)

    edit class attributes,

    edit library attributes,

    edit value areas of enumeration attributes,

    pre-define analysis queries and

    pre-define evaluation queries.

Option checks (see chap. 2.2, p. 278)

    check class attributes and

    check library attributes.

Option management (see chap. 2.3, p. 281)

    import application libraries,

    export,

    delete or

    rename application libraries.

Option "Data administration" (menu "Extras", sub menu item "Data administration") allows for editing of external data files stored in a database (see chap. 15., p. 606), which are required for application library specific functions (e.g. documentation generation).

If you wish to make use of the services available within Library Management, follow the steps below:

1. Start (see Part IV., p. 527) the ADONIS Administration Toolkit (see chap. 3., p. 16).

2. Activate "Library Management" by clicking the corresponding smart-icon 📖 in the component panel.

Alternatively, you can click with the right mouse button on the Component Panel (see chap. 2., p. 30) (on the right side of the smart-icons for the components) and select the option "Library management" from the pop-up menu that appears. Another alternative is to open the pop-up menu with the function key <F9> and activate the library management by pressing the key <l>.

After you have activated the library management component, the quick-access panel (see fig. 112) will display the smart-icons for the settings, checks, administration, the import migration assistant and the export migration assistant.
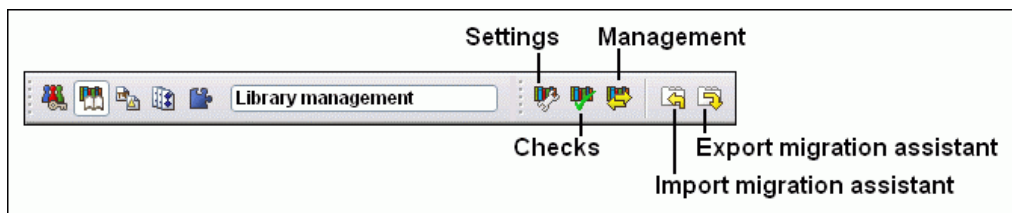
Figure 112: Quick-access bar of the Library Management

In the application library list the already created ADONIS application libraries currently saved in the ADONIS database will be listed in alphabetical order.

To display (see fig. 113) the application library list, select from the menu "Library" the menu item "Settings", "Checks" or "Administration", or click on the according SmartIcon 📖 in the quick-access bar.
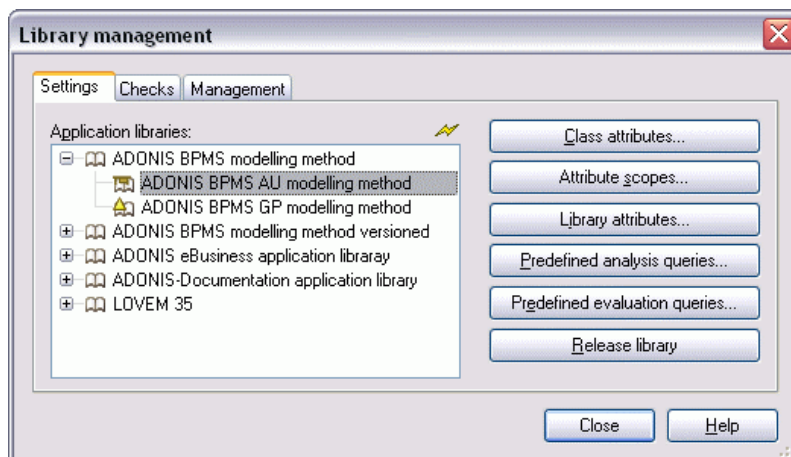
Figure 113: Application libraries list

2.1 Settings

Library configuration supports you when editing both class and library attributes.

Select the option "Library configuration" from the "Libraries" menu or click on the corresponding smart-icon 📖 in the Quick-access panel (see chap. 3.3, p. 33).

The window "Library management - library configuration" (see fig. 114) appears, listing all ADONIS application libraries stored in your ADONIS database.
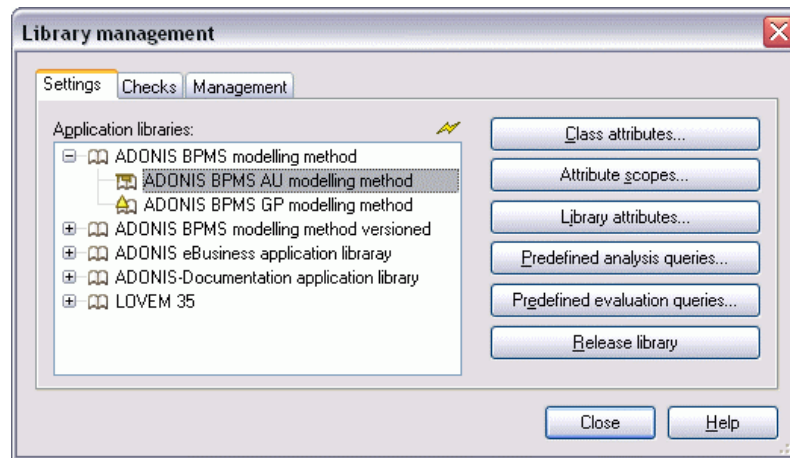
Figure 114: Application libraries' list - settings

Double-click on the application library you wish to edit to display the contianed BP and WE libraries. Select a library by clicking it and then choose one of the following actions:

- Edit class attributes (see chap. 2.1.1, p. 129)

- Extend attribute scopes (see chap. 2.1.2, p. 184)

- Edit library attributes (see chap. 2.1.3, p. 188)

- Edit analysis queries (see chap. 2.1.4, p. 254)

- Edit evaluation queries (see chap. 2.1.5, p. 277)

After clicking on the corresponding button, the previously selected library is loaded.

2.1.1   Edit class attributes

The following class attributes are created for all instantiable classes of every application library and control different graphical representations, the adjustment of attributes in ADONIS Notebooks and the connections to other models (Attribute "Modellzeiger").

After the loading of an application library a window "<Application library name> - Edit class attributes" (see fig. 115) will be opened.
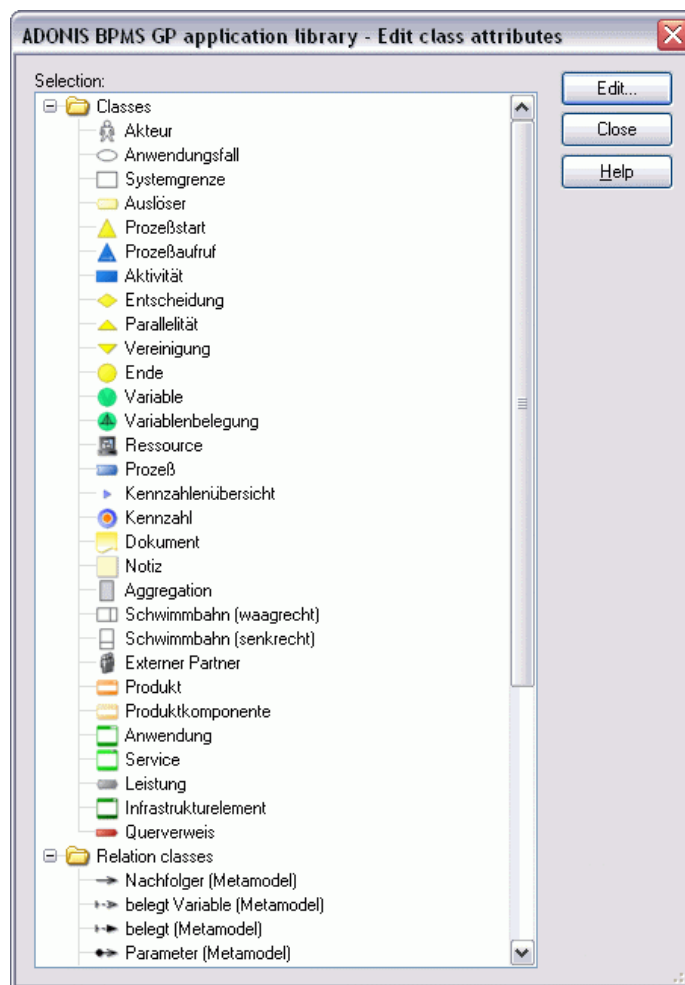
Figure 115: Edit class attributes

Select each class or relation, which attribute you want to edit and click on the button "Edit". You can edit the following class attributes:

GraphRep	Graphical representation of a class or relation during modelling. Graphical representation contains the form and colour of a class/relation and adjustment of visualised attribute values (see chap. 2.1.1.1, p. 131).
AttrRep	Adjustment of an attribute of a class or relation in ADONIS Notebook (see chap. 2.1.1.2, p. 171).
Modellzeiger	Specification of the attribute for model navigation via the "<Ctrl>+Double-click" functionality in the Modelling Component of the Business Process Management Toolkit. (see chap. 2.1.1.3, p. 180).
Klassenkardinalität	Restrictions for the number of relations (see chap. 2.1.1.4, p. 181)
Zulässige Objekte	Allows modelling classes in swimlanes.
Conversion	Rules for converting objects from one class to another.

After clicking "Edit" a notebook showing the class attributes that can be edited appears (see fig. 116).
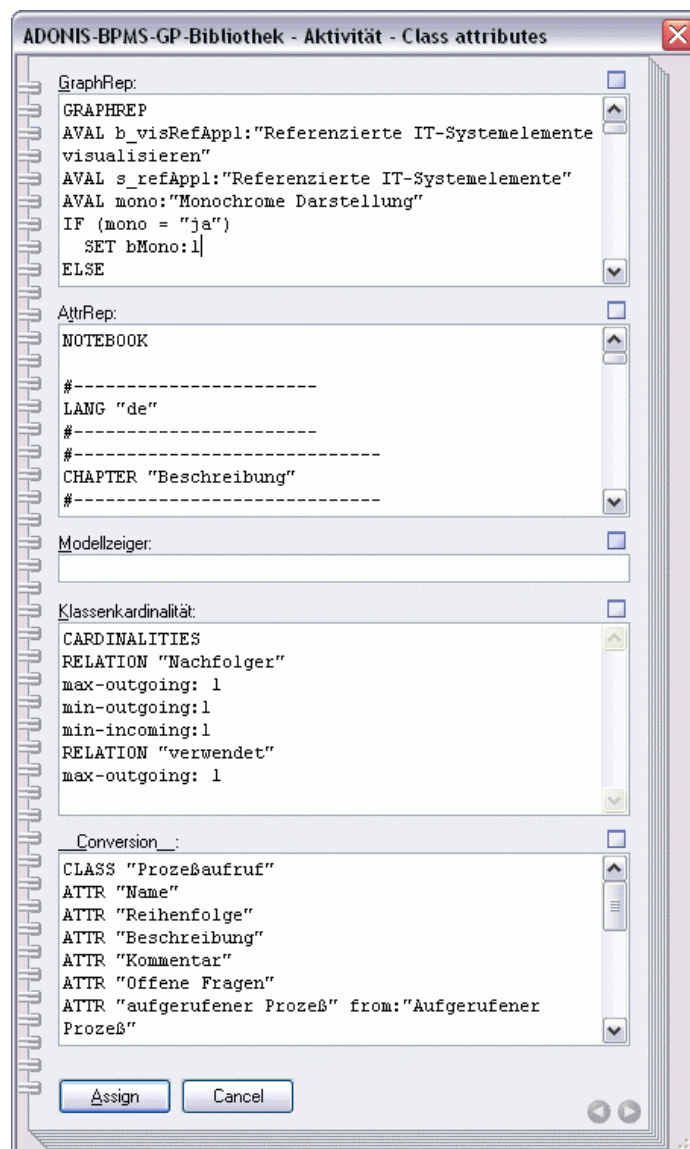
Figure 116: Class attributes which can be edited

2.1.1.1   GraphRep

The class attribute "GraphRep" controls the graphical representation of a class or relation and the arrangement of any visualised attribute values when modelling in the ADONIS Business Process Management Toolkit.

The language for the graphical representation of objects and relations is based on the GRAPHREP syntax (see p. 134).

By clicking the "Dialogue" button (see chap. 4.2.2, p. 40) in the class attributes' notebook (see fig. 116) you can display a preview of the attribute definition in the window "<*Class/Relation*> - GraphRep" (see fig. 117).
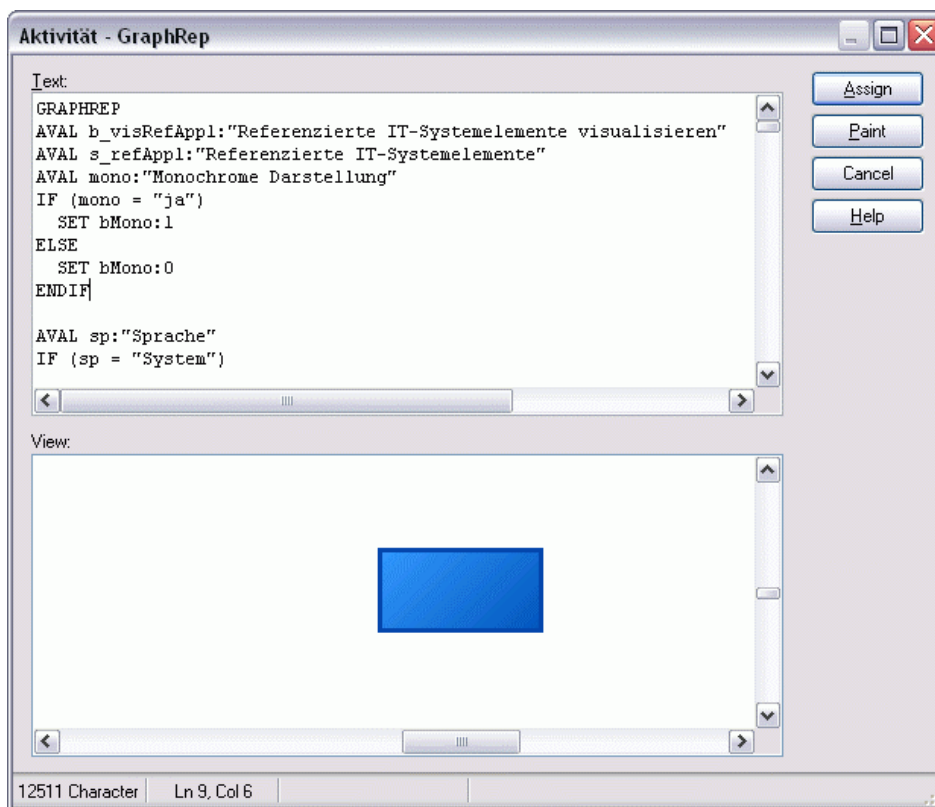
Figure 117: Graphical representation

The attribute value can be changed in the **field "Text"**. Click on the button **"Assign"** and the current value will be stored. >

If you click on the button **"Paint"**, the current attribute definition will be graphically represented in the **field "View"**.

In the **status bar** you are shown: in the first field (to the left) the number of characters in the field "Text", in the second field (middle) the current cursor position in the field "Text" and in the third field (right) the co-ordinates of the mouse points in the field "View".

The graphical representation can consist of basic shapes, compounds (e.g. blue triangle with arrow), and visualised texts and attributes.

> **Hint:** Visualised attributes are not shown in the field "View".

The following types of elements are distinguished:

- Classifying elements,
- Stylistic elements,
- Graphic elements,
- Variable assigning elements and
- Controlling elements.

Classifying elements only exist for relations (see below). They specify whether the starting, the middle or the endpoint of the relation is being defined. (i.e. once the keyword "START" is used, the following description refers to the startpoint of the relation until the next classifying element is specified.(**START/MIDDLE/END**). A fourth classifying element concerns the drawing of a relation's edge. This is the line from the starting point via possible bendpoints to the end point of a relation. (**EDGE**).

Style Elements define the look for the graphic elements following the definition:

- **PEN** sets the characteristics of the "Pen",

- **FILL** sets the characteristics of the "Fill-in brush",

- **SHADOW** switches the shadow on or off,

- **STRETCH** switches geometric stretches on or off,

- **FONT** sets the font and

- **GRADIENT** defines a colour gradient.

Pen determines in which manner the lines and curves are drawn, i.e. how broad, in which colour and in which style (e.g. broken line). In the case of graphical elements that can be filled, only the outline of the surface is drawn by the "Pen". The **Fill-in brush** determines how the object is filled.

Graphical elements which can **not** be filled are point (**POINT**), line (**LINE**), polyline (**POLYLINE**), Bezier curve (**BEZIER**), arc (**ARC**) and function curve (**CURVE**). Available surface elements, which can be filled with a colour, are rectangle (**RECTANGLE**), rectangle with rounded corners (**ROUNDRECT**), polygon (**POLYGON**), ellipse (**ELLIPSE**), pie (**PIE**) and compound (**COMPOUND**).

The coordinates (positions) and dimensions must be specified for graphical elements. Coordinates here are relative to the position of the particular object, i.e. they are added to the object's position.

The starting, the middle or the end (point) of the graphical representation can be defined for **relations**. Positions then refer to one of these three points. However, the coordinate system is turned depending on the course of the relation. When you define the position, you must regard the relation as leading horizontally from left to the right. The coordinate system's origin then is the point of the relation for which the graphical representation is just being defined i.e. Start, Midpoint or End.

On the x-axis the coordinate values increase from the left to right, on the y-axis they increase from top to bottom (differing from mathematics). Arcs and pies are rotated counter-clockwise (see fig. 118).
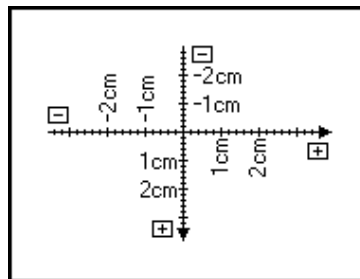
Figure 118: Co-ordinate system for the graphical representation

ATTENTION: The unit of measure for positions and proportions (cm or pt) must be defined in every case. Pixels cannot be used.

Language-specific elements (see p. 170) can be added too. A predefined variable **_uilang** contains the language of the user interface with which ADONIS was launched. This variable can be used in the graphical design.

Furthermore, **Device-specific elements** (see p. 171) can provide different graphical representations for different target devices (drawing area, printer, SVG, etc.).

When drawn, the elements are processed sequentially. The **control elements**, however, make it possible to skip sections during processing depending on variables. Attribute values of the object to be represented may for example be assigned to variables. A graphical representation depending on object attributes can therefore be obtained using variable assignment elements combined with control elements. Additional possibilities result from using variables in graphical elements.

GRAPHREP syntax

The language that describes the graphical representation of objects and relations is based on the following syntax:

<ins>*GraphRep*</ins> **:**	**GRAPHREP** *Traits* [**threshold:***intVal*] [**icon-scale:***realVal*] *ElementSequence* .
Traits **:**	*SwimlaneTraits* │ *NodeTraits* │ *EdgeTraits* .
SwimlaneTraits **:**	**swimlane:***SwimlaneType* .
SwimlaneType **:**	**horizontal** │ **vertical** .
NodeTraits **:**	[**layer:***n*] [**sizing:***SizingType*] [**width-sizing:***SizingType*] [**height-sizing:***SizingType*] [**smart-symbol-size**] .
SizingType **:**	**keep-aspect-ratio** │ **symmetrical** │ **asymmetrical** .
EdgeTraits **:**	[**layer:***n*] [**rounded:** *translation*] [**bridge-radius:** *radius*] [**bridge-seg-count:** *intVal*] [**no-edge**] [**start-trans:***translation*] [**end-trans:***translation*] .
ElementSequence **:**	{ *GraphElement* } .
GraphElement **:**	*Edge* │ *Start* │ *Middle* │ *End* │ *Pen* │ *Fill* │ *Shadow* │ *Stretch* │ *Font* │ *ClipRect* │ *ClipRoundRect* │ *ClipPoly* │ *ClipEllipse* │ *ClipOff* │ *Point* │ *Line* │ *PolyLine* │ *Arc* │ *Bezier* │ *Curve* │ *Rectangle* │ *RoundRect* │ *Polygon* │ *Ellipse* │ *Pie* │ *Compound* │ *Bitmap* │ *GradientRect* │ *GradientTri* │ *Text* │ *Attr* │ *Hotspot* │ *Set* │ *Aval* │ *Table* │ *TextBox* │ *AttrBox* │ *BitmapInfo* │ *IfStatement* │ *WhileStatement* │ *ForNumStatement* │ *ForTokenStatement* .
Edge **:**	**EDGE** .
Start **:**	**START** .
Middle **:**	**MIDDLE** .
End **:**	**END** .
Pen **:**	**PEN** [**style:***PenStyle*] [**w:***width*] [**color:***ColorSpecOrExpr*] [**endcap:***EndCap*] [**join:***Join*] .
PenStyle **:**	**solid** │ **dot** │ **dash** │ **dashdot** │ **inside-frame** │ **null** .
ColorSpecOrExpr **:**	*ColorSpec* │ *intExpr* .
EndCap **:**	**round** │ **square** │ **flat** .
Join **:**	**bevel** │ **miter** │ **round** .
Fill **:**	**FILL** [**style:***BrushStyle*] [**color:***ColorSpecOrExpr*] [**fcolor:***ColorSpecOrExpr* │ **transparent**] .
BrushStyle **:**	**solid** │ **horz** │ **vert** │ **cross** │ **diagcross** │ **updiag** │ **downdiag** │ **mix25** │ **mix50** │ **mix75** │ **null** .
Shadow **:**	**SHADOW** *OnOrOff* .
Stretch **:**	**STRETCH** *OnOrOff* .

OnOrOff :	`on` | `off` .
Font :	`FONT` [*fontNameExpr*] [**h:***fontHeightExpr*] *FontStyle* [**color:***ColorSpecOrExpr*] [**line-orientation:***realExpr*] .
FontStyle :	**style:***fontStyleExpr* | *DirectFontStyle* .
DirectFontStyle :	[`bold`] [`underline`] [`italic`] .
ColorSpecOrExpr :	*ColorSpec* | *intExpr* .
ClipRect :	`CLIP_RECT` [**x:***xposExpr*] [**y:***yposExpr*] [**w:***widthExpr*] [**h:***heightExpr*] [**combine-mode:***CombineMode*] .
ClipRoundRect :	`CLIP_ROUNDRECT` [**x:***xExpr*] [**y:***yExpr*] [**w:***wExpr*] [**h:***hExpr*] [**rx:***rxExpr*] [**ry:***ryExpr*] [**combine-mode:***CombineMode*] .
ClipPoly :	`CLIP_POLY`*n* [**x1:***xposExpr*] [**y1:***yposExpr*] ... [**x***n***:***xposExpr*] [**y***n***:***yposExpr*] [**combine-mode:***CombineMode*] .
ClipEllipse :	`CLIP_ELLIPSE` [**x:***xposExpr*] [**y:***yposExpr*] [**rx:***xradiusExpr*] [**ry:***yradiusExpr*] [**combine-mode:***CombineMode*] .
CombineMode :	`copy` | `and` | `or` | `diff` | `xor` .
ClipOff :	`CLIP_OFF` .
Point :	`POINT` [**x:***xposExpr*] [**y:***yposExpr*] .
Line :	`LINE` [**x1:***xposExpr*] [**y1:***yposExpr*] [**x2:***xposExpr*] [**y2:***yposExpr*] .
Rectangle :	`RECTANGLE` [**x:***xposExpr*] [**y:***yposExpr*] [**w:***widthExpr*] [**h:***heightExpr*] .
RoundRect :	`ROUNDRECT` [**x:***xposExpr*] [**y:***yposExpr*] [**w:***widthExpr*] [**h:***heightExpr*] [**rx:***xradiusExpr*] [**ry:***yradiusExpr*] .
PolyLine :	`POLYLINE` *n* [**x1:***xposExpr*] [**y1:***yposExpr*] ... [**x***n***:***xposExpr*] [**y***n***:***yposExpr*] .
Polygon :	`POLYGON` *n* [**x1:***xposExpr*] [**y1:***yposExpr*] ... [**x***n***:***xposExpr*] [**y***n***:***yposExpr*] .
Ellipse :	`ELLIPSE` [**x:***xposExpr*] [**y:***yposExpr*] [**rx:***xradiusExpr*] [**ry:***yradiusExpr*] .
Arc :	`ARC` [**x:***xposExpr*] [**y:***yposExpr*] [**rx:***xradiusExpr*] [**ry:***yradiusExpr*] [**x1:***xposExpr*] [**y1:***yposExpr*] [**x2:***xposExpr*] [**y2:***yposExpr*] .
Pie :	`PIE` [**x:***xposExpr*] [**y:***yposExpr*] [**rx:***xradiusExpr*] [**ry:***yradiusExpr*]

	[**x1:**_xposExpr_] [**y1:**_yposExpr_] [**x2:**_xposExpr_] [**y2:**_yposExpr_] .

Bezier :	**BEZIER** _n_
	    [**x1:**_xposExpr_] [**y1:**_yposExpr_]
	    ...
	    [**x**_n_**:**_xposExpr_] [**y**_n_**:**_yposExpr_] .

Curve :	**CURVE** _varName_ **from:**_realVal_ **to:**_realVal_ **fx:**_realExpr_ **fy:**_RealExpr_ .
Bitmap :	**BITMAP** _fileNameExpr_ [**x:**_xposExpr_] [**y:**_yposExpr_] [**w:**_widthExpr_] [**h:**_heightExpr_] .
GradientRect :	**GRADIENT_RECT** [**x:**_xExpr_] [**y:**_yExpr_] [**w:**_wExpr_] [**h:**_hExpr_] [**mode:**_GradientMode_] [**color1:**_ColorSpecOrExpr_] [**color2:**_ColorSpecOrExpr_] **color3:**_ColorSpecOrExpr_ **color4:**_ColorSpecOrExpr_ .
GradientMode :	**horz** | **vert** | **updiag** | **downdiag** | **diagcross** .
GradientTri :	**GRADIENT_TRI** [**x1:**_xExpr_] [**y1:**_yExpr_] [**color:**_ColorSpecOrExpr_] [**x2:**_xExpr_] [**y2:**_yExpr_] [**color:**_ColorSpecOrExpr_] [**x3:**_xExpr_] [**y3:**_yExpr_] [**color:**_ColorSpecOrExpr_] .
Compound :	**COMPOUND** _n_ { _CompoundableElem_ } .
	Hint: Die Anzahl der _CompoundableElem_ ist _n_.
CompoundableElem :	_Line_ | _PolyLine_ | _Curve_ .
Text :	**TEXT** _StrExpr_ [**line-break:**_LineBreakMode_] [_TextXCoord_] [_TextYCoord_] [_TextWidth_] [_TextHeight_] [_LineHeight_] .
Attr :	**ATTR** _attrName_ [**text:**_strExpr_] [**format:**_strValue_] [**sep:**_strValue_] [**row:**_intExpr_] [**col:**_attrName_] [**line-break:**_LineBreakMode_] [_TextXCoord_] [_TextYCoord_] [_TextWidth_] [_TextHeight_] [_LineHeight_] .
Hotspot :	**HOTSPOT** _attrName_ [**text:**_strExpr_] [**row:**_intExpr_] [**col:**_attrName_] [**x:**_xposExpr_] [**y:**_yposExpr_] [**w:**_widthExpr_] [**h:**_heightExpr_] .
Set :	**SET** { _var_ **:**_anyExpr_ } .
Aval :	**AVAL** { _AvalAssignment_ } .
AvalAssignment :	[**set-format:**_strValue_] [**set-sep:**_strValue_] [**set-default:**_strValue_] [**as-original-type**] [**set-count-rows:**] [**set-row:**_intExpr_ **set-col:**_attrName_] _var_**:**_attrName_ .
Table :	**TABLE** [**x:**_xpos_] [**y:**_ypos_] [**w:**_width_] [**h:**_height_] **cols:**_n_ **rows:**_m_ [**w1:**_width_] ... [**w**_n_ **:**_width_] [**h1:**_height_] ... [**h**_m_ **:**_height_] .
TextBox :	**TEXTBOX** _strExpr_ [**line-break:**_LineBreakMode_]

	[*TextXCoord*] [*TextYCoord*] [*TextWidth*] [*TextHeight*] [*LineHeight*] .
AttrBox :	**ATTRBOX** *attrName* [**text:***strExpr*] [**format:***strValue*] [**sep:***strValue*] [**row:***intExpr*] [**col:***attrName*] [**line-break:***LineBreakMode*] [*TextXCoord*] [*TextYCoord*] [*TextWidth*] [*TextHeight*] [*LineHeight*] .
LineBreakMode :	**off** | **words** | **rigorous** .
TextXCoord :	**x**[**:abs**]**:***xposExpr* .
TextYCoord :	**y**[**:abs**]**:***yposExpr* .
TextWidth :	**w:***widthExpr* | **w:***WAlign* [**:***widthExpr*] .
WAlign :	**l** | **c** | **r** .
TextHeight :	**h:***heightExpr* | **h:***HAlign* [**:***heightExpr*] .
HAlign :	**t** | **c** | **b** .
LineHeight :	**line-height:***heigthExpr* .
BitmapInfo :	**BITMAPINFO** *fileNameExpr* .
IfStatement :	*NewIfStatement* | *OldIfStatement* .
NewIfStatement :	**IF** *boolExpr* **{** *StatementSequence* **}** **{ ELSIF** *booleanExpr* **{** *StatementSequence* **} }** [**ELSE { ** *StatementSequence* **}] .**
OldIfStatement :	**IF** *condExpr* *ElementSequence* **{ ELSIF** *condExpr* *ElementSequence* **}** [**ELSE** *ElementSequence*] **ENDIF** .
WhileStatement :	**WHILE** *condxpr* **{** *ElementSequence* **}** .
ForNumStatement :	**FOR** *varName* **from:** *numExpr* **to:** *numExpr* [**by:** *numExpr*] **{** *ElementSequence* **}** .
ForTokenStatement :	**FOR** *varName* **in:** *strExpr* [**sep:** *strExpr*] **{** *ElementSequence* **}** .

- *n*, *m* and *intVal* are integers (not expressions).

- *intExpr* is either an integer value or an integer expression.

- *realVal* is a floating-point number (not an expression).

- *realExpr* is either a floating-point number or a floating-point number expression.

- *translation*, *radius*, *xpos*, *ypos*, *width* and *height* are measured values (not expressions).

- *fontHeightExpr*, *xPosExpr*, *yPosExpr*, *widthExpr* and *heightExpr* are either measured values, or measured value expressions.

- *attrName* and *varName* are strings (not expressions).

- *fontNameExpr*, *fontStyleExpr*, *fileNameExpr* and *strExpr* are either strings or string expressions.

- *condExpr* is a condition (boolean expression).

- *anyExpr* is an arbitrary expression or a value.

- *var* is a variable (not a string like *varName*).

- *rectExpr* and *xyExpr* are arrays with measured values.

Example:

```
GRAPHREP
FILL r:0 g:128 b:255
ELLIPSE x:-1cm rx:1cm ry:2cm
POLYGON 3 x1:0cm y1:-2.4cm x2:2.22cm y2:1cm x3:-.5cm y3:1.3cm
CURVE "t" from:0 to:3.14 fx:(t) fy:(sin (t))
FONT h:10pt
TEXT "aha" x:-2.5cm w:5cm
```

GRAPHREP

The **GRAPHREP** element initiates the definition of graphical representation of objects and relations.

The language that describes the graphical representation of objects and relations is based on the following syntax:

GraphRep :	**GRAPHREP** *Traits* [**threshold:***intVal*] [**icon-scale:***realVal*] *ElementSequence* .
Traits :	*SwimlaneTraits* | *NodeTraits* | *EdgeTraits* .
SwimlaneTraits :	**swimlane:***SwimlaneType* .
SwimlaneType :	**horizontal** | **vertical** .
NodeTraits :	[**layer:***n*] [**sizing:***SizingType*] [**width-sizing:***SizingType*] [**height-sizing:***SizingType*] [**smart-symbol-size**] .
SizingType :	**keep-aspect-ratio** | **symmetrical** | **asymmetrical** .
EdgeTraits :	[**layer:***n*] [**rounded:***translation*] [**bridge-radius:***radius*] [**bridge-seg-count:***intVal*] [**no-edge**] [**start-trans:***translation*] [**end-trans:***translation*] .
ElementSequence :	{ *GraphElement* } .
GraphElement :	*Edge | Start | Middle | End |* *Pen | Fill | Shadow | Stretch | Font |* *Point | Line | PolyLine | Arc | Curve |* *Rectangle | RoundRect | Polygon | Ellipse | Pie |* *Compound | Bitmap | Text | Attr | Hotspot |* *Set | Aval | Table | TextBox | AttrBox | BitmapInfo |* *IfStatement* .

Here the **layer** in which the objects will be drawn can be defined. The default value is 1, the greater the value, the higher level at which the objects or relations of the particular class are situated.

By default all the objects have fixed, unchangeable **height**. However, you can define in GraphRep whether and to what extent the size of an object can be changed. There are the following possibilities to do it:

- **sizing**: both dimensions simultaneously and

- **width-sizing**, **height-sizing**: separately for the width and the height.

In case of **sizing** a size changing type can be set that ensures that the relation between the sides stays constant (**keep-aspect-ratio**). Changing the object's width then causes the height to be adjusted

simultaneously - and vice versa. A **symmetrical** change of size on the other hand means that if one side of an object is changed, the opposite side will automatically be influenced accordingly. If the right side of an object is pulled to the right the left side of the object moves to the left at the same time. In the case of an **asymmetrical** change of size, then, an object's sides can be changed independently of each other.

Rounded defines a "rounding" of the edges' corners at bendpoints for relations.

Intersections of connector edges are shown normally without bridges. If you want to have such bridges in the standard version of a system, the bridgeovers have to defined in **bridge-radius** attribute. The bigger the radius, the more exaggerated the bridges. If you do not specify the bridges, they will possess 5 segments. To have more or fewer of them, please create attribute **bridge-seg-count** (recommended: odd number).

> **Hint:** Each user can albeit show and remove standard definition for each separate model.

The attribute **no-edge** is also only used for relations. When it is set, the automatic insertion of an **EDGE** element (see p. 150) is prevented. This means for example that a relation can be invisible.

The attributes **start-trans** and **end-trans** can move the starting or end points of an edge. If you would like the endpoint of an edge to be further away from the objects it joins, then you will need to enter a negative value. (This is the most common use of start-trans and end-trans). The endpoints of the edges are by default 0.15cm away from an object. This is the value you have to enter to ensure that the relations join up exactly with the objects they connect.

Example:

```
GRAPHREP
PEN color:lightblue w:.08cm
EDGE start-trans:-.3cm end-trans:-.3cm
START
POLYLINE 4 x1:0cm y1:0cm x2:-.1cm y2:.18cm
          x3:-.2cm y3:-.18cm x4:-.3cm y4:0cm
END
POLYLINE 3 x1:-.4cm y1:.15cm x2:0cm y2:0cm x3:-.4cm y3:-.15cm
```

The graphical elements do in fact touch the two originating points of the relation (the left-most and the right-most point of the connector represented in the picture below (see fig. 119)). This, however, does not hold for the edge of the relation - each starts and ends 0.3cm prior to these points which creates the visible spaces.

A special kind of object represents the swim lanes. They are in the lowest level and adjust automatically to the drawing area. The important thing about a swim lane is not the GraphRep description but the fact that it's derived from a swim lane meta model class. Within the **GRAPHREP** element the attribute **swimlane** determines, whether it is a horizontal (**horizontal**) or a vertical (**vertical**) swim lane. Horizontal swim lanes are always as big as the drawing area and the height can be changed. Vertical swim lanes always adapt the height of the drawing area and can be changed in the width.

Example:

```
GRAPHREP swimlane:horizontal
SHADOW off
FILL color:lightblue
PEN style:null
RECTANGLE w:4cm h:4cm
PEN color:black
```

```
LINE x1:0cm y1:0cm x2:4cm y2:0cm
LINE x1:0cm y1:4cm x2:4cm y2:4cm
```

It is a light blue horizontal swim lane which is separated on the upper and the lower edge through a black line from the adjacent swim lanes.
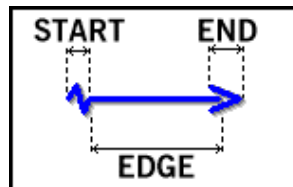
Figure 119: Composition of a relation

SET

Set :   **SET** { *var :anyExpr* } **.**

The **SET** element makes is possible to set runtime variables. No runtime variables are set before the *GraphRep* program is processed. A value assigned may be a constant or the result of an expression. In expressions it is possible to access runtime variables already set.

Example:

```
SET ei:10 day:"Monday" x:2.5 len:.73cm
SET ideal:(cond (day = "Saturday" OR day = "Sunday",
               3 * ei, 2 * ei))
```

Here, the value 20 is assigned to the runtime variable **ideal**, since **day** has the value "Monday". Therefore the condition specified is not fulfilled and **ei** has the value 10.

AVAL

Aval :   **AVAL** { *AvalAssignment* } **.**

AvalAssignment :   [**set-format:***strValue*] [**set-sep:***strValue*]
                         [**set-default:***strValue*] [**as-original-type**]
                         [**set-count-rows:**] [**set-row:***intExpr* **set-col:***attrName*]
                         var:attrName **.**

The **AVAL** element can assign attribute values of the current object to runtime variables. As for **SET** (see p. 140), for each attribute specified in **AVAL** a runtime variable with the same name is generated (over-written). The value specified behind the attribute's name has to be a string. This is interpreted as the name of the instance attribute to be evaluated. The value of this instance attribute is calculated and assigned to the runtime variable.

The variables set in this way can be used in other elements (especially in the **IF**-Element (see p. 142). This enables the system to make attribute-dependent graphical representations. The positions in which expressions may be used and thus runtime variables be set can be recognised by "*...ValueOrExpr*" in the syntax definition.

Example:

```
GRAPHREP
AVAL w:"Value"
TEXT (cond (VAL w < 0, "-", "+"))
```

If the attribute's value is a negative number, a "-" is the output, otherwise it is a "+".

The symbols in the modelling bar do not belong to concrete objects. Therefore, **AVAL** uses the default values of the instance attributes when evaluating them. This could lead to an undesired representation in the modelling panel.

Example:

The attribute "Description" (default value: empty text) is to give a warning, if the attribute is not set. This warning, however will not be shown in the modelling bar.

Entering the attribute **set-default** causes the following variable to be assigned a different value ("xyz") for the representation of the modelling bar.

```
AVAL set-default:"xyz" n:"Description"
IF (NOT LEN n)
  TEXT "<Description missing!>"
ENDIF
```

This means that the warning "<Description missing!>" will always occur for objects without a description, but not in the modelling bar.

Through inserting the attribute **set-format** into **Type INTERREF attributes and text entering (***strValue***)**, (see chap. 5.12, p. 539) the user can define his own order and run-time variable. As a result the following place holders will be respectively replaced:

- **%o** through the object names,

- **%c** through the class names,

- **%m** through the model names,

- **%M** through the model names and version numbers,

- **%t** through the model type names,

- **%v** through the internal version number (in format YYYY:MM:TT),

- **%V** through the internal version number defined in format of versioning.

  Hint: The place holders **%M**, **%v** and **%V** will be analysed only in application libraries with time-related versioning (see chap. 6.1.2, p. 69).

Through inserting the attribute **set-sep** in case of multivalued **attributes of type INTERREF** (see chap. 5.12, p. 539), the user can define separators (*strValue*) between references. Standard separator "\r\n" results in the display of references in separate lines.

If the parameter **as-original-type** is specified, the variable will not be of type STRING, but of the same type as the attribute read.

With **set-count-rows** and **set-row**/**set-col** it is possible to access attributes of type *Table* (RECORD). **set-count-rows** counts the table rows, **set-row** and **set-col** define the access to a specific table cell.

Example:

```
AVAL set-count-rows rowcount:"Responsible Persons"
FOR i from:1 to:(rowcount)
{
  AVAL set-row:(i) set-col:"Department" dep:"Responsible Persons"
  TEXT (dep) x:0cm y:(1cm + i * 0.5cm) w:c h:t
}
```

AVAL can be accessed not only in instance attributes, but also in *class attributes*. However, because changing class attributes is not possible during modelling, clicking on a class attribute only opens a viewer.

IF ... ELSIF ... ELSE ... ENDIF / WHILE / FOR

IfStatement :	*NewIfStatement	OldIfStatement .*
NewIfStatement :	**IF** *boolExpr* **{** *ElementSequence* **}** **{ ELSIF** *booleanExpr* **{** *ElementSequence* **} }** **[ELSE {** *ElementSequence* **}]** .
OldIfStatement :	**IF** *boolExpr ElementSequence* **{ ELSIF** *boolExpr ElementSequence* **}** **[ELSE** *ElementSequence* **]** **ENDIF** .
WhileStatement :	**WHILE** *boolExpr* **{** *ElementSequence* **}** .
ForStatement :	*ForNumStatement	ForTokenStatement .*
ForNumStatement :	**FOR** *varName* **from:** *numExpr* **to:** *numExpr* **[by:** *numExpr* **]** **{** *ElementSequence* **}** .
ForTokenStatement :	**FOR** *varName* **in:** *strExpr* **[sep:** *strExpr* **]** **{** *ElementSequence* **}** .

IF

The **IF** element is used to code branches depending on one or more conditions (*boolExpr*). The first condition is specified with **IF**, all others (optional) follow with **ELSIF** commands. The conditions are checked one after another, until one of them is recognised as "true". As soon as this happens, the element sequence following this condition is executed; all other elements of the IF statement (before and after) are ignored. Afterwards, the program execution resumes with the code after the **ENDIF** statement. If none of the conditions evaluated to be true, the sequence following **ELSE** is processed or if there is no ELSE clause is included then the program continues from the next ENDIF. Any number of **ELSIF**s is possible. **IF** elements may be used with as many hierarchical levels as required.

If you use a combination of **IF** and **AVAL** (see p. 140), different objects of the same class can have different graphical representations depending on the respective instance attribute's value.

Example: Global end

```
GRAPHREP
ELLIPSE rx:.7cm ry:.7cm
AVAL t:"Type"
```

```
IF (t = "global")
{
  ELLIPSE rx:.45cm ry:.45cm
}
```

The class for which the graphic representation is defined here has an instance attribute "Type" of the type ENUMERATION: The possible values are "local" and "global". If the instance attribute's value is set to "local", the resulting symbol is the one on the left hand side of the figure below (see fig. 120), in case of "global" it is the one on the right.
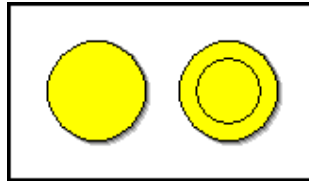
Figure 120: An example of the graphical presentation of a local/global end

WHILE

WHILE repeats the drawing of an element or a block of elements as long as a given expression is TRUE.

> **Hint:** Endless WHILE loops are *not* aborted automatically by ADONIS. Therefore make sure, that within the loop at least one variable of the WHILE expression changes to enable reaching the state FALSE within a finite number of runs.

Example:

```
GRAPHREP
SHADOW off
PEN style:null
SET t:-3.14
WHILE (t < 3.14)
{
  SET x:(t) y:(sin(t))
  ELLIPSE x:(CM x) y:(CM y) rx:0.2cm ry:0.2cm
  SET t:(t + 0.1)
}
```
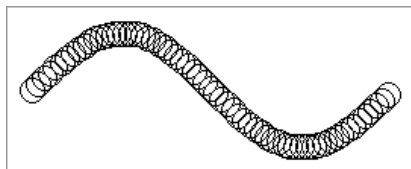

Figure 121: WHILE as a steering element for positions

FOR

FOR is another possibility for loops in GraphRep code with two variants.

The first form of **FOR** (*ForNumStatement*) repeats an element sequence according to a variable between two values. First the **from** value is assigned to the variable and the loop is executed. At the end

of the loop, the value of the variable is incremented by **by** (or by 1, if by is not specified). Afterwards the loop is executed again. This procedure is repeated until the value of the variable reaches or exceeds **to**.

Example:

```
GRAPHREP
SHADOW off
PEN style:null
SET count:30 size:0.2cm
FOR (x from:0 to:(count-1)
{
  FOR (y from:1 to:(count-1)
  {
    FILL color:(hsv2rgb(0, x / count, y / count))
    RECTANGLE x:(x * size) y:((count - y) * size) w:(size) h:(size)
  }
}
```
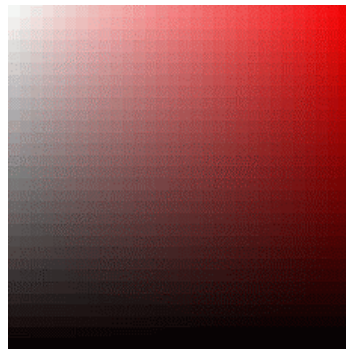

Figure 122: Example of a ForNumStatement

The second form of **FOR** (*ForTokenStatement*) repeats an element sequence according to a variable moving through a series of tokens. This token sequence is specified with **in**. Optionally, **sep** can specify a token separator (otherwise the blank is used as default).

Example:

```
GRAPHREP
SHADOW off
SET colors:"red yellow green cyan blue magenta"
SET x:0cm w:0.25cm
FOR c in:(colors)
{
  FILL color:(c)
  RECTANGLE x:(x) y:-0.25cm w:(w) h:0.5cm
  SET x:(x + w)
}
```


Figure 123: Example of a ForTokenStatement

PEN

<ins>*Pen :*</ins>	**PEN** [**style:***PenStyle*] [**w:***width*] [**color:***ColorSpecOrExpr*] [**endcap:***EndCap*] [**join:***Join*].
***PenStyle* :**	**solid** | **dot** | **dash** | **dashdot** | **inside-frame** | **null** .
***ColorSpecOrExpr* :**	*ColorSpec* | *intExpr* .
***EndCap* :**	**round** | **square** | **flat** .
***Join* :**	**bevel** | **miter** | **round** .

The **PEN** element sets the characteristics - width (**w**), style of line (**style**) and colour - of the "Pen". The current pen determines in which manner the lines of graphical elements are drawn. In the case of elements which can be filled, only the outline is drawn according to the current pen. The style element **FILL** (see p. 147) determines how the element is to be filled.

If no width or a width of 0cm is specified, the pen draws with the smallest possible width.

> **Hint:** Please notice that widths which barely differ often look alike on the screen, since a pixel is relatively wide here. Printers, however usually have a finer scaling so that even differences of 0.01cm width can be recognised.

The following line styles are available:

- **solid** - solid line (default setting),
- **dot** - dotted line,
- **dash** - broken / dashed line,
- **dashdot** - dashed and dotted line,
- **inside-frame** - solid line, whose whole width lies *within* the area boundaries (default: half of the breadth lies within the boundaries, half outside),
- **null** - invisible (i.e. no) line.

It is advised to switch the pen off (**style:null**) when a fillable shape is to be drawn without an outline.

The **endcap** parameter determines, how the ends of lines are drawn:
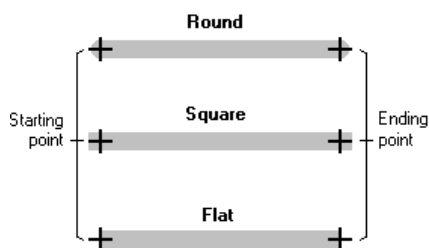
Figure 124: The endcap variants

The round and the square end extend over the start and end point of the line, the flat end is flush with the end of the line.

The **join** parameter determines, how the joint of two lines is drawn - **bevel**, **round** or **miter**:
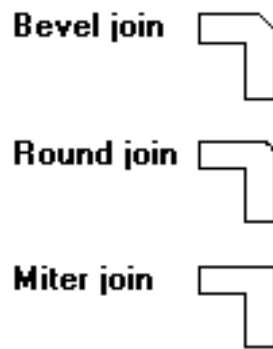
Figure 125: The join variants

Colours are either assigned by name or by RGB value. The colour names permitted are listed in the GRAPHREP syntax (see p. 134) (see "ColorName").

If you wish to assign a colour using an RGB value, a <u>decimal value between 0 (dark) and 255 (light)</u> or a <u>hexadecimal value between $00 (dark) and $FF (light)</u> must be assigned to three colour components (red, green, blue). According to the principle of additive colour synthesis the colour shade is then calculated.

The default colour is black (ColorName: "**black**" or. ColorRGB: "**r:0 g:0 b:0**").

> **Hint:** Default values of **PEN style:solid color:black** are preset.

Example 1: Line Types

```
GRAPHREP
SHADOW off
PEN w:0.15cm color:$bb9922 style:solid
FILL color:$ffee66
POLYGON 3 x1:-1.4cm y1:1.2cm x2:1.4cm y2:1.2cm x3:0cm y3:-1.2cm
PEN w:0.1cm color:red style:dot
LINE x1:-1.4cm y1:-0.6cm x2:1.4cm y2:-0.6cm
PEN w:0.1cm color:green style:dashdot
LINE x1:-1.4cm y1:0cm x2:1.4cm y2:0cm
PEN w:0.1cm color:blue style:dash
LINE x1:-1.4cm y1:0.6cm x2:1.4cm y2:0.6cm
```


Figure 126: Examples of the line types

Example 2: Impacts of endcap and join on a connector

```
GRAPHREP
PEN w:0.25cm color:$605030 endcap:round join:round
EDGE
END
POLYLINE 3 x1:-.8cm y1:.5cm x2:0.8cm y2:0cm x3:-.8cm y3:-.5cm
```

Figure 127: endcap:round join:round

```
GRAPHREP
PEN w:0.25cm color:$605030 endcap:flat join:bevel
EDGE
END
POLYLINE 3 x1:-.8cm y1:.5cm x2:0.8cm y2:0cm x3:-.8cm y3:-.5cm
```


Figure 128: endcap:flat join:bevel

```
GRAPHREP
PEN w:0.25cm color:$605030 endcap:square join:miter
EDGE
END
POLYLINE 3 x1:-.8cm y1:.5cm x2:0.8cm y2:0cm x3:-.8cm y3:-.5cm
```


Figure 129: endcap:square join:miter

The modelling bar shows only the outline of deactivated symbols. Only black pixels are regarded as outlines. By specifying the attribute **outline**, the current colour of the pen can be defined as an additional outline colour.

> **Hint:** The command **PEN style:inside-frame** is not supported in SVG Graphics (see chap. 9.14, p. 559). Instead, half of the line width is drawn within and half outside the border (default).

FILL

**Fill :**	**FILL** [**style:**_BrushStyle_] [**color:**_ColorSpecOrExpr_] [**fcolor:**_ColorSpecOrExpr_ | **transparent**] .
**BrushStyle :**	**solid** | **horz** | **vert** | **cross** | **diagcross** | **updiag** | **downdiag** | **mix25** | **mix50** | **mix75** | **null** .
**ColorSpecOrExpr :**	_ColorSpec_ | _intExpr_ .

The **FILL** element sets the characteristics of the fill brush: one pattern (**style**) and a maximum of two colours (**color**, **fcolor**). The current fill-in brush determines in which way the closed graphic elements - **RECTANGLE** (see p. 152), **POLYGON** (see p. 154), **COMPOUND** (see p. 157) - are filled in. Please note, that the edge is drawn with the current **PEN** (see p. 145).

The following filling patterns are available:

- **solid** - solid surface (default setting)
- **horz** - horizontally hatched surface
- **vert** - vertically hatched surface
- **cross** -horizontally and vertically hatched surface
- **diagcross** - diagonally upward and diagonally downward hatched surface
- **updiag** - diagonally upward hatched surface
- **downdiag** - diagonally downward hatched surface
- **mix25** - surface filled in by two mixed colours (25:75)
- **mix50** - surface filled in by two mixed colours (50:50)
- **mix75** - surface filled in by two mixed colours (75:25)
- **null** - invisible (i.e. no) filling

The main colour is specified with **color**. In the case of **solid** this colour is used as the filling colour and for all hatched patterns (all patterns apart from **solid** and **null**) this colour refers to that of the hatching lines. A second colour can be specified for the hatched patterns (which fills in the spaces between the lines). This second colour is specified by the keyword **fcolor**.

> **Hint:** If **FILL** is not specified, the default setting **FILL style:null** (transparent) applies.

<u>**Example:**</u>

In order to define a yellow filled area with blue hatching in an upward direction the following should be specified:

```
FILL style:updiag color:blue fcolor:yellow
```

The same can be obtained by

```
FILL style:updiag color:(rgbval(0,0,128)) fcolor:(rgbval(255,255,0))
```

or

```
FILL style:updiag color:$000080 fcolor:$ffff00
```

.

If the filling colour is to be set arbitrarily from a range of 16 million colour values depending on an instance attribute, this can be done with a combination of **AVAL** and **FILL**. The colour can be entered by the user as an RGB value or colour name. For the ADONIS Notebook, a special colour selection dialogue is available which can be provided via the class attribute "AttrRep" (see chap. 2.1.1.2, p. 171).

<u>**Example:**</u>

GraphRep:
```
AVAL fc:"Fill color"
FILL color:(fc)
```

AttrRep: `ATTR "Fill color" dialog:color`

> **Hint:** The command **FILL style:**BrushStyle is not fully supported in SVG Graphics (see chap. 9.14, p. 559). While **style:solid** and **style:null** are displayed correctly, all other styles are replaced with **solid**.

SHADOW

Shadow :

```
SHADOW OnOrOff .
```

OnOrOff :

```
on | off .
```

By default all graphic elements are drawn with a shadow. This can be switched off with **SHADOW off** and of course reactivated with **SHADOW on**.

STRETCH

Stretch :

```
STRETCH OnOrOff .
```

OnOrOff :

```
on | off .
```

If an object's size is changed (under **GRAPHREP** a **sizing** attribute is listed), all graphical elements of the object are stretched (or compressed) geometrically. With the style element **STRETCH**, geometric stretching can be switched off (**STRETCH off**) - and of course on again (**STRETCH on**).

Example:

```
GRAPHREP sizing:asymmetrical
RECTANGLE x:0cm y:0cm w:3cm h:2cm
STRETCH off
LINE x1:1cm y1:0cm x2:1cm y2:1cm
```

The graphical representation consists of a rectangle that is divided into two parts by a 1cm wide vertical line. The width of the left part is always 1cm, regardless of the current size of the object. The width of the right side, however, varies depending on the object's size.

START / MIDDLE / END

Start :

```
START .
```

Middle :

```
MIDDLE .
```

End :

```
END .
```

These keywords specify whether the following elements define the graphical representation of the starting point (**START**), the middle point (**MIDDLE**) or the end point (**END**) when defining relations. If **MIDDLE** is defined, the connector shows a movable central point.

EDGE

Edge :

```
          EDGE  .
```

The **EDGE** element is only used for relations and causes a relation's edge to be drawn. This is a poly-line reaching from the starting point via possible bendpoints up to the end point of a relation. The poly-line is drawn according to the current pen. The start and the end point can be defined separately (**START** or. **END** (see p. 149)).

If the **EDGE** element is not specified, it will be automatically inserted after the obligatory **GRAPHREP** element (see p. 138) - only, however, if the value **no-edge** has not been set. In the case of an auto-matic insertion, the edge is drawn with the default pen.

Example:

```
GRAPHREP
PEN color:blue
EDGE
```

A connector with a blue edge.

Example: Double-lined connector

Use the **EDGE** element twice, to create a double-lined connector:

```
GRAPHREP rounded:0.1cm
SHADOW off
PEN w:0.2cm color:blue
EDGE
PEN w:0.1cm color:white
EDGE
END
PEN w:0.2cm color:blue miterlimit:0.02cm join:round endcap:square
POLYLINE 3 x1:-.5cm y1:.1cm x2:0cm y2:0cm x3:-.5cm y3:-.1cm
PEN w:0.1cm color:white miterlimit:0.02cm join:round endcap:flat
POLYLINE 3 x1:-.5cm y1:.1cm x2:0cm y2:0cm x3:-.5cm y3:-.1cm
```

Figure 130: Example of a double-lined connector

CLIP_RECT / CLIP_ROUNDRECT / CLIP_POLY / CLIP_ELLIPSE / CLIP_OFF

ClipRect :

```
CLIP_RECT [ x:xposExpr ] [ y:yposExpr ]
[ w:widthExpr ] [ h:heightExpr ]
[ combine-mode:CombineMode ] .
```

ClipRoundRect :

```
CLIP_ROUNDRECT [ x:xExpr ] [ y:yExpr ]
[ w:wExpr ] [ h:hExpr ] [ rx:rxExpr ] [ ry:ryExpr ]
[ combine-mode:CombineMode ] .
```

ClipPoly **:**	`CLIP_POLY` *n*
	`[` **x1:**`xposExpr` `]` `[` **y1:**`yposExpr` `]`
	`...`
	`[` **x**n:`xposExpr` `]` `[` **y**n:`yposExpr` `]`
	`[` **combine-mode:**`CombineMode` `]` `.`
ClipEllipse **:**	`CLIP_ELLIPSE` `[` **x:**`xposExpr` `]` `[` **y:**`yposExpr` `]`
	`[` **rx:**`xradiusExpr` `]` `[` **ry:**`yradiusExpr` `]`
	`[` **combine-mode:**`CombineMode` `]` `.`
CombineMode **:**	`copy` `|` `and` `|` `or` `|` `diff` `|` `xor` `.`
ClipOff **:**	`CLIP_OFF` `.`

With clipping, further drawing of elements can be restricted to a given area (clipping region). All drawing which lies outside the current clipping region is ineffective, while all parts of elements inside are visible. The clipping region can be of any shape. Basic shapes are rectangle (**CLIP_RECT**), rounded rectangle (**CLIP_ROUNDRECT**), polygon (**CLIP_POLY**) and ellipse (CLIP_ELLIPSE). Basic shapes can be combined with different operators (and, or, diff, xor) to build complex clipping regions. A clipping region can even consist of disjoint parts.

At the beginning of a GraphRep drawing, no clipping is active. When reaching a clipping element, the current clipping region is set to the given shape. If clipping is already active, the new region can either replace the current clipping region (**combine-mode:copy**, which is the default) or can be combined with the current clipping region with one of the following combine mode operators:

● **and**: the resulting clipping region includes everything which belongs to both regions,

● **or**: the resulting clipping region includes everything which belongs at least to one of both regions,

● **diff**: the resulting clipping region is the current clipping region *minus* the given region or

● **xor**: the resulting clipping region includes everything which belongs to *exactly one* of both regions.

The other parameters have the same meaning as for their corresponding shape elements:

● **CLIP_RECT** -> **ROUNDRECT**

● **CLIP_ROUNDRECT** -> **RECTANGLE**

● **CLIP_POLY** -> **POLYGON**

● **CLIP_ELLIPSE** -> **ELLIPSE**

With **CLIP_OFF**, clipping is deactivated.

Example 1: CLIP_POLY

```
GRAPHREP
FILL color:red
ELLIPSE rx:0.8cm ry:0.8cm
CLIP_POLY 4
  x1:-1.5cm y1:0.7cm x2:-1.5cm y2:0.2cm
  x3:1.5cm y3:-0.7cm x4:1.5cm y4:-0.2cm
FILL color:white
ELLIPSE rx:0.8cm ry:0.8cm
```


Figure 131: Example of CLIP_POLY

Beispiel 2: CLIP_ELLIPSE / CLIP_RECT

```
GRAPHREP
SHADOW off
PEN style:null
FILL color:$008833
RECTANGLE x:0cm y:0cm w:2cm h:2cm
FONT "Arial" h:60pt line-orientation:10 bold underline
TEXT "TestTest" x:-3.5cm y:1cm
CLIP_ELLIPSE x:0cm y:2cm rx:1.5cm ry:1.5cm
CLIP_ELLIPSE x:2cm y:2cm rx:1.5cm ry:1.5cm combine-mode:xor
CLIP_RECT x:-0.5cm y:1.3cm w:3cm h:0.2cm combine-mode:diff
CLIP_RECT x:-0.5cm y:1.9cm w:3cm h:0.2cm combine-mode:diff
CLIP_RECT x:-0.5cm y:2.5cm w:3cm h:0.2cm combine-mode:diff
FILL color:$0099ff
RECTANGLE x:-2cm y:0cm w:6cm h:4cm
```


Figure 132: Example of CLIP_ELLIPSE / CLIP_RECT

Hint: The modes **combine-mode:and**, **combine-mode:diff**, and **combine-mode:xor** für **CLIP-Befehle** are not allowd in SVG Graphics (see chap. 9.14, p. 559). Instaead of combinations, separate clippings have to be defined.

POINT / LINE

Point :

> **POINT** [**x:***xposExpr*] [**y:***yposExpr*] .

Line :

> **LINE** [**x1:***xposExpr*] [**y1:***yposExpr*]
> [**x2:***xposExpr*] [**y2:***yposExpr*] .

With **POINT** a point is drawn at the position (**x**, **y**), with **LINE** a line is drawn from (**x1**, **y1**) to (**x2**, **y2**) using the current pen.

RECTANGLE

Rectangle :

> **RECTANGLE** [**x:***xposExpr*] [**y:***yposExpr*]
> [**w:***widthExpr*] [**h:***heightExpr*] .

A rectangle is drawn with (**x**, **y**) as upper left corner and with a width **w** and height **h**. The left and the right side of the rectangle are always parallel to the y-axis, the other two sides thus parallel to the x-axis.

> **Hint:** Since **RECTANGLES** cannot be rotated, we recommend that you use the **POLYGON** (see p. 154) in combination with relations.

ROUNDRECT

RoundRect :

```
ROUNDRECT [ x:xExpr ] [ y:yExpr ]
    [ w:wExpr ] [ h:hExpr ]
    [ rx:rxExpr ] [ ry:ryExpr ] .
```

ROUNDRECT allows the display of a rectangle with rounded corners and with (**x**, **y**) as the upper left corner, width **w** and height **h**. The left and the right side of the rectangle are always parallel to the y-axis, the other two sides thus parallel to the x-axis. Both parameters **rx** and **ry** give (elliptical) radii of curvature of corners.

> **Hint:** Because of the fact, that **ROUNDRECT** cannot be rotated, it is recommended to use **CURVE** (see p. 156) in case of relations.

POLYLINE

PolyLine :

```
POLYLINE n
    [ x1:xposExpr ] [ y1:yposExpr ]
    ...
    [ xn:xposExpr ] [ yn:yposExpr ] .
```

With **POLYLINE** a path of several joined lines can be drawn in one go. *n* is the number of points to be connected with the POLYLINE.

A succession of lines is drawn from (**x1**, **y1**) via (**x**i, **y**i) to (**x**n, **y**n). Note that $1<i<n$.

Example:

```
POLYLINE 3 x1:-1cm y1:-1cm x2:0cm y2:1cm x3:1cm y3:-1cm
```

draws a "V" consisting of two joined lies.

BEZIER

Bezier :

```
BEZIER n
    [ x1:xposExpr ] [ y1:yposExpr ]
    ...
    [ xn:xposExpr ] [ yn:yposExpr ] .
```

BEZIER draws a bezier curve with *n* points or *k* Bezier segments. Note that:

- $n>=4$ und n MOD 3 = 1

- $k = (n - 1) / 3$

The drawn curve goes through the points 1, 4, 7, 10, ..., *n*; the others are control points which pull the curve into their directions.

Example:

```
GRAPHREP
SHADOW off
PEN color:$115599 w:0.2cm
FILL color:$5599dd
ELLIPSE rx:1.5cm ry:1.5cm
PEN color:white w:0.2cm endcap:flat
FOR i from:0 to:2 {
  SET y:(-0.5cm + i * 0.5cm)
  BEZIER 4 x1:-1cm y1:(y) x2:-0.25cm y2:(y - 1cm)
          x3:0.25cm y3:(y + 1cm) x4:1cm y4:(y)
}
```


Figure 133: Example of Bezier curves

POLYGON

Polygon :

```
POLYGON n
    [ x1:xposExpr ] [ y1:yposExpr ]
    ...
    [ xn:xposExpr ] [ yn:yposExpr ] .
```

With **POLYGON**, a closed form of lines is drawn. The polygon is drawn with *n* corners - the points (**x***i*, **y***i*) - (1<=*i*<=*n*).

Example:

```
POLYGON 3 x1:-1cm y1:-1cm x2:0cm y2:1cm x3:1cm y3:-1cm
```

A triangle is drawn.

> **Hint:** In contrast to the POLYLINE, the first and the last points will be connected to form a closed shape.

ELLIPSE

Ellipse :

```
ELLIPSE [ x:xposExpr ] [ y:yposExpr ]
      [ rx:xradiusExpr ] [ ry:yradiusExpr ] .
```

An ellipse with a centre (**x**, **y**) is drawn. The radius **rx** and **ry** specify the distance from the centre to the horizontal (rx) or to the vertical (ry) edges respectively. If the same value is assigned to **rx** and **ry**, a circle with the corresponding radius will be drawn.

> **Hint:** Rotation around the object's centre which would be necessary for drawing relations is not possible. Therefore, you should only use circles (ellipses with two identical radii) in combination with relations.

Example:

The ellipse (see fig. 134) represented in the picture below is drawn by

```
ELLIPSE x:0.5cm y:0cm rx:2.5cm ry:2cm
```
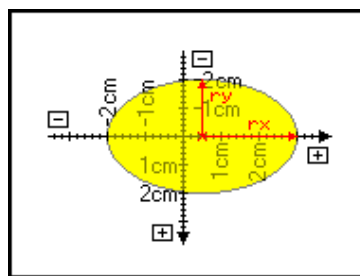

Figure 134: Example of an ellipse

ARC / PIE

Arc :
```
ARC [ x:xposExpr ] [ y:yposExpr ]
    [ rx:xradiusExpr ] [ ry:yradiusExpr ]
    [ x1:xposExpr ] [ y1:yposExpr ]
    [ x2:xposExpr ] [ y2:yposExpr ] .
```

Pie :
```
PIE [ x:xposExpr ] [ y:yposExpr ]
    [ rx:xradiusExpr ] [ ry:yradiusExpr ]
    [ x1:xposExpr ] [ y1:yposExpr ]
    [ x2:xposExpr ] [ y2:yposExpr ] .
```

Arcs (**ARC**) and segments (**PIE**) are based on ellipses, in which semi-straight lines - originating in the ellipse's centre (**x**, **y**) - define a sector. The first semi-straight line takes its course through point (**x1**, **y1**), the second through point (**x2**, **y2**).

In the case of **ARCS** only the ellipse's edge of this sector is drawn. For **PIES** the edge including the semi-straight line from the centre to the edge is drawn. A pie shape can also be filled in using the current fill-in brush **FILL** (see p. 147).

Example:

The pie below (see fig. 135) is drawn with

```
GRAPHREP
PIE x:0.5cm y:0cm rx:2.5cm ry:2cm x1:-2cm y1:2cm x2:3cm y2:0cm
```
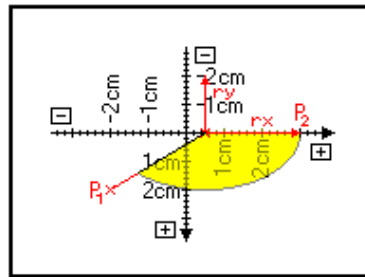
Figure 135: Example of an ellipse segment

CURVE

Curve :

```
CURVE varName from:realVal to:realVal
     fx:realExpr fy:realExpr .
```

The **CURVE** element enables you to draw two-dimensional function curves specified by parameters. The key-word **CURVE** is followed by the name of a variable, which runs through the interval specified by **from** and **to**. Since the width of a step is always exactly one pixel and thus depends on the output medium and the scaling factor chosen, in principle, a clear representation is guaranteed, particularly when enlarging objects.

For **fx** and **fy** LEO expressions have to be specified. These should not access variables other than the one specified after the keyword **CURVE** ("varName"; in the following examples "t"). Accessing other variables is possible, but terribly slow.

Example:

The symbol (see fig. 136) shown in the picture/image below is drawn by

```
GRAPHREP
CURVE "t" fx:(sin(.77*t+sin(t))) fy:(.8*cos(t))
from:-3.5 to:3.5
```


Figure 136: Example of a curve

You can also draw curved surfaces. To do this, use **CURVE** in a **COMPOUND** sequence (see p. 157). The sequence may even consist of only one element - the curve itself.

Example:

The symbol (see fig. 137) shown in the picture below is drawn by

```
GRAPHREP
COMPOUND 1
```

```
CURVE "t" fx:(sin(t)+.1*sin(8*t)) fy:(cos(t)-.1*cos(8*t))
from:0 to:6.28
```
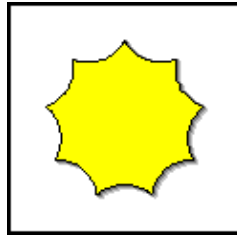

Figure 137: Example of an area composed by curves

COMPOUND

Compound :
>   **COMPOUND** *n* { *CompoundableElem* } .

***CompoundableElement* :**
>   *Line │ PolyLine │ Curve* .

The **COMPOUND** element helps you to create shapes from *n* other elements. You can use the following elements to do this: **LINE** (see p. 152), **POLYLINE** (see p. 153) and **CURVE** (see p. 156).

The **COMPOUND** element is especially useful combined with the **CURVE** element, as otherwise **POLYGON** would produce the same result. The number *n* following the keyword **COMPOUND** indicates how many of the following elements belong to the compound. The compound's outline results from the *n* elements, where the end point of each element is connected to the start point of the following element, unless the two points are congruent. The last end point is connected to the first start point in such a way that the result is always a closed figure. Please note therefore, that the drawing direction of the single elements is especially important.

Example:

The symbol (see fig. 138) represented in the picture below is drawn by

```
GRAPHREP
COMPOUND 2
LINE x1:1cm y1:-.7cm x2:-1cm y2:-.7cm
CURVE "t" f:(t) g:(-.2*sin(3.14*(t+1))+.7) from:-1 to:1
```

The connecting lines between LINE and CURVE are automatically drawn by COMPOUND.
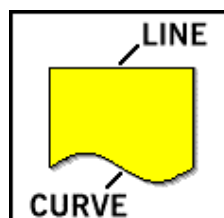
Figure 138: Example of a combination of graphical elements (line and curve)

Example:

The symbol (see fig. 139) represented in the picture below is drawn by

```
GRAPHREP
FILL color:yellow
COMPOUND 4
CURVE "t" fx:(.8+.3*sin(t)) fy:(.5+.3*cos(t)) from:0 to:1.57
CURVE "t" fx:(.8+.3*sin(t)) fy:(-.5+.3*cos(t)) from:1.57 to:3.14
CURVE "t" fx:(-.8+.3*sin(t)) fy:(-.5+.3*cos(t)) from:3.14 to:4.71
CURVE "t" fx:(-.8+.3*sin(t)) fy:(.5+.3*cos(t)) from:4.71 to:6.28
```
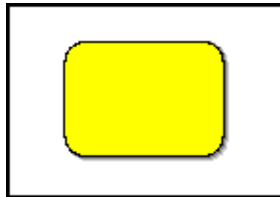
Figure 139: Example of combined graphical elements (curves)

Example:

The symbol (see fig. 140) represented in the picture below is drawn by

```
GRAPHREP
FILL color:yellow
COMPOUND 4
CURVE "t" fx:(.8-.3*sin(t)) fy:(.8-.3*cos(t)) from:0 to:1.57
CURVE "t" fx:(-.8-.3*sin(t)) fy:(.8-.3*cos(t)) from:4.71 to:6.28
CURVE "t" fx:(-.8-.3*sin(t)) fy:(-.8-.3*cos(t)) from:3.14 to:4.71
CURVE "t" fx:(.8-.3*sin(t)) fy:(-.8-.3*cos(t)) from:1.57 to:3.14
```
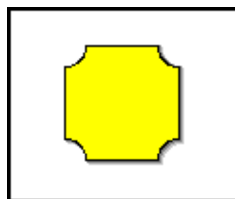
Figure 140: Example of combined graphical elements (curves)

> **Hint: ROUNDRECT (see p. 152)** is recommended for rectangles with rounded corners.

BITMAP / BITMAPINFO

Bitmap :

```
BITMAP fileNameExpr
      x:xposExpr y:yposExpr
      w:widthExpr h:heightExpr .
```

BitmapInfo :

```
BITMAPINFO fileNameExpr .
```

With the **BITMAP**-element a graphic file can be added to the graphic object representation. The file will be loaded from the file system or the ADONIS database via the file name *FileNameExpr*. The quotation of the path and the file names of the graphic file have to be in quotation marks . If loading the file is not possible, the graphical representation will be shown as if no file was specified.

> **Hint:** The back slashes "\" in the path quotation of the graphic file must be masked with '\' (e.g. "c:\\bitmaps\\logo.bmp" for the file "logo.bmp" in the directory "bitmaps" on the disk "C:").

> **Hint:** For the inclusion of bitmap-files which are saved as external files in the ADONIS database (see chap. 15., p. 606), use '**db:**' as path (e.g. "db:\\logo.bmp" for a file in the database saved under "logo.bmp").

The BMP file will be drawn as a square (**x**, **y**) as the above left corner the width **w** and the height **h**.

> **Hint:** If the view relation (width to height) between the bmp file and the defined square are different the graphic will be shown distorted.

Through the **BITMAPINFO** element the size (width and height in pixels) of a graphic file will be read. The determined size will be assigned to the variables **bmpwidth** (for the width) and **bmpheight** (for the height) and can be used for the definition of the square in which the graphic file will be represented. This way, the file can be represented without distortion.

Supported file formats:

(colour depth in parentheses)

"bmp"
> Windows Bitmap (1, 4, 8, 16, 24, 32)

"gif"
> Graphics Interchange Format (8)

"ico"
> Windows Icon (1, 4, 8, 16, 24, 32)

"jpeg"
> JPEG Graphics Format (8, 24)

"png"
> Portable Network Graphics (1, 4, 8, 24, 32)

"targa"
> Truevision Targa (8, 16, 24, 32)

"tiff"
> Tagged Image File Format (1, 4, 8, 24, 32)

"wbmp"
> Wireless Bitmap (1)

"xpm"
> X11 Pixmap (24)

Example: How to display a bitmap file undistorted

This GraphRep code reads a file name from an instance attribute "Bitmap file", loads the bitmap and displays it with 96dpi resolution.

```
GRAPHREP
```

```
AVAL filename:"Bitmap file"
BITMAPINFO (filename)
BITMAP (filename) x:0cm y:0cm w:(bmpwidth / 96 * 2.54cm) h:(bmpheight / 96 *
2.54cm)
```

If the size of the object doesn't just depend on the graphic size but also shall be modifiable by the user without distorting the picture, the first line has to be changed to

```
GRAPHREP sizing:keep-aspect-ratio.
```

Example: How to display a bitmap file undistorted within an asymmetrical, sizeable object

```
GRAPHREP sizing:asymmetrical
AVAL grfk:("Bitmap file")
SET d:3cm  # default width and height
PEN color:lightgray
FILL style:null
RECTANGLE w:(d) h:(d)  # a border marking the object's size
TABLE w:(d) h:(d) rows:1 cols:1 w1:100% h1:100%  # get the current size of the
object
BITMAPINFO (grfk)  # get the bitmap size
STRETCH off
IF (bmpwidth / CMS tabw1 < bmpheight / CMS tabh1)
{
  # use maximum height, space left and right
  SET w:(tabh1 * (bmpwidth / bmpheight))
  BITMAP (grfk) x:((tabw1 - w) / 2) y:0cm w:(w) h:(tabh1)
}
ELSE
{
  # use maximum width, space at top and bottom
  SET h:(tabw1 * (bmpheight / bmpwidth))
  BITMAP (grfk) x:0cm y:((tabh1 - h) / 2) w:(tabw1) h:(h)
}
```

GRADIENT_RECT / GRADIENT_TRI

GradientRect :

GRADIENT_RECT
[**x:***xExpr*] [**y:***yExpr*] [**w:***wExpr*] [**h:***hExpr*]
[**mode:***ColorSpecOrExpr*]
[**color1:***ColorSpecOrExpr*] [**color2:***ColorSpecOrExpr*]
[**color3:***ColorSpecOrExpr*] [**color4:***ColorSpecOrExpr*] .

GradientMode **:**

horz | **vert** | **updiag** | **downdiag** | **diagcross** .

ColorSpecOrExpr **:**

ColorSpec | *intExpr* .

GradientTri :

GRADIENT_TRI
[**x1:***xExpr*] [**y1:***yExpr*] [**color:***ColorSpecOrExpr*]
[**x2:***xExpr*] [**y2:***yExpr*] [**color:***ColorSpecOrExpr*]
[**x3:***xExpr*] [**y3:***yExpr*] [**color:***ColorSpecOrExpr*] .

With **GRADIENT** commands, rectangles and triangles can be filled with continuous colour gradients.

> **ATTENTION:** Gradients can *not* be displayed in Windows 95 and Windows NT environments. There, the average colour tone will be shown instead.

GRADIENT_RECT:

GRADIENT_RECT fills a rectangle with smooth, shaded colours. The colouring is defined with **mode** and two or four basic colours. These gradient modes are possible:

- **horz**: from the left with colour 1 to the right with colour 2

- **vert**: from the top with colour 1 to the bottom with colour 2

- **updiag**: from the bottom left corner with colour 1 to the top right corner with colour 2

- **downdiag**: from the top left corner with colour 1 to the bottom right corner with colour 2

- **diagcross**: four-colour shading, one colour per corner: colour 1 top left, colour 2 top right, colour 3 bottom right and colour 4 bottom left corners

GRADIENT_RECT does *not* automatically draw a border around the gradient. For this, two more elements are needed: **FILL style:null** followed by a **RECTANGLE** with the same coordinates as GRADIENT_RECT.

Example:

```
GRAPHREP
GRADIENT_RECT x:-1.4cm y:-.7cm w:2.8cm h:1.4cm style:downdiag
  color1:$666699 color2:$444466
PEN w:0.05cm
FILL style:null
RECTANGLE x:-1.4cm y:-.7cm w:2.8cm h:1.4cm
```


Figure 141: Example of a gradient

Example:

```
GRAPHREP
GRADIENT_RECT x:-1.4cm y:-.7cm w:2.8cm h:1.4cm style:updiag
  color1:yellow color2:red
PEN w:0.05cm
FILL style:null
RECTANGLE x:-1.4cm y:-.7cm w:2.8cm h:1.4cm
```
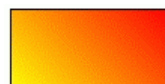

Figure 142: Example of a gradient

GRADIENT_TRI:

GRADIENT_TRI fills a triangle area with smooth shaded colours. This area is defined by three points (**x1**, **y1**), (**x2**, **y2**) and (**x3**, **y3**). For each of these corners a basic colour is defined (**color1** through **color3**). The rest of the triangle is filled with the appropriate combination colour. By combining several GRADIENT_TRI elements, any colouring of a polygon can be achieved (see example below).

Example:

```
GRAPHREP
SET x:({-2cm, 0cm, 2cm, 2cm, 0cm, -2cm})
SET y:({-0.8cm, -2cm, -0.8cm, 0.8cm, 2cm, 0.8cm})
SET c:({$ff00ff, $0000ff, $00ffff, $00ff00, $ffff00, $ff0000})
GRADIENT_TRI x1:(x[0]) y1:(y[0]) color1:(c[0])
             x2:(x[1]) y2:(y[1]) color2:(c[1])
             x3:(x[2]) y3:(y[2]) color3:(c[2])
GRADIENT_TRI x1:(x[0]) y1:(y[0]) color1:(c[0])
             x2:(x[2]) y2:(y[2]) color2:(c[2])
             x3:(x[5]) y3:(y[5]) color3:(c[5])
GRADIENT_TRI x1:(x[2]) y1:(y[2]) color1:(c[2])
             x2:(x[3]) y2:(y[3]) color2:(c[3])
             x3:(x[5]) y3:(y[5]) color3:(c[5])
GRADIENT_TRI x1:(x[3]) y1:(y[3]) color1:(c[3])
             x2:(x[4]) y2:(y[4]) color2:(c[4])
             x3:(x[5]) y3:(y[5]) color3:(c[5])
SHADOW off
PEN w:0.07cm
POLYGON 6 x1:(x[0]) y1:(y[0]) x2:(x[1]) y2:(y[1]) x3:(x[2]) y3:(y[2])
          x4:(x[3]) y4:(y[3]) x5:(x[4]) y5:(y[4]) x6:(x[5]) y6:(y[5])
```
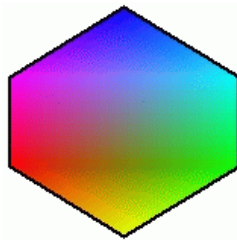

Figure 143: Example of a gradient

Hint: The command **GRADIENT_RECT style:diagcross** and all styles of **GRADIENT_TRI** are not fully supported in SVG Graphics (see chap. 9.14, p. 559). Instead of the original ones, SVG-specific algorithms are used, yielding very similar results.

FONT

Font :
      FONT [*strExpr*] [**h:***measureExpr*] *FontStyle*
        [**color:***ColorSpecOrExpr*] [**line-orientation:***realExpr*] .

FontStyle :
      style:*strExpr* | *DirectFontStyle* .

DirectFontStyle :
      [**bold**] [**underline**] [**italic**] .

ColorSpecOrExpr :
      ColorSpec | *intExpr* .

The **FONT** element determines which type of font is used for the following **TEXT** and **ATTR** elements (see p. 164). The term following the key-word **FONT** - such as "Helvetica", "Times New Roman", "Arial" and so on - specifies the font. The font's height is specified by the attribute **h**. Usually, the font

height is specified in pt, but sometimes it is possible to choose between cm and pt - as everywhere else measures are specified in LEO.

> **Hint:** All fonts available in the operating system can be used. Default is "Helvetica" 10pt.

The font style can be defined in two ways. Either the **style** attribute is used, or the parameters **bold**, **underline** and **italic** can be entered directly. These **style** tokens are recognised:

- **bold** - bold
- **underline** - underlined
- **strikeout** - crossed out
- **italic** - italics
- **outline** - outlined
- **shadow** - shadowed

These tokens can be used in any combination. However, **shadow** suppresses **outline**.

Example:

```
FONT "Times New Roman" h:32pt
TEXT "V" x:0cm y:-.07cm w:c h:c
```

draws a "V" in the font type "Times New Roman" with a font height of 32pt, centred at point (0,0 cm,-0,07cm).

For font name, fonts style and font size, an *expression* can be specified so that these parameters can be set according to corresponding instance attributes.

Example:

```
AVAL fn:"font name"
AVAL fh:"font height (in points)"
FONT (fn) h:(PT fh)
```

With **line-orientation** the text can be rotated. The value of this parameter determines the rotation angle (*anticlockwise*) for **TEXT** or **ATTR**. 0 (default) means horizontally from left to right. Rotated text can be combined with all other options (hyperlinks, absolute coordinates etc.).

Example:

```
GRAPHREP
SHADOW off
PEN w:0.1cm color:$000080
FILL color:$6060d0
RECTANGLE x:0cm y:-1cm w:2.8cm h:2.4cm
LINE x1:1cm y1:-1cm x2:1cm y2:1.4cm
FONT "Arial" h:24pt line-orientation:90 style:"outline" color:$000080
TEXT "Hello!" y:1.2cm
FONT "Arial" h:12pt bold line-orientation:45
TEXT "Description\nLine 2" x:-2cm y:-0.5cm w:3cm
```
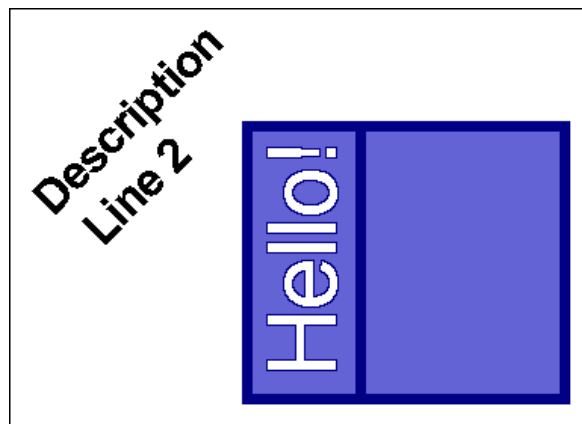
Figure 144: Rotated text

The colour assignment (**color**) is executed through the name or the rgb values. The allowed color names are listed in the GraphRep grammar (see p. 134) (s. "ColorName") and are according to theLEO colour names (see chap. 17.3.10.1, p. 681).

For the color assignment over the rgb values there are always three color components used (red, green, blue); <u>decimal value from between 0 (dark) and 255 (bright)</u> or <u>hexadecimal value from between $00 (dark) and $FF (bright)</u>, from which according to the principle of the additive color synthesis the mixed colors will be calculated.

The standard colour is black (ColorName: "**black**" or ColorRGB: "**r:0 g:0 b:0**").

Example:

If the text should be bold and written in light green you have to quote the following:

```
FONT bold color:lightgreen
```

The same is achieved with:

```
FONT bold r:0 g:255 b:0
```

or

```
FONT bold r:$00 g:$FF b:$00
```

> **Hint:** The command **FONT style:shadow** is not supported in SVG Graphics (see chap. 9.14, p. 559). Instead, the text is shown without shadow.

TEXT / ATTR

Text :
```
TEXT strExpr [ line-break:LineBreakMode ]
        [ TextXCoord ] [ TextYCoord ]
        [ TextWidth ] [ TextHeight ] [ LineHeight ] .
```

Attr :
```
ATTR attrName [ text:strExpr ]
        [ row:intExpr ] [ col:attrName ]
        [ format:strValue ] [ sep:strValue ] [ line-
break:LineBreakMode ]
        [ TextXCoord ] [ TextYCoord ]
        [ TextWidth ] [ TextHeight ] [ LineHeight ] .
```

LineBreakMode :
```
off | words | rigorous .
```

***TextXCoord* :**	`x`[`:abs`]`:`*xposExpr* `.`
***TextYCoord* :**	`y`[`:abs`]`:`*yposExpr* `.`
***TextWidth* :**	`w:`*widthExpr* | `w:`*WAlign*[`:`*widthExpr*] `.`
***WAlign* :**	`l` | `c` | `r` `.`
***TextHeight* :**	`h:`*heightExpr* | `h:`*HAlign*[`:`*heightExpr*] `.`
***HAlign* :**	`t` | `c` | `b` `.`
***LineHeight* :**	`line-height:`*heightExpr* `.`

The keyword **TEXT** causes the text specified to be displayed in the location specified ("fixed text"). In contrast, the keyword **ATTR** allows you to display the contents of an instance attribute from the notebook of an object ("visualised instance attribute"). If a value is displayed using ATTR, a click on it opens an input box allowing the user to change the value. Exceptions are attributes of the types **INTERREF** (see chap. 5.12, p. 539) and **PROGRAMCALL** (see p. 176): Here the visualised attribute serves as a hyperlink to the target.

The name of the instance attribute to be visualised must be specified by a constant string. For **TEXT** it is also possible to enter an expression. Fixed text is in a way an ordinary graphic element. Therefore, changing the font size in the graphical model editor (menu "View") does effect displayed attribute values, but not fixed text.

Example:

```
AVAL k:"Comment"
TEXT (copy (k, 0, 10))
```

Here, the first ten characters of the value of the notebook attribute "Comment" are displayed. Since the text displayed in the model editor behaves like an ordinary graphic element, clicking on the text selects an object but does not call an input window in which a value for "Comment" could be entered.

In the case of **ATTR** the attribute **text** defines what is actually shown in the representation. Mostly, this will be exactly the value of the instance attribute to be visualised. In this case, you do not have to specify **text**. However, if a different text is to be represented, **text** must be specified by a chain of symbols or an expression resulting in a chain of symbols such as the result of a calculation in which the value of the instance attribute visualised here is also considered.

Example:

```
AVAL k:"Comment"
ATTR "Comment" text:(copy (k, 0, 10))
```

As in the example above only the first ten symbols of the instance attribute "Comment's" value are displayed. Here, though, you can - by clicking on the visualised text - call an input window for entering a value for the attribute "Comment". In this input window the complete attribute value is displayed.

A speciality of the **attributes from the type INTEREF** (see chap. 5.12, p. 539) is that for the representation a format specification can be quoted. This is executed through the quotation of the attribute **format** and a text (*strValue*), which describes how the reference can be represented. The following listed place holders are replaced accordingly:

- **%o** through the object names,
- **%c** through the class names,
- **%m** through the model names,

- **%t** through the model type names;

- **%v** through the internal version number (in format: YYYY:MM:TT),

- **%V** through the internal version number defined in the format of versioning.

    Hint: The place holders **%M**, **%v** and **%V** will be analysed only in application libraries with time-related versioning (see chap. 6.1.2, p. 69).

Additionally, you can change the display of references and their hyperlinks' functionality of multivalued **attributes of type INTERREF** (see chap. 5.12, p. 539). It is possible through specification of **sep** and separators (*strValue*). Standard presentation of more than one reference at the same time means that each reference will be displayed in a separate line. It responds to standard separators "\r\n".

Examples:

```
ATTR "Referenced cost department" format:"%o"

ATTR "called process" format:"%m (%t)"
```

In the first case the object name (the referenced cost department), in the second case the model name (of the called process) and the model type in brackets will be quoted.

There are nine possibilities to align the lines of the text displayed. These result from the possible combinations of the horizontal ("l" - left, "c" - centred, "r" - right) with the vertical ("t" - top, "c" - centred, "b" - bottom) alignment. The type of alignment determines how the rectangle including the whole text is positioned in relation to the text's position (**x**, **y**). The lines are aligned according to the horizontal alignment specified within this rectangle.

The following possibilities exist for the automatic **line-break**: If it is set to **words** or **rigorous**, lines longer than the width of the concerned text box will be broken in front of the first word which goes beyond the permitted width. However, a line may consist of a single word longer than the width of the text box. In the case of **line-break:words** such lines reach beyond the text box, in the case of **line-break:rigorous** the word is broken in front of the first letter which goes beyond the permitted width. You can completely de-activate the automatic line-break using **line-break:off**. The default setting is **line-break:words**.

An **x**- or. **y**-coordinate can be **TEXT** and **ATTR** quoted absolutely - through setting the modifier **abs**. It refers not to the object's position but to the left (**x**) or above (**y**) edge of the drawing area. Therefore it can be determined that an attribute is always in a determined column on the drawing area and is displayed independent of how far left or right the object is.

With **row** and **col** a specific cell in an attribute of type *Table* (RECORD) can be accessed.

Example:

```
AVAL set-count-rows rowcount:"Responsible Persons"
FOR i from:1 to:(rowcount)
{
  ATTR "Responsible Persons" row:(i) col:"Department"
}
```

TEXTBOX / ATTRBOX

TextBox :   **TEXTBOX** *strExpr* [**line-break:***LineBreakMode*]
                    [*TextXCoord*] [*TextYCoord*]

	[*TextWidth*] [*TextHeight*] [*LineHeight*] .
<u>*AttrBox :*</u>	**ATTRBOX** *attrName* [**text:***strExpr*]
	[**row:***intExpr*] [**col:***attrName*]
	[**format:***strValue*] [**sep:***strValue*] [**line-break:***LineBreakMode*]
	[*TextXCoord*] [*TextYCoord*]
	[*TextWidth*] [*TextHeight*] [*LineHeight*] .
LineBreakMode **:**	**off** │ **words** │ **rigorous** .
TextXCoord **:**	**x**[**:abs**]**:***xposExpr* .
TextYCoord **:**	**y**[**:abs**]**:***yposExpr* .
TextWidth **:**	**w:***widthExpr* │ **w:***WAlign*[**:***widthExpr*] .
WAlign **:**	**l** │ **c** │ **r** .
TextHeight **:**	**h:***heightExpr* │ **h:***HAlign*[**:***heightExpr*] .
HAlign **:**	**t** │ **c** │ **b** .
LineHeight **:**	**line-height:***heightExpr* .

TEXTBOX and **ATTRBOX** are elements which do not produce direct output to the drawing area. They are designed to calculate the needed (rectangle) area for **TEXT** or **ATTR** elements (see p. 164).

TEXTBOX and **ATTRBOX** use the same parameters as the corresponding **TEXT** and **ATTR** elements. The parameters resulting from the calculation are assigned to these variables:

textx1 x-coordinate of the left margin of the text box

texty1 y-coordinate of the above margin of the text box

textx2 x-coordinate of the right margin of the text box

texty2 y-coordinate of the under margin of the text box

textw width of the textbox

texth height of the text box

<u>**Example:**</u>

Two or more line text attributes in different text colours should be displayed one under the other while between the text boxes - independent of the attribute values and their row number - a distance of 0.4cm should be kept.

```
GRAPHREP
# ...
FONT color:seagreen
ATTR "Attribute 1" x:2cm y:-1cm w:4cm
ATTRBOX "Attribute 1" x:2cm y:-1cm w:4cm
  # now texty2 is the bottom of the ATTR above
FONT color:crimson
ATTR "Attribute 2" x:2cm y:(texty2 + 0.4cm) w:4cm
```

The **attributes of type INTERREF** (see chap. 5.12, p. 539), for which format specification can be defined, are known as attributes with special features. This means in this case that each user can define by himself the **format** and text (*strValue*) describing how references should be presented. As a result, place holders from the list are replaced as follows:

- **%o** through the object names,

- **%c** through the class names,

- **%m** through the model names,

- **%M** through the model names and version numbers,

- **%t** through the model type names,

- **%v** through the internal version number (in format YYYY:MM:TT),

- **%V** through the internal version number defined in the format of versioning.

   Hint: The place holders **%M**, **%v** and **%V** will be analysed only in application libraries with time-related versioning (see chap. 6.1.2, p. 69).

Additionally, you can change the display of references and their hyperlinks' functionality of multivalued **attributes of type INTERREF** (see chap. 5.12, p. 539). It is possible through specification of **sep** and separators (*strValue*). Standard presentation of more than one reference at the same time means that each reference will be displayed in a separate line. It responds to standard separators "\r\n".

row and **col** enable access to distinct cells in attributes of type *Table* (RECORD).

ATTRBOX can be accessed not only in instance attributes, but also in class attributes. However, because changing of class attributes while modelling (in the notebook) is not possible, clicking on a class attribute opens only a Viewer.

HOTSPOT

Hotspot :   **HOTSPOT** *attrName* [**text:***strExpr*]
                          [**row:***intExpr*] [**col:***attrName*]
                          [**x:***xposExpr*] [**y:***yposExpr*]
                          [**w:***widthExpr*] [**h:***heightExpr*] .

A **HOTSPOT** element enables the connection of a right squared area within the GRAPHREP-definition with an object attribute. The value will be defined through the values **x** and **y** (left above corner) as well as **w** (width) and **h** (height), and with the quoted attribute (*attrName*) connected.

The square will not be displayed in the graphical representation. If the mouse pointer in the model graphic will be moved over the square its appearance changes   , if a hotspot of the attribute type "Reference" (see chap. 5.12, p. 539) or "Program call" (see chap. 5.11, p. 539) is contained. Additionally the value of the quoted attribute or a defined text (**text**) will be faded in.

On clicking the hotspot, the same action as when clicking an **ATTR** element (see p. 164) of the same attribute type is triggered.

Example 1: Hotspot of an INTERREF attribute

```
GRAPHREP
PEN w:0.05cm
FILL color:dodgerblue
RECTANGLE x:-1.4cm y:-.7cm w:2.8cm h:1.4cm
```

```
AVAL vr:"Responsible role" # responsible role
IF (LEN vr)
  FILL color:lightblue
  ELLIPSE x:1.4cm y:-0.3cm rx:0.35cm ry:0.25cm
  FONT h:0.4cm
  TEXT "R" x:1.4cm y:-0.26cm w:c h:c
  HOTSPOT "Responsible role" x:1.05cm y:-0.55cm w:0.7cm h:0.5cm
ENDIF
```
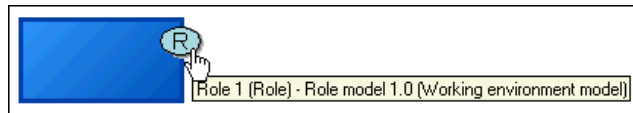

Figure 145: Example of an INTERREF hotspot

Example 2: Hotspot of a PROGRAMCALL attribute

```
GRAPHREP
PEN w:0.05cm
FILL color:dodgerblue
RECTANGLE x:-1.4cm y:-.7cm w:2.8cm h:1.4cm
AVAL ed:"Externe Dokumentation"
IF (LEN ed)
  FILL color:$cafe69
  POLYGON 7 x1:1.2cm y1:-0.4cm x2:1.6cm y2:-0.4cm x3:1.6cm y3:-0.5cm
            x4:2.0cm y4:-0.3cm x5:1.6cm y5:-0.1cm x6:1.6cm y6:-0.2cm
            x7:1.2cm y7:-0.2cm
  HOTSPOT "Externe Dokumentation" x:1.2cm y:-0.5cm w:0.8cm h:0.4cm
          text:("Execute: " + copy (ed, 0, search (ed, "@", 0)))
ENDIF
```

If the attribute "external documentation" of type "program call" contains a value, an arrow will be displayed and additionally a hotspot is available with which you can call the quoted program with the quoted parameters. As Info text the word "Execute:" followed by the name of the program is displayed.
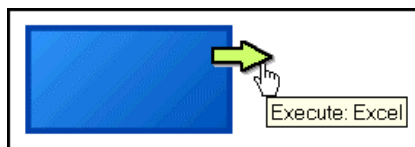
Figure 146: Example of a PROGRAMCALL hotspot

row and **col** enable access to distinct cells in attributes of type *Table* (RECORD).

TABLE

Table :
```
TABLE [ x:xpos ] [ y:ypos ]
      [ w:width ] [ h:height ]
      cols:n rows:m
      [ w1:width ] ... [ wn :width ]
      [ h1:height ] ... [ hm :height ] .
```

The **TABLE** element is an assigning element used for re-sizeable objects. A table consists of columns and lines which each have either a fixed or a variable width or height. Variable widths or heights change when the object's size is changed. The sizes calculated are assigned to runtime variables

which the elements below can access. Usually, a **TABLE** element is followed by a **STRETCH off**, to avoid double stretching of the elements.

A table defines a rectangular area divided into a fixed number of columns (**cols**) and rows (**rows**) The attributes for the single column widths of a table of *n* columns bear the names **w1** to **w*n***. Similarly, the attributes for the line heights for *m* lines are named **h1** to **h*m***. If you wish to specify a fixed column width or line height, such as **w1:0.2cm** you have to enter this directly. For variable sizes, an amount such as **w2:50%** or **w:0.5** is specified instead. This amount refers to the rest of the table's height or width after the fixed sizes have been subtracted.

From a **TABLE** element with *n* columns and *m* lines the following runtime variables are calculated/derived:

- **tabw1** to **tabw*n*** - column widths

- **tabh1** to **tabh*m*** - line heights

- **tabx0** to **tabx*n*** - x-positions of the columns

- **taby0** to **taby*m*** - y-positions of the columns

tabx*i* means left border of column *i*+1 (*i* = 0, ..., *n*-1) or right border of column *i* (*i* = 1, ..., *n*).

taby*i* means upper border of line *i*+1 (*i* = 0, ..., *n*-1) or lower border of line *i* (*i* = 1, ..., *n*).

<u>**Example:**</u>

The following GraphRep definition is given:

```
GRAPHREP sizing:asymmetrical
RECTANGLE x:0cm y:0cm w:3cm h:2cm
ATTR "Description" x:0.2cm y:0.2cm w:2.6cm h:1.6cm
```

A text in a rectangle is displayed. The distance of the text's margin to the rectangle's side is to be 0.2 cm. If the object size is changed, both the rectangle's and the text field's size will be changed. The geometric stretching here would also effect the width of the edges. To prevent this we need a table with three columns and three lines where the first and the third columns and lines each have a fixed width or height of 0.2 cm:

```
GRAPHREP sizing:asymmetrical
RECTANGLE x:0cm y:0cm w:3cm h:2cm
TABLE x:0cm y:0cm w:3cm h:3cm cols:3 rows:3
      w1:0.2cm w2:100% w3:0.2cm
      h1:0.2cm h2:100% h3:0.2cm
STRETCH off
ATTR "Description" x:(tabx1) y:(taby1) w:(tabw2) h:(tabh2)
```

> **Hint:** When defining resizable GraphReps, it is advised to use the parameter **smart-symbol-size** (see p. 134) wherever possible. This makes the definition of a symbol with both flexible and fixed parts much easier.

_uilang

The ADONIS user interface language is determined during login. At this time, a variable **_uilang** is set which can be used for defining the graphical representation. Thus, language-dependent attribute values can be displayed.

The value of _uilang is the ISO-639 code (see chap. 6., p. 541) of the language (two letters).

Example:

```
GRAPHREP
# ...
IF (_uilang = "en")
{
  ATTR "Comment" w:c h:c
}
ELSIF (_uilang = "de")
{
  ATTR "Kommentar" w:c h:c
}
```

_outdevtype

The output device for a graphical representation can be queried in the GraphRep. A predefined variable **_outdevtype** contains the information needed to create device-specific looks. It can have these values:

- **"drawingarea"**: The device is a drawing area inside a model window.

- **"printer"**: The device is a printer.

- **"icon"**: The device is an icon for the modelling bar, for a list box or for a menu.

- **"bitmap"**: The device is a pixel graphics file (bmp, png, jpg etc.).

- **"emf"**: The device is a Windows Enhanced MetaFile.

- **"svg"**: The device is an SVG file.

Example: Hiding a "button" at any output device but the drawing area

```
GRAPHREP
# ...
IF (_outdevtype = "drawingarea") {
  # draw a "button"
  RECTANGLE x:1cm y:-1cm w:0.3cm h:0.3cm
}
```

2.1.1.2   AttrRep

The class attribute "AttrRep" controls the ADONIS Notebook structure of a class or a relation. Each notebook consists of chapters which contain the attributes of a class or relation. In addition, a chapter's attributes may be arranged in group boxes.

The language describing the notebook's structure is based on the following syntax:

Notebook :	**NOTEBOOK** [**with-relations** | **move-relations:***intValue*] { *NBElement* | *SetAccess* | *Language* } .
***NBElement* :**	*Chapter* | *Group* | *Attribute* .
***Chapter* :**	**CHAPTER** *chapterName* [**color:***ColorSpec*] .
***Group* :**	**GROUP** *groupName* [**color:***ColorSpec*] { *Attribute* } **ENDGROUP** .

Attribute :	**ATTR** *AttrName* [**write-protected**] [**format:**strValue]
	[**dialog:**Dialog]
	[**lines:**intValue] [**font-family:**FontFamily]
	[**color:**ColorSpec]
	[**ctrltype:**ControlType]
	[**unchecked-value:**strValue] [**checked-value:**strValue]
	[**no-auto**] [**no-param**]
	[**push-button**] [**align:**Alignment] .

| *FontFamily* : | **decorative** │ **modern** │ **roman** │ **script** │ **swiss** │ **system** . |

| *Dialog* : | **time** │ **date** │ **datetime** │ **distribution** │ **actor** │ **subprocess** │ **re-**
source │**modelname** │ **instancename** │ **color** │ **person-calendar**
│**processstart-calendar** │ **transcond** │ **acfilter** │**wizard** . |

| *ControlType* : | **radio** │ **dropdown** │ **check** . |

| *SetAccess* : | **SET_ACCESS usergroup:**UserGroupSpec **mode:**AccessMode . |

| *UserGroupSpec* : | userGroupName │ **all** . |

| *AccessMode* : | **blocked** │ **protected** │ **full** . |

| *Alignment* : | **l** │ **c** │ **r** . |

| *Language* : | **LANG** langSpec . |

The notebook description is introduced by **NOTEBOOK**. The chapter "Relations" is only displayed in the notebook of the particular object, when **with-relations** is specified. The position at which the "Relations" chapter is to be inserted in the list of chapters, can be specified by the attribute **move-relations**. Here **0** indicates "right at the beginning", **1** following the first chapter and so on. By default the "Relations" chapter is added at the end.

A notebook is first structured by **CHAPTERS** with non-ambiguous names (*ChapterName*).

> **Hint:** Should no chapters be defined or should the notebook definition contain a Group Box or attribute elements prior to a first chapter definition, a chapter without a name will be automatically generated.

Each notebook chapter contains fields which show attribute values and allow the user - if necessary - to edit these. Within a chapter these attribute fields may be grouped together by the expression **GROUP**. Each group then is awarded a group title (*GroupName*).

> **ATTENTION:** Each group definition must be finished by the expression **ENDGROUP**.

The expression **ATTR**, followed by the attribute's name (*AttrName*) causes the particular attribute to become part of the notebook.

> **Hint:** The attribute specified under *AttrName* must have been defined within the library.

The look of an attribute field (text field, enumeration etc.) depends on one hand on the attribute type and on the other hand on the modifiers (**lines**, **dialogue**, **ctrltype**) specified in the notebook definition. The following types of attributes exist in ADONIS:

- **Integer - INTEGER** (see p. 173)

- **Floating-point number - DOUBLE** (see p. 174)

- **Text - STRING** (see p. 174)

- **Distribution - DISTRIBUTION** (see p. 179)

- **Date - DATE** (see p. 180)

- **Date and time - DATETIME** (see p. 180)

- **Time - TIME** (see p. 175)

- **Enumeration - ENUMERATION** (see p. 175)

- **Enumerationlist - ENUMERATIONLIST** (see p. 176)

- **Longtext - LONGSTRING** (see p. 175)

- **Programcall - PROGRAMCALL** (see p. 176)

- **Inter-model references - INTERREF** (see p. 179)

- **Attribute profile reference - ATTRIBUTEPROFILEREFERENCE** (see p. 179)

- **Record - RECORD** (see p. 177)

- **Expression - EXPRESSION** (see p. 178)

The **write-protected** modifier can be used for all types of attributes. When specified, the attribute values in a notebook displayed may not be changed.

The modifier **color** can be used in attribute groups and chapters to set the color of the notebook element explicitly (for definition of colors see chapter GRAPHREP-syntax (see p. 134)).

Language-dependent notebooks:

Depending on the ADONIS configuration, a user interface language can be chosen at login. This, of course, influences the notebook structure as well - e.g. for language-dependent chapters or the order of the attributes.

The keyword **LANG** together with a language code (see chap. 6., p. 541) makes all following **CHAPTER**, **GROUP**, and **ATTR** elements invisible for all other languages. **LANG all** cancels this and makes the following elements globally visible.

Example:

```
NOTEBOOK
   LANG "de"
      ATTR "Deutsche Bezeichnung"
   LANG "en"
      ATTR "English Identifier"
   LANG "all"
# The following attributes are visible in all languages
```

Attribute of type INTEGER (integer)

An INTEGER Attribute is defined as an integer from between -1,999,999.999 and +1,999,999.999. The ADONIS integer is limited to 10 decimal places plus (optionally) algebraic sign. Standard value is 0, or other value defined in application library.

> **Hint:** The INTEGER attribute domain can be limited by the definition of application library. When there are inputs which do not fulfil the defined requirements, an error massage will be shown.

In case of integer attributes, the attribute value is always displayed in its own text box (see fig. 147).

Figure 147: Input box for integer attribute

Additionally, there is a possibility to define an integer attribute as an option box (see fig. 148), by a modifier **ctrltype: check**.
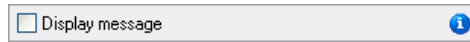
Figure 148: Option box for integer attribute

The deactivated option ☐ of the integer attributes has the value null, and the activated option ☑ has the value one.

Attribute of type DOUBLE (floating-point number)

Attribute of type DOUBLE comes from between +/- 999,999.999,999,999 for whole numbers and +/- 999,999.999,999,999 for numbers with (maximum) 6 decimal places (in a sum 15 places plus, if necessary, algebraic sign). Default value is 0,000000 or any other defined in the application library.

> **Hint:** The range of values of a DOUBLE Attribute and the number of decimal places can be defined during the definition of the application library. When there are inputs which do not fulfil the defined requirements, error massage will be shown.

A floating-point number attribute value is shown by default:

Figure 149: Input box for floating-point number attribute

Attributes of type STRING

Attributes of type STRING can contain texts with a maximum 3700 characters. The standard value is "" (empty) or any other, defined in the application library.

> **Hint:** Entity names can contain texts with maximum 250 characters.

For text attributes in the library definition you specify whether the field for the attribute will be displayed as one line (see fig. 150) or several lines (see fig. 151).
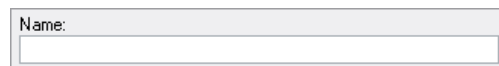
Figure 150: Input field for single-line text attribute
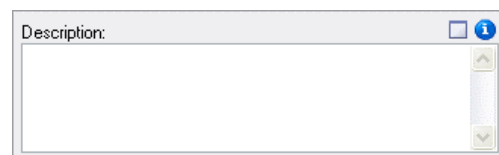
Figure 151: Input field for multi-line text attribute

For text fields that are comprised of several lines, the number of lines can be specified by using the modifier **lines** in the notebook definition. The default value is 5.

Independent of a text field's size you can press - in a displayed notebook - the "Maximise" button to open a window with a larger text field so that it is easier for you to enter or view the text.

For attributes for which a specific dialogue has been specified in the notebook definition, the "Dialogue" button which supports the input will be shown.

The following types of dialogues are available:

actor   dialogue for the assignment of performers,

distribution   dialogue for the assignment of a distribution function,

subprocess   dialogue for the selection of a subprocess,

resource   dialogue for the assignment of resources,

model name   dialogue which allows you to select the name of a model which will be entered in the attribute field,

instance name   dialogue which allows you to select an object from the models currently opened - the name of the object will be entered in the attribute field,

person's calendar   dialogue for defining the personal calendar for a performer,

process start calendar   dialogue for defining a process start calendar.

Attribute of type LONGSTRING (Long text)

The rules concerning long text attributes are the same as for attributes of type STRING (see p. 174), but the maximum length of the texts here can be 32.000 characters.

Attributes of type TIME

The value of time attributes is always shown in a single-line text field. The format is always `YY:DDD:HH:MM:SS (Years:Days:Hours:Minutes:Seconds)` (see fig. 152). In addition, it is possible to call a supporting dialogue via the "Dialog" icon displayed in the notebook. This interface allows you to enter time values easily and accurately. This dialogue can also be called for write-protected time attributes - however in this case it only provides a clearer representation of the value.
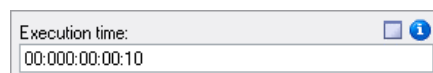
Figure 152: Input field for a time attribute

Attributes of type ENUMERATION

ENUMERATION attribute values can only be selected from its pre-defined list. This list is defined in an application library, but can always be widened.

The following types of representation are available for attributes of type enumeration:

● Complete list of all values.

You select a value by clicking on it, the current value is then marked by an activated radio button (see fig. 153).
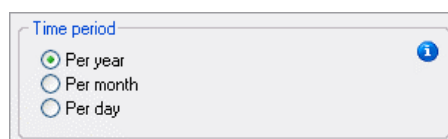
Figure 153: Input field for an enumeration attribute as a list

- Display of current value.

  You select the particular value from a dropdown list:
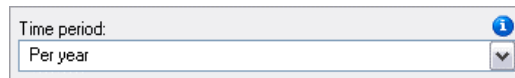
Figure 154: Input field for an enumeration attribute as a list

- Display as option field.

  This selection is possible through making an attribute active, or inactive:
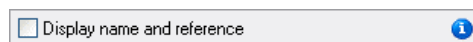
Figure 155: Input field for a two-valued enumeration attribute

The type of representation can be defined by the modifier **ctrltype**.

The complete representation ("**ctrltype:radio**") is normally chosen when there is a smaller range of values. The representation of only the current value ("**ctrltype:dropdown**") is normally selected when there is a larger number of values to select from.

> **Hint:** By default, all values are listed for up to four possible values. Otherwise, only the current value is represented.

For the representation of two attribute values it is possible to define one option field ("**ctrltype:check**"). However, in such a case it must be quoted additionally which of the two values stand for the activated option ☑ (e.g. '**checked value:"yes"**') and which one for the deactivated option ☐ (e.g. '**unchecked value:"no"**').

Attributes of type ENUMERATIONLIST

ENUMERATIONLIST attributes can only be selected from its pre-defined list. In contrats to the attributes of type ENUMERATION, the Enumerationlists (see p. 175) can contain one, more than one, or no values. The standard value is defined in the application library, and can always be changed.

The enumeration list attributes are shown in a list consisting of several rows. Values are added via a dialog which can be called using the "Add" button ➕.
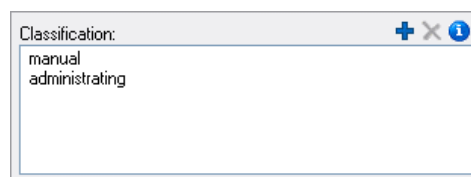
Figure 156: Input field for an enumeration list attribute

Entries selected in the list can be removed using the "Delete" button ✖ in the notebook.

Attributes of type PROGRAMCALL

The PROGRAMCALL attribute is defined by a fixed sentence of input (Items). These inputs are connected to AdoScripts, which can be called in a user surface area. Each attribute value consists of (maximum) one defined item and one optional parameter. The Syntax looks like the following:

__ProgramCallDomain :__   `{ ItemDefinition } .`

***ItemDefinition* :**	`ITEM` *itemText* [*ParameterDefinition*] { *FDlgFilter* } *AdoScript* .
***ParameterDefinition* :**	`param:` *paramText* [`:`*defaultTextValue*] .
***FDlgFilter* :**	`fdlg-filter`*<i>*`:` *filterText* `fdlg-type`*<i>*`:` *filterDescriptionText* .

***itemText, paramText, defaultTextValue, filterText, filterDescriptionText* :** *strValue* .

Program call attributes are represented in notebooks by two fields grouped together (see fig. 157). One field defines the program to be called, the other the program parameters.
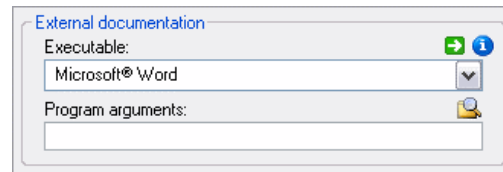
Figure 157: Input field for a program call attribute

PROGRAMCALL in the Notebook:

A program specified in the program call attribute can be called from the notebook using the "Execute" button . The available programs are selected from a dropdown list . Thus the selection is restricted to the programs listed here. Any text can be entered for the parameters or the dialogued "Search file" can be used to select a particular file.

This standard setup can be modified with several notebook definition commands in the class attribute "AttrRep" (see chap. 2.1.1.2, p. 171):

Through the quotation of the modifier **push-button** at only one available program instead of the selection list "executing program" a button for the program call will be displayed (see fig. 158).
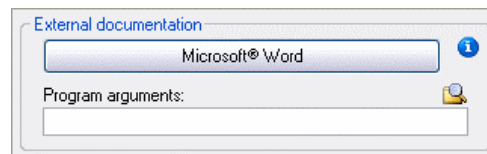
Figure 158: Input field for a program call attribute with Program call button

The quotation of the modifier **no-param** suppresses the display of the attribute field "parameter" (see fig. 159).
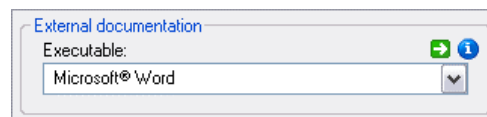
Figure 159: Input field for a program call attribute without giving parameters

The quotation of the modifier **no-auto** suppresses the value "<automatically>" in the field "executing program" . By selection of the value "<automatically>" the appropriate application will be started automatically (with respect to dependency of entered parameters).

Attribute of the type RECORD

Tables (RECORDs) will be represented in the notebooks according to the definition of the existing table classes.

     ATTENTION: Table attributes may only be used in objects and not in relations!

The following types of attributes can be contained in tables:

- **Integer - INTEGER** (see p. 173)

- **Floating-point number - DOUBLE** (see p. 174)

- **Text - STRING** (see p. 174)

- **Distribution - DISTRIBUTION** (see p. 179)

- **Date - DATE** (see p. 180)

- **Date and time - DATETIME** (see p. 180)

- **Time - TIME** (see p. 175)

- **Enumeration - ENUMERATION** (see p. 175)

- **Enumerationlist - ENUMERATIONLIST** (see p. 176)

- **Long text - LONGSTRING** (see p. 175)

- **Program call - PROGRAMCALL** (see p. 176)

- **Predominant model reference - INTERREF** (see p. 179)

- **Expression - EXPRESSION** (see p. 178)

The quotation of the modifier **width:** and a floating-point number enables the determination of the column width of the attribute within a table. The quoted values of the column width will be relatively calculated on the available space in the table through which all columns in the table will be displayed. When extending and reducing the table, the column widths will be adapted accordingly.

    Hint: The width modifiers must define <u>**all**</u> contained attributes (**ATTR**), as otherwise the display of the optimal column width is executed and therefore the width of all columns can be larger than the place available in the table.

Attribute of the type EXPRESSION

EXPRESSION attributes can be used to save formulas, which can then be used to calculate new values. The formulas can be defined while characterising an application library ("**fixed**"), or can be changed according to the current needs in models at any time ("**non-fixed**").

Each expression attribute has its own *type of result*, defined in an application library. Possible types are STRING (see p. 174), INTEGER (see p. 173), DOUBLE (see p. 174) or TIME (see p. 175).

Expressions (EXPRESSIONs) will be displayed in notebooks according to their definition in a text field with one or more lines (see fig. 160).
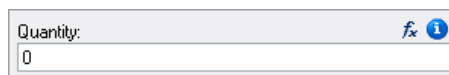
Figure 160: Input field for an expression attribute (example)

For text fields with more than one line, it is possible to determine the number of lines over the modifier **lines** in a notebook. The default number is 5.

The definition of an expression through the ADONIS user can be done through clicking on the Smart-Icon *fx* .

Attributes of type INTERREF (Reference)

Inter-model references are displayed in notebooks in a special list (see fig. 161).
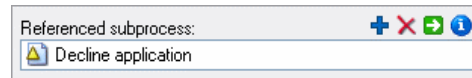
Figure 161: Input field for an inter-model reference

A new reference may be created ("Create" button ✚), existing references deleted ("Delete" button ✖) and existing references can be followed ("Follow"-Button ➡). If you wish to create references, a dialog is called when you click on the "Add" button, which will support the selection of the model or object(s) referenced.

Attribute of the type ATTRPROFREF (Attribute profile reference)

Attribute profile references will be represented according to the definition in the notebooks of the referenced attribute profiles (see chap. 8., p. 77).

The following types of attribute can be defined in attribute profiles:

- **Integer - INTEGER** (see p. 173)
- **Floating-point number - DOUBLE** (see p. 174)
- **Text - STRING** (see p. 174)
- **Distribution - DISTRIBUTION** (see p. 179)
- **Date - DATE** (see p. 180)
- **Date and time - DATETIME** (see p. 180)
- **Time - TIME** (see p. 175)
- **Enumeration - ENUMERATION** (see p. 175)
- **Enumerationlist - ENUMERATIONLIST** (see p. 176)
- **Long text - LONGSTRING** (see p. 175)
- **Program call - PROGRAMCALL** (see p. 176)
- **Predominant model reference - INTERREF** (see p. 179)
- **Table - RECORD** (see p. 177)
- **Expression - EXPRESSION** (see p. 178)

Attributes of type DISTRIBUTION

The values inserted in distribution attributes are shown in a text field (see fig. 162). In addition, it is possible to call a supporting window via the "Dialogue" icon ▢, where the modifier **dialog:distribution** is shown.
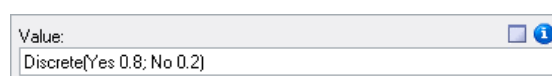
Figure 162: Input field for a distribution attribute

The modifier **lines** allow you to specify the number of lines.

Attribute of Type DATE

The assignment of date attributes will always be made in a row text field, in the following form

```
YYYY:MM:DD (year:month:day)
```

(see fig. 163) Additionally, the user can (by using a "Dialogue" icon from a notebook) ☐ call the supporting window where the modifier **dialogue:date** will be set. It refers also to read-only attributes, but in such a case the dialogue only helps to improve clarity of a presentation.
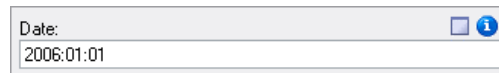
Figure 163: Input field for a data attribute

Attribute of the type DATETIME

The assignment of date and time attributes will always be made in a row text field, in the following form

```
YYYY:MM:DD hh:mm:ss (Year:month:day hour:minute:second)
```

(see fig. 164) Additionally, the user can (by using a "Dialogue" icon from a notebook) ☐ call the supporting window where the modifier **dialogue:date** will be set. It refers also to read-only attributes, but in such a case the dialogue helps only to improve clarity of a presentation.

Figure 164: Input field for a date and time attribute

2.1.1.3   Modellzeiger

The class attribute "Modellzeiger" controls the showing/hiding of submodels and the model navigation via "<Ctrl>+double click" on objects.

On showing a submodel, a referenced model is shown within the active model. The attribute, whose reference is used for that action, is defined in the Modellzeiger library attribute.

Through double clicking on the objects while simultaneously holding the <Ctrl> key or <Strg> key, the model referenced in the object will be opened and, in the case of having the model pointer defined, will be brought to the foreground. In the class attribute "Modellzeiger" the name of that attribute of type "inter-model reference" (INTERREF) is entered, which may contain a model pointer to another model.
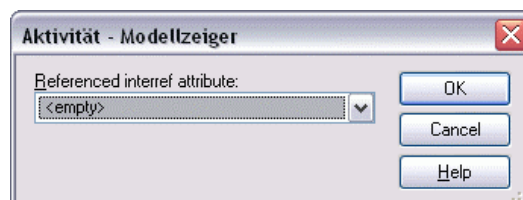
Figure 165: Input support for model pointers

2.1.1.4   Klassenkardinalität

In this attribute you can define how many relations may lead to or from the particular class. If no cardinality is specified it is assumed that the number of relations is not restricted.

In addition you can define how many objects of one class can have as a maximum or must have as a minimum to exist in one model.

The language for describing class cardinalities is based on the following syntax:

Cardinalities : `CARDINALITIES` [**max-objects:***NumValue*]
                 [**max-relations:***NumValue*]
                 [**max-outgoing:***NumValue*]
                 [**max-incoming:***NumValue*]
                 [**min-objects:***NumValue*]
                 [**min-relations:***NumValue*]
                 [**min-outgoing:***NumValue*]
                 [**min-incoming:***NumValue*]
                 { *RelationCardinalities* } .

RelationCardinalities : `RELATION` "*RelationName*"
                 [**max-outgoing:***NumValue*]
                 [**max-incoming:***NumValue*]
                 [**default-max-outgoing:***NumValue*]
                 [**default-max-incoming:***NumValue*]
                 [**min-outgoing:***NumValue*]
                 [**min-incoming:***NumValue*]
                 [**default-min-outgoing:***NumValue*]
                 [**default-min-incoming:***NumValue*]
                 { *ToClassCardinality* | *FromClassCardinality* } .

ToClassCardinality : `TO_CLASS` "*ClassName*"
                 [**max-outgoing:***NumValue*]
                 [**min-outgoing:***NumValue*] .

FromClassCardinality : `FROM_CLASS` "*ClassName*"
                 [**max-incoming:***NumValue*]
                 [**min-incoming:***NumValue*] .

NumValue :   A number between 0 and 65535.

RelationName: A name of a relation.

ClassName: A name of a class.

The expression **max-objects** specifies the maximum number of objects of this class which may exist in a model.

The expression **min-objects** specifies how many objects of a class have to exist in a model.

The expression **max-relations** specifies the maximum number of relations which may be connected to an object of this class.

The expression **min-relations** specifies how many relations have to be connected to an object of this class.

The expression **max-outgoing** below **CARDINALITIES** specifies the maximum number of outgoing relations from an object of this class.

The expression **min-outgoing** below **CARDINALITIES** specifies how many relations have to come out from an object of this class.

The expression **max-incoming** below **CARDINALITIES** indicates the maximum number of incoming relations to an object of this class.

The expression **min-incoming** below **CARDINALITIES** indicates the minimum number of incoming relations to an object of this class.

These restrictions may be further restricted by the expression **RELATION** for as many relation classes as desired.

The expression **max-outgoing** within **RELATION** specifies the maximum number of relations which may start at an object of this class.

The expression **min-outgoing** within **RELATION** specifies the minimum number of relations which have to start at an object of this class.

The expression **max-incoming** within **RELATION** specifies the maximum number of relations which may end at an object of this class.

The expression **min-incoming** within **RELATION** specifies the minimum number of relations which have to end at an object of this class.

The expression **default-max-outgoing** specifies the maximum number of objects (**of the same** class) from which originating relations of the respective relation type may exist. A default value is specified which applies to all target classes which are not explicitly named by the expression **TO_CLASS**.

The expression **default-min-outgoing** specifies for how many objects (**of the same** class) originating relations of the respective relation type have to exist. A default value is specified which applies to all target classes which are not explicitly named by the expression **TO_CLASS**.

The expression **default-max-incoming** specifies the maximum number of objects (**of the same** class) from which relations of the respective relation type may lead to an object of the current class. A default value is specified which applies to all classes which are not explicitly named by the expression **FROM_CLASS**.

The expression **default-min-ingoing** specifies for how many objects (**of the same** class) originating relations of the respective relation type have to lead. A default value is specified which applies to all target classes which are not explicitly named by the expression **TO_CLASS**.

The expression **TO_CLASS** defines the maximum number of relations of the particular relation type which may lead from an object of the current class to objects of the same class. For the class specified the default value defined by **default-max-outgoing** will be overwritten.

The expression **FROM_CLASS** defines the maximum number of relations of the particular relation type which may lead from objects of this class to an object of the current class. For the class specified the default value defined by **default-max-incoming** will be overwritten.

> **Hint:** Increased focus means that the restrictions can only be intensified and not eased. For example, the maximum number of outgoing relations belonging to one relation type cannot be bigger as defined in the appropriate restriction.

2.1.1.5   Zulässige Objekte

> **Hint:** The class attribute "Zulässige Objekte" will only be displayed at specifically defined swim lane classes. Contact your ADONIS administrator for more information.

The class attribute "Zulässige Objekte" in the swim lane class defines, whether instances of certain object classes may be placed on this swimlane or not.

If you try to place (create object, move, copy or paste) a not allowed object (i.e. an object of a class which is not allowed to be modelled in a determined swim lane class) on a swim lane, the mouse pointer changes its form (prohibition sign) and the action will not be executed.

The language for the definition of the allowed objects is based on the following syntax:

___AllowedObjects :___	`ALLOWED from:`*`Basis`*
	`{ `*`InclOrExclClass`*` } .`
___Basis___ :	`all` | `none` .
___InclOrExclClass___ :	`INCL "`*`className`*`"` | `EXCL "`*`className`*`"` .

The list of the permitted classes is basically determined with the attribute **from:** and modified afterwards.

- **from:all** means, that all instanced classes are permissible.

- **from:none** means, that no instanced classes are permissible (empty quantity).

Additionally it is possible to execute the following quantity operations:

- **INCL** adds the quoted classes (*className*) to the current quantity (in case the basis is **from:none**).

- **EXCL** removes the quoted class (*className*) from the current quantity(in case the basis is **from:all**).

2.1.1.6   Conversion

> **Hint:** The class attribute "Conversion" will only be displayed if it was defined for the class. Please contact, if necessary ADONIS Administrator.

The class attribute "Conversion" controls the conversion of an object of a certain class into an object (target object) of another class (target class).

Through opening the object context menu and the selection of the menu item "transform", the names of the classes will be displayed as a sub menu item, in which the current object will be transformed to. Through the selection of a class name the selected object will be turned into an object of the selected class: The new object is created, some attributes are handed over and the old object is deleted.

The language for the description of the transformation bases on the following syntax:

___Conversion :___	`{ `*`ClassConversion`*` } .`
___ClassConversion___ :	`CLASS `*`className`*` { `*`AttrConversion`*` } .`
___AttrConversion___ :	`ATTR `*`attrName`*` [ from:`*`attrName`*` ] .`

In the expression **CLASS**, followed by the class name (*className*), the aim class, in which the object should be transformed to, should be displayed.

In the expression **ATTR**, followed by the attribute name (*attrName*), the name of the target attribute, in which an attribute value of an original object should be copied to, will be quoted. In the expression **from:** followed from the attribute name (*attrName*), the name of the original attribute, from which the attribute value should be copied, will be quoted.

> **Hint:** If the name of the original attribute is the same as the name of one of the aim attributes, the expression **from:** and the quotation of the original attribute do not have to be given.

2.1.2 Edit attribute scopes

The possible attribute values (value ranges) for attributes of the types ENUMERATION, ENUMERATIONLIST and PROGRAMCALL are specified by the definition in the application library.

The value range of attributes of these types can be extended later.

ATTENTION: Attributes defined in the ADONIS meta-model (such as the validity of a variable) can **not** be extended.

ATTENTION: An attribute's value range can only be extended. Restrictions are **not** possible since these could lead to inconsistencies in existing models.

As soon as the library has been loaded, the window "<Library Name> - Extend Value Range" (see fig. 166) will appear.
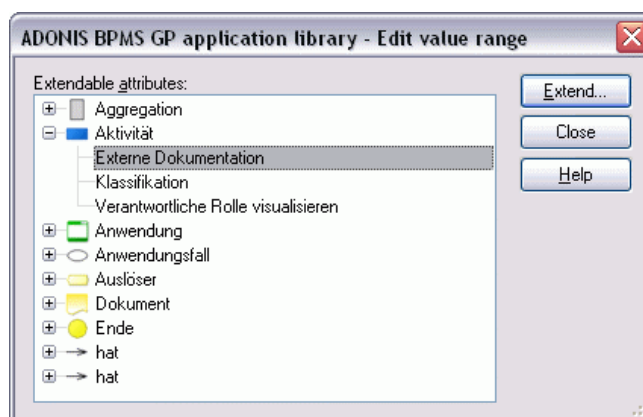
Figure 166: Extend attribute scopes

Select the attribute of which value area you want to enlarge and then click on the button "Extend" to:

● enlarge values of enumeration or enumeration list attributes (see chap. 2.1.2.1, p. 184) or

● to enlarge values of the program call attributes (see chap. 2.1.2.2, p. 185).

2.1.2.1 Extend value areas of enumeration (list)

In the window "<*library name*> - extend value area - <*attribute name*>" (see fig. 167)the already defined values of the enumeration (list) attributes will be displayed.
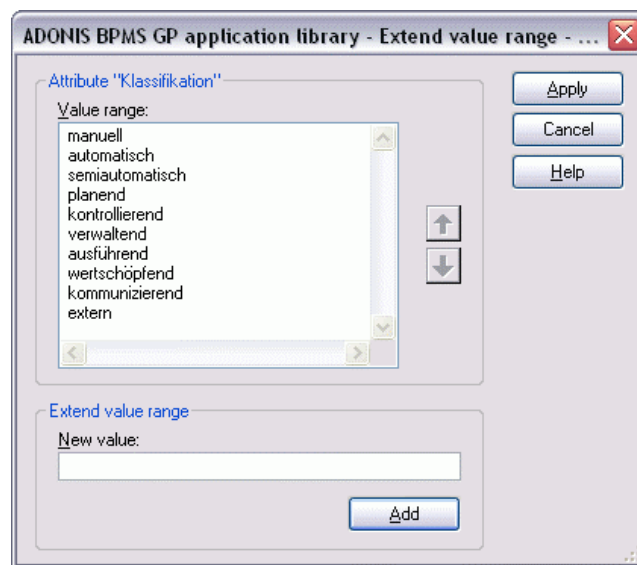
Figure 167: Extend value area

Enter the new attribute value "New value" and click on the button "Add". The new attribute value will be inserted in the list "value area" on the last position.

Additionally to the extension of the value area you can with the window "*<library name>* - enlarge the value area - <attribute name>" (see fig. 167) and also change the **order of the single attribute values**.

To do this select the moving attribute value and click on the button ⬆, to move the attribute value in a row up or on the button ⬇, to move the attribute value one row down.

Through clicking on the button "assign" the new value area will be saved after the saving query in the application library.

> **Hint:** An extended value area of an attribute can **not** be restricted, the order of the attribute value is changeable

2.1.2.2   Extend value area of the program call attributes

In the window"*<library name>* - extend value areas - *<Attribute names>*" (see fig. 168)will be displayed in the already defined values of the program attributes.
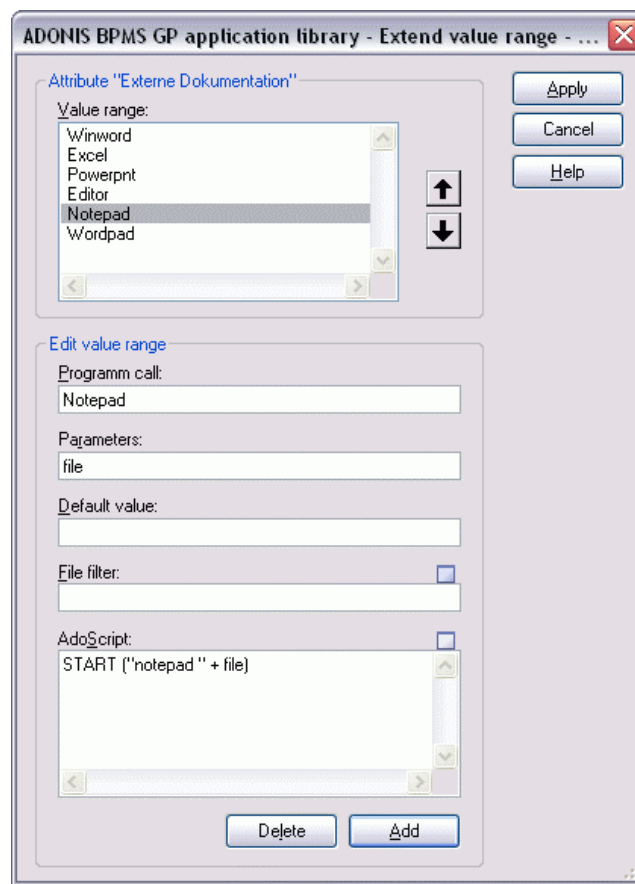
Figure 168: To edit a value area

Enlarge the value area, through inserting a new value in the field of the group "extend value area" as follows:

- In **the field "program call"** you enter the name which is displayed in the ADONIS-Notebook in the field "Executing program".

- In the **field "Parameter"** you need to define an AdoScript variable which will be filled automatically with the value in the ADONIS-Notebook in the field "Parameter" .

- In **the field "Standard value"** insert the value which is standard in the ADONIS-Notebook in the field "Parameter" and should be displayed.

- In **the field "file filter"** you define the filter which will restrict the input in the file dialogue window after clicking the file dialogue icon () to display the relative files.
  Note: For definition of the file filter (see p. 187)click on the dialog icon .

- The parameter in the **field "AdoScript"** defined AdoScript will be executed when you click on the ADONIS-Notebook on the executing-Icon () .

Click after the insertion on the **Button "Add"**, to enlarge the value area with the new insertion.

Through selecting an added value and clicking on the **Button "Delete"** you can delete the value from the value area.

> **Hint:** An added value can only be deleted if the enlargement of the value area is not saved by clicking the button "add" in the application library.

Additionally to the enlargement of the value area you can change in the window "*<library name>* - perform value area- <Attribute name>" (see fig. 168)also the **order of the single attribute values**.

In this case select the moving attribute value and click on the button 🔼 to move the attribute value one row above or on the button 🔽 to move the attribute value one row under.

Through clicking on the **button "Add"** the change will be saved after a saving query in the application library .

> **Hint:** An extended value area of an attribute can**not** be restricted if the order of the attribute values is changeable.

File filter

In the window "Program call - file filter" (see fig. 169) the already defined file filters will be listed.
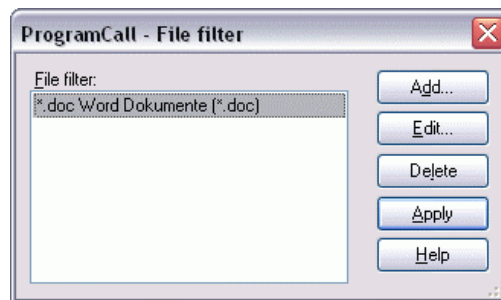
Figure 169

Through clicking the button:

- "**Add**" you can define a new file filter (see p. 187),
- "**Edit**" you can change (see p. 187) an already changed file filter,
- "**Delete**" you can remove a selected file filter from a list.

Click on the **Button "Add"** to overtake the changes.

Add file filter

In the window "Program call - Add file filter" (see fig. 170) you can define a new file filter.
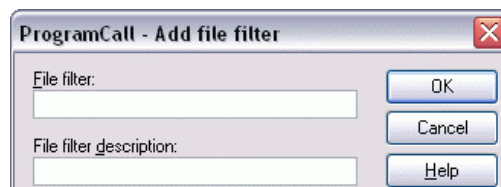
Figure 170: Add file filter

Quote in the field "file filter" the filter (e.g. "`*.htm*`") and in the field "file filter description an explanation of the file filter.

Edit file filter

In the window "Program filter - edit file filter" (see fig. 171) you change a specific file filter.
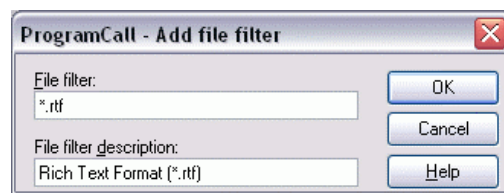
Figure 171: Edit file filter

Change in the field "file filter" the filter (e.g. "`*.txt`") or in the field "file filter description" the description to the file filter.

2.1.3   Edit library attributes

All libraries (ADONIS BPMS BP Library and ADONIS BPMS WE Library) do have library attributes. These library attributes serve two purposes: They describe the library and they are needed for layout, analysis, simulation, evaluation and more.

The library attributes are filled with values in the ADONIS Notebook. The notebook is structured as described below:

Chapter	Attribute	
Beschreibung	Schlagworte (see chap. 2.1.3.1, p. 190)	
	Beschreibung (see chap. 2.1.3.2, p. 190)	
	Kommentar (see chap. 2.1.3.3, p. 191)	
	Service (see chap. 2.1.3.4, p. 191)	*
	Autor (see chap. 2.1.3.5, p. 191)	
	Angelegt am (see chap. 2.1.3.6, p. 191)	
	Letzter Bearbeiter (see chap. 2.1.3.7, p. 191)	
	Letzte Änderung am (see chap. 2.1.3.8, p. 191)	
Erweiterungen	Modi (see chap. 2.1.3.9, p. 191)	
	Versionierungsformat (see chap. 2.1.3.10, p. 195)	*
	Externe Anbindung (see chap. 2.1.3.11, p. 197)	
Modellierung	Voreinstellungen (see chap. 2.1.3.12, p. 198)	
	Seitenlayouts (see chap. 2.1.3.13, p. 203)	
	Connector marks - Numerierung (see chap. 2.1.3.14, p. 207)	
	Connector marks - Grafische Darstellung (see chap. 2.1.3.14, p. 207)	
	Anordnungsfunktion (see chap. 2.1.3.15, p. 208)	*
Analyse	Beziehungsauswertungen (see chap. 2.1.3.16, p. 214)	
Simulation	Simulation definition - Simtext (see chap. 2.1.3.17, p. 217)	*
	Simulation definition - Simmapping (see chap. 2.1.3.18, p. 218)	*
	Simulation definition - Simergebnis-Mapping (see chap. 2.1.3.19, p. 220)	
	Simulation definition - Variablenprüfung (see chap. 2.1.3.20, p. 221)	*

	Agenten-Definition (see chap. 2.1.3.21, p. 222)	*
	Enterprise time - Tage pro Jahr (see chap. 2.1.3.22, p. 242)	*
	Enterprise time - Stunden pro Tag (see chap. 2.1.3.23, p. 242)	*
Evaluation	Activity based costing - CCC-Mapping (see chap. 2.1.3.24, p. 242)	*
	Activity based costing - CCC-Grundeinstellung (see chap. 2.1.3.25, p. 243)	*
	Dynamische Evaluationsmodule (see chap. 2.1.3.26, p. 243)	*
Dokumenta-tion	Dokumentations-Konfiguration (see chap. 2.1.3.27, p. 246)	*

Table 2: Notebook structure for library attributes

> **Hint:** The attributes marked by an asterisk ("*") can only be edited in the business process library's notebook. The attributes "Service", "User defined", "Library icons" and "Agent definition" refer to the application library. All other attributes marked are only used in the modelling of business processes.

> **ATTENTION:** An application library, which attributes are edited, is locked for other users. It means, that other ADONIS administrators cannot change its attributes (parallel) from a dibeing fferent place.

After you have selected the library to be edited in the window "Library management - library configuration" (see fig. 114) and clicked on the button "Library attributes" the notebook "<Library Name> - Library attributes" (see fig. 172) will appear.

Figure 172: Library attributes

After completing your changes, close the ADONIS library notebook by clicking on the button "Assign". Any changed attributes will be stored in the ADONIS database after a confirmation query.

> **ATTENTION:** We recommend that you check (see chap. 2.1.5, p. 277) the library attributes after editing. This allows you to ensure that the library attributes are correct and prevent possible runtime errors during the work with the ADONIS Business Process Management Toolkit.

> **Hint:** We also recommend that you talk to your ADONIS consultant prior to making any changes of library attributes.

2.1.3.1 Schlagworte (Beschreibung)

Enter keywords or a short description of the library here. For documentation purposes.

2.1.3.2 Beschreibung (Beschreibung)

Enter a description of the library here. For documentation purposes.

2.1.3.3 Kommentar (Beschreibung)

Enter a comment on the library here. For documentation purposes.

2.1.3.4 Service (Beschreibung)

An attribute where you can name a person (e.g. the ADONIS administrator), or give an address of the institution to contact in case of help needed (by the user).

The text entered here will be displayed in the Option "Service" from the "Help" menu of the Business Process Management Toolkit (see chap. 4., p. 18).

> **Hint:** The attribute "Service" is only defined in the business process library. It refers, however, to the application library to which this BP library is assigned.

2.1.3.5 Autor (Beschreibung)

A system attribute which is automatically filled with the name of the user who performed the library import into the database.

2.1.3.6 Angelegt am (Beschreibung)

A system attribute which is automatically filled with date and time of the library import into the database.

2.1.3.7 Letzter Bearbeiter (Beschreibung)

A system attribute which is automatically filled with the name of the last user editing the library attributes.

2.1.3.8 Letzte Änderung am (Beschreibung)

A system attribute which is automatically filled with the date and the time of the last changes to the library.

2.1.3.9 Modi (Erweiterungen)

In the library attribute "Modi" model types and (view) modes for model types can be defined.

A **model type** specifies a subset of all instantiable classes and relations. Every model is of a certain model type which cannot be changed once the model has been created.

Model type groups should be defined, in case an application library contains many model types. Thanks to this, relative model types can be put together and clarity will be improved.

A **mode** is a further restriction to a model type. It defines a subset of all the instantiable modelling classes, allowing the user to have different views on one model. In contrast to the model type a model's mode can be changed at any time so that a mode can serve to temporarily make certain modelling classes invisible.

The language for defining model types and (view) modes is based on the following syntax:

__ModelTypes :__   [*GeneralSettings*] [*Method*] { *ModelTypeDef* } .

GeneralSettings **:**	**GENERAL** [**order-of-classes:***OrderOfClasses*] **.**
OrderOfClasses **:**	**default** │ **custom .**
Method **:**	**METHOD** [**graphrep:***attrName*] **{** {*ModelTypeGroupDef*} **} .**
ModelTypeGroupDef **:**	**GROUP** *groupName* **{** { *ModelType* } **} .**
ModelType **:**	**MODELTYPE** *modelTypeName* **.**
ModelTypeDef **:**	**MODELTYPE** *modelTypeName*
	[**plural:***modelTypePluralName*]
	[**pos:***index*] [**from:***MTSource*]
	[**bitmap:***fileName*]
	[**auto-connect:***AutoConnectCriterion*]
	[**modeling-modi-accessible:***boolVal*]
	[**not-simulateable**] [**abstract**]
	[**default-access:***AccessMode*] [*BGBitmap*]
	[**attrrep:***attrName*] [**graphrep:***attrName*]
	{ *Access* } { *MTClasses* } { *ModeDef* } **.**
MTSource **:**	**all** │ **none** │ *modelTypeName* **.**
AutoConnectCriterion **:**	**pos** │ **part** │ **full .**
BGBitmap **:**	**bg-bitmap:***fileName*
	bg-bitmap-w:*width* **bg-bitmap-h:***height*
	[**bg-bitmap-repeat:***BGBitmapRepeatMode*] **.**
BgBitmapRepeatMode **:**	**none** │ **x** │ **y** │ **xy .**
Access **:**	**ACCESS usergroup:***userGroupName* **mode:***AccessMode* **.**
AccessMode **:**	**blocked** │ **protected** │ **full .**
MTClasses **:**	**INCL** *classOrRelnClassName* │ **EXCL** *classOrRelnClassName* │
	OR_ASSIGN *modelTypeName* │
	AND_ASSIGN *modelTypeName* **.**
ModeDef **:**	**MODE** *modeName*
	[**from:***ModeSource*]
	[**abstract**]
	[**no-modeling**]
	[**no-documentation**]
	{ *ModeClasses* } **.**
ModeSource **:**	**all** │ **none** │ *modelName* **.**
ModeClasses **:**	**INCL** *classOrRelnClassNameOfMT* │
	EXCL *classOrRelnClassNameOfMT* │
	OR_ASSIGN *modeName* │
	AND_ASSIGN *modeName* **.**

   Hint: GENERAL [**order-of-classes:***OrderOfClasses*] applies to the whole library, not only
      a part!

Model types:

A model type is marked by an unambiguous **name** *(modelTypeName)*. The plural form of the model
name *(modelTypePlural)* should also be specified.

With **pos**, the order of model types in the dialogue "Create new model" can be changed. By default,
the modeltypes are shown in the same order as in the model type definition. Using **pos** moves the re-

spective modeltype *upwards*. Consequently, if you want do move a modeltype down, you have to move the following modeltypes upwards.

If the attribute **abstract** is given, the model type can only be used as basis for creating other model types. It means that the user cannot create any other models of this type.

The attribute **auto-connect** sets how automatic relations between objects should be placed:

- At the definition of the **auto-connect:pos** a connector between two objects will be created if the position of the object is within the area of the other object.

- The definition of the **auto-connect:part** generates a connector if an object in the area of another object is touched.

- **auto-connect:full** only creates a connector if an object is fully in the area of another object.

With the attribute **attrrep** you are able to add your own defined model attributes.

> **Hint:** The attribute (*attributeName*), which contains a notebook definition for the model at-
> tribute is contained, must be defined in a class "__ModelTypeMetaData__". Please con-
> tact your ADONIS consultant, to properly extend the library definitions.

If a model type is defined as **not-simulateable**, models of this type will not be listed in model selection boxes, as they show models which can only be simulated.

Instantiable class:

The attribute **from** specifies a basis - which will then be further modified - for the instantiable classes of the model type.

- **from:all** includes all instantiable classes to be included.

- **from:none** specifies that we start with an empty set. The specification of a different model type al-
  ready defined determines that classes instantiable in that model will be the basis for the current
  model type

- Defining *another* (previously defined) modeltype sets the available classes there as the starting
  base for the new model type.

In addition, the following set operations can be carried out:

- **INCL** adds a class (or relation class) to the current set of classes.

- **EXCL** removes a class from the current set of classes.

- **OR_ASSIGN** unites the current set of classes with the class set of the model type specified and
  assigns this merged set to the current set of classes.

- **AND_ASSIGN** causes the intersection of the current set of classes and the class set of the model
  type specified and assigns this intersection to the current set of classes.

As a final result the user assigns the classes available for modelling within this model type.

Graphics:

Every model type can get a bitmap graphic (*fileName*) as a symbol via the attribute **bitmap**. This graphic will be used in all model and model type selection lists. The BMP file will be loaded from a file system over the file name *fileName* whereas the quotation of the path and the file name must be un-
der quotation marks .

> **Hint:** The graphic file must have the file format ***.bmp** and a size of 16*16 Pixel with the 16 colours of the standard palette. Afterwards it has to be copied into the ADONIS installation directory. If no bitmap file is specified, the symbol for Business Process Models (🔺) is used for the model types of the BP library and the symbol for Working Environment models (🖥) is used for the model types of the WE library.

"General Settings":

With help of the **GENERAL** Element you can influence the order of the classes and relations in the modelling bar (Modelling Component in the Business Process Management Toolkit) . The attribute **order-of-classes** can be used as follows:

- **order-of-classes:default** means, that the order of the library will be overtaken i.e. the order of the classes and relations in the modelling bars is caused by the order of the definition of the classes and the relations in the ABL file.

- **order-of-classes:custom** means, that the order of the classes and relations in the modelling bars through the order of the **INCL** Parameter (*ModeClasses*) is caused for the single modi.

Modi:

When the instantiable classes of a model type have been defined, the modes for the current model type may be defined. This happens in a similar way to the model type definition. Basis for quantity of links are only modes, which have already been defined for the same type of model.

With **modeling-modi-accessible**, the direct access to the menu entry "Mode" can be removed. This is useful, if the mode shall only be changed via AdoScript mechanisms.

As in case of model types, the user can set the starting point with **from:all** or **from:none**. The classes with **INCL**, **EXCL**, **OR_ASSIGN** and **AND_ASSIGN** will be assigned or removed.

If **no-modelling** is specified, the mode concerned cannot be applied for modelling and will thus not be shown in the option "Modes" in the Modelling Component.

By adding **no-documentation** a mode is specified which cannot be used for the generation of documentation.

Access for user groups:

Both model types and modes can by using **ACCESS usergroup:**userGroupName **mode:**accessMode be locked for certain user groups. In principle it is true, that a user group has a default access rights, provided no other (special) access rights has been defined for it. It is also a standard situation, that default access rights means (**full**) access rights, provided no other access rights have been defined as default rights in a model type or mode definition (element **default-access**). Read-only access rights to model types (**protected**) mean, that models of this type cannot be created, changed or deleted. Locked access rights to model types (**blocked**) mean, that models of that type are invisible and cannot be opened.

Model type groups / Method diagram:

For modelling methods, which are either complex or use model types which can be categorised into groups ("levels", "application scenarios"), it is suitable to use a method diagram. There, the model types are divided into as many sub groups as needed. On the user interface, a method diagram is shown as a method box in dialogues. The syntax is as follows:

MethodBox :	**METHOD** [**graphrep:***attrName*] **{** {*ModelTypeGroupDef*} **}** .
ModelTypeGroupDef :	**GROUP** *groupName* **{** { *ModelType* } **}** .
ModelType :	**MODELTYPE** *modelTypeName* .

The **METHOD** element launches creation of a method diagram. **GROUP** defines a group, **MODELTYPE** assigns a model type to the appropriate model group. The **graphrep** allows for the definition of an attribute, which sets the appearance of the group (in ADONIS GraphRep language). Within the GraphRep code a predefined variable **mtgroup** contains the current modeltype group name.

Beispiel:

```
METHOD graphRep:"Method GraphRep" {
    GROUP "Group 1" {
        MODELTYPE "Model type 1"
        MODELTYPE "Model type 2"
    }
    GROUP "Group 2" {
        MODELTYPE "Model type 3"
        MODELTYPE "Model type 4"
    }
}
```

2.1.3.10   Versionierungsformat (Erweiterungen)

ADONIS supports various types of model (and attribute profile) versioning:

● Model-based versioning

● Time-based versioning

The library attribute "Versionierungsformat" is defined in the BP library for the whole application library and is valid for all model types and attribute profiles.

Model-related versioning:

The version number can contain any user-defined text (20 characters maximum).

> **Hint:** In model-related versioning, only models have version numbers, while attribute profiles have none.

Time-related versioning:

On creation of a model/attribute profile under time-related versioning it receives a validity date. This validity ends with the beginning of the validity of the next version. The valid version is always the most recent one.

The language for the definition of the versioning is based on the following syntax:

VersioningFormat :	**VERSIONING** *start_date fields* .
start_date :	**START_DATE** [**day:***start_day*] [**month:***startMonth*] [**year:***start_year*] .
fields :	[*text_field*] [*day_field*] [*text_field*] [*month_field*] [*text_field*] [*year_field*] [*text_field*] .
text_field :	**TEXT_FIELD "***text***"** .
day_field :	**DAY_FIELD** [**default:***def*] [**minimum:***min*] [**maximum:***max*] .
month_field :	**MONTH_FIELD** [**default:***def*]

[**minimum:***min*] [**maximum:***max*] [**full**] .

year_field :   **YEAR_FIELD** [**default:***def*] [**minimum:***min*] [**maximum:***max*] .

The definition of versioning format begins with the key word **VERSIONING**. In the element **START_DATE** a start value (*Day*, *Month* and/or *Year*) can be quoted, which will be set when creating the new ADONIS-model.

The format for the display of the version date will consist of the following elements:

- **DAY_FIELD** for the day,

- **MONTH_FIELD** for the month,

- **YEAR_FIELD** for the year,

- **TEXT_FIELD** for a text for and between the date values.

    Hint: If in the value *text* **quotation marks** (") are contained they must be marked i.e. instead of ' " ' must ' \" ' be quoted.

Additionally to every date value (**DAY_FIELD**, **MONTH_FIELD** and/or **YEAR_FIELD**) a standard value (**default:**) as well as a minimum value (**min:**) or a maximum value (**max:**) must be quoted.

Through the quotation of the attribute **full** in the **MONTH_FIELD** the value of the month will be displayed as text.

    Hint: The parameter "default" is used only when the corresponding START_DATE does not exist.

Example:

```
VERSIONING
START_DATE month:1 year:2002
TEXT_FIELD "<"
MONTH_FIELD full
TEXT_FIELD " "
YEAR_FIELD default:2002 minimum:1950 maximum:2100
TEXT_FIELD ">"
```

Through the above definition the version date will be displayed in the form of <month yeas> where as the month will be fully written and the year is in the area between 1950 and 2100 (see fig. 173). When creating a new ADONIS-model the value "January 2001" will be suggested.
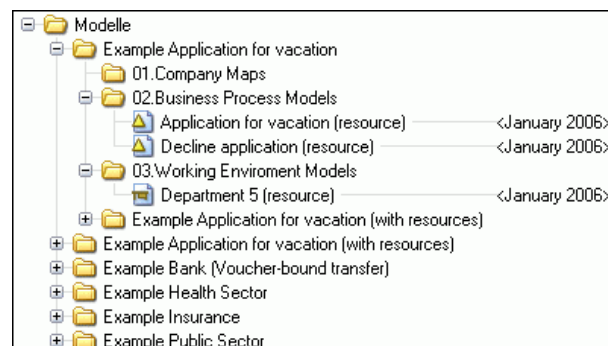
Figure 173: Time-related versioning (example)

2.1.3.11   Externe Anbindung (Erweiterungen)

With the library attribute "Externe Anbindung" functions in the Business Process Management Toolkit can be added, extended, modified or removed. For each component it is possible to define library-specific menus calling AdoScripts. These changes are possible either for all users or only for certain user groups.

To define new functions, *AdoScript* is used. This e.g. enables the design of:

- interfaces to objectiF and case 4/0 (manufacturer: microtool)

- new menu items and smart icons (actions)

- EventHandlers

   Hint: Please refer to the appropriate chapters of the documentation for details.

The language for the external coupling is based on the following syntax:

ExternalTools **:**	[*Case40Classes*] [*ObjectifClasses*] { *SetAccess* | *ToolsItem* | *EventHandler* } .
Case40Classes **:**	**CASE** { **CLASS** *className* } .
ObjectifClasses **:**	**OBJECTIF** { **CLASS** *className* } .
SetAccess **:**	**SET_ACCESS usergroup:***UserGroupSpec* **mode:***AccessMode* .
UserGroupSpec **:**	*strValue* | **all** .
AccessMode **:**	**blocked** | **protected** | **full** .

ToolsItem **:**	**ITEM** [*itemText* | **separator**]
	   { **name_***langCode* **:***itemText* } { *Component* }
	   [**sub-of:***strValue*] [**sub-of_***langCode* **:***strValue*]
	   [**pos1:***intValue*] [**pos2:***intValue*] [**pos3:***intValue*]
	   AdoScript .

Component **:**	*ComponentName* [**:***itemName*] { *ComponentName* **_***langCode* **:***itemName* } .
ComponentName **:**	[**acquisition**] [**modeling**] [**analysis**] [**simulation**] [**evaluation**] [**importexport**] .
EventHandler **:**	**ON_EVENT** *eventName* { *AdoScript* } .

   ATTENTION: An *AdoScript* belonging to the *AdoScript* will **never** be put into brackets. An *AdoScript* belonging to the **ON_EVENT** must **always** be put into brackets.

   Hint: *langCode* are two lower case letters defining a language according to ISO-639. As an example, german *itemText* can be specified with **name_de**.

Menu extensions:

With the **ITEM** constructs a new menu item can be defined. It will be inserted into the quoted component (or the quoted components) in the menu. The name of the menu item (*itemText*) will be quoted directly behind the keyword **ITEM**. Behind the component quotation the name of the insertion in the menu bar (*itemName*) will be quoted. If this quotation is left out the name of the "tool" will be set as name. It can be a new or an already existing insertion in the menu bar. If there is no insertion in the menu bar with the quoted name it will be inserted automatically. New menu items can not only be inserted into the menus which belong to the insertions of the menu bar but also to the sub menus. In this case the insertion has to be specified with **sub-of** which is connected with the submenu.

With **pos1** the position of the insertion can be determined in the menu bar. With no quotation it will be inserted prior to the "Extras menu". With **pos2** or **pos3** the position in the menu or the sub menu can be determined. With no quotation it will be added on the end.

An accelerator can be defined through a tilde ~ prior to the accelerator of the desired letter. If there is no tilde in the name the accelerator will be generated automatically.

If the so defined menu item is called in the Business Process Management Toolkit the **ITEM** belonging to *AdoScript* (see chap. 16., p. 609) will be executed.

As it is possible to select a language during login, do-it-yourself **ITEM**s have to be multilingual too, if this option is active. If only one language is available, a definition may look like:

```
ITEM "Test" modeling:"Extras" sub-of:"Submenu"
```

This adds a menu entry "Extras - Submenu - Test" to the menu bar of the Modelling Component. If more than one language is available, all further text items have to be specified with **name_<language>** and **sub-of_<language>**. If German (default) and English are available, a definition could look like:

```
ITEM "Käse" name_en:"Cheese"
modeling:"Brot" modeling_en:"Bread"
sub-of:"Ei" sub-of_en:"Egg"
```

In this case the German menu entry is "Brot - Ei - Käse" while the English one is "Bread - Egg - Cheese".

EventHandler:

EventHandler are Ado Scripts, which are executed when determined events happen. The available events are described in thechapter "Event Handler" (see chap. 16.19, p. 664) .

Access rights for user groups:

With **SET_ACCESS** the following **ITEM** s can be determined user group specific. As standard all insertions are accessible for all user groups i.e. all defined extensions are available for all users. If the following insertions should be invisible to a user group the user group must be set **mode:blocked**. To make insertions visible but not selectable set the mode to **mode:protected**. With **mode:full** you set the access right for a user group to full access.

If a user belongs to more user groups he will get the maximum access right for this user group.

In order to define the access rights for all user groups at the same time, **usergroup:all** should be used. It would be particularly useful, when access to one particular function should be limited and e.g. be allowed only for administrators.

case/4/0 and objectiF interface:

For every class listed (*classname*), the context menu (right mouse button) of the object of this class contains a menu entry "case/4/0" or "objectiF".

2.1.3.12   Voreinstellungen (Modellierung)

The library attribute **Voreinstellungen** is a place, where different patterns of behaviour for new models are stored:

● Snap-grid settings

- Move references on "Save as" and "Paste"

- Cardinality checks

- Switch to US time format

- SVG options

- Password restrictions

- Set change counter

- Gradient printing

The general language for Voreinstellungen is based on the following syntax:

DefaultSettings :	[*Grid*] [*MoveRefsOnSaveAs*] [*CheckCardinalities*] [*FormatSettings*] [*SVGOptions*] [*PasswordRule*] [*ChangeCounterDef*] [*GradientPrinting*] .
Grid :	**GRID** [**snap:***SwitchModifier*] [**visible:***SwitchModifier*] [**x:***measureVal*] [**y:***measureVal*] [**w:***measureVal*] [**h:***measureVal*] .
SwitchModifier :	**on** | **off** .
MoveRefsOnSaveAs :	**MOVE_REFS_ON_SAVEAS mode:***MoveRefsMode* .
MoveRefsMode :	**all** | **if-all-writeable** | **none** .
CheckCardinalities :	**CHECK_CARDINALITIES** [**before-save:***boolVal*] [**after-modeling-action:***boolVal*] .
boolVal :	**0** | **1** .
FormatSettings :	**FORMAT time:locale:"US"** .
SVGOptions :	**SVG_GENERATION** [**generateOnClick:***boolVal*] [**generateOnMouseDown:***boolVal*] [**generateOnMouseUp:***boolVal*] [**generateOnMouseOver:***boolVal*] [**generateOnMouseMove:***boolVal*] [**generateOnMouseOut:***boolVal*] [**clickRegionCursorHand:***boolVal*] [**generateConnectorClickRegion:***boolVal*] [*SVGClassesOptions*] .
SVGClassesOptions :	{ *ClassesDef* } .
ClassesDef :	**CLASS** *ClassName* [**generateOnClick:***boolVal*] [**generateOnMouseDown:***boolVal*] [**generateOnMouseUp:***boolVal*] [**generateOnMouseOver:***boolVal*] [**generateOnMouseMove:***boolVal*] [**generateOnMouseOut:***boolVal*] [**clickRegionCursorHand:***boolVal*] [**generateConnectorClickRegion:***boolVal*] .
PasswordRule :	**PASSWORD_RULE** *regExpr* **errMsg:***strVal* .
ChangeCounterDef :	**INCR_CHANGE_COUNTER_OF_SOURCE_MODELS enable:***boolVal* .
GradientPrinting :	**GRADIENT_PRINTING mode:***GradientPrMode* .
GradientPrMode :	**default** | **first-color** | **avg-color** .

regExpr is a string which is interpreted as regular expression (see chap. 4., p. 534).

> **Hint:** *ClassesDef* (from *SVGClassesOptions*) always has to be put in curly braces.

Snap grid settings:

The language for the snap grid settings is based on the following syntax:

<u>*Grid :*</u>	`GRID`
	`[ snap:`*SwitchModifier* `] [ visible:`*SwitchModifier* `]`
	`[ x:`*measureVal* `] [ y:`*measureVal* `]`
	`w:`*measureVal* `h:`*measureVal* `.`
SwitchModifier **:**	`on	off .`

The snap-grid can be activated (**snap:on**) or deactivated (**snap:off**) and faded in (**visible:on**) or faded out (**visible:off**). The horizontal (**w**) and vertical (**h**) distance between the grid-coordinates as well as the right (**x**) and the upper (**y**) offset of the snap-grid in cm can also be defined.

Example:

```
GRID snap:on visible:off w:0.5cm h:0.5cm x:0cm y:0cm
```

The snap grid is activated but faded out, the distance between the grid-coordinates both horizontally and vertically is 1/2 cm without offset.

Move references on "Save as" and "Paste":

The language for the references settings is based on the following syntax:

<u>*MoveRefsOnSaveAs :*</u>	`MOVE_REFS mode:`*MoveRefsMode* `.`
MoveRefsMode **:**	`all	from-writeable-models	none .`

When saving models in the ADONIS Business Process Management Toolkit via "Save as" option or when pasting model content it is possible to move ingoing references in the active model from the old instances to the new ones - even if some of the superordinated models have read-only access rights. As this can lead to confusion, it is possible to limit this behaviour: The re-assignment of references is then only possible if all the superordinated models have write access rights (**if-all-writeable**) or it is totally forbidden (**none**).

Cardinality checks:

The language for cardinality checks is based on the following syntax:

<u>*CheckCardinalities :*</u>	`CHECK_CARDINALITIES`
	`[ before-save:`*boolVal* `] [ after-modeling-action:`*boolVal* `] .`
boolVal **:**	`0	1 .`

boolVal is an integer value either 0 = false, or 1 = true. In the basic settings both checks are deactivated (0). If the **before-save** check is active, flawed models are not saved.

<u>**Switch to US time format:**</u>

The language for the time format switch is based on the following syntax:

FormatSettings :   `FORMAT time:locale:"US" .`

As a default, clock times in ADONIS are displayed in the European 24-hour format. With this command, the time format is switched to the US AM/PM format.

> **Hint:** At the moment, no other time formats are available. Therefore, every other specification means "European 24-hour format".

<u>**SVG options:**</u>

The language for the SVG options is based on the following syntax:

SVGOptions: `SVG_GENERATION`
`[ generateOnClick:`*boolVal* `]`
`[ generateOnMouseDown:`*boolVal* `]`
`[ generateOnMouseUp:`*boolVal* `]`
`[ generateOnMouseOver:`*boolVal* `]`
`[ generateOnMouseMove:`*boolVal* `]`
`[ generateOnMouseOut:`*boolVal* `]`
`[ clickRegionCursorHand:`*boolVal* `]`
`[ generateConnectorClickRegion:`*boolVal* `]`
`[ `*SVGClassesOptions* `] .`

SVGClassesOptions : { *ClassesDef* } .

ClassesDef :   `CLASS `*ClassName*
`[ generateOnClick:`*boolVal* `]`
`[ generateOnMouseDown:`*boolVal* `]`
`[ generateOnMouseUp:`*boolVal* `]`
`[ generateOnMouseOver:`*boolVal* `]`
`[ generateOnMouseMove:`*boolVal* `]`
`[ generateOnMouseOut:`*boolVal* `]`
`[ clickRegionCursorHand:`*boolVal* `]`
`[ generateConnectorClickRegion:`*boolVal* `] .`

> **Hint:** *ClassesDef* (from *SVGClassesOptions*) always has to be put in curly braces.

The attributes, their default values and meaning:

generateOnClick
> 1: The event "onclick" is available in the SVG file (default).
>
> 0: The event "onclick" is not available in the SVG file.

generateOnMouseDown
> 1: The event "onmousedown" is available in the SVG file.
>
> 0: The event "onmousedown" is not available in the SVG file (default).

generateOnMouseUp
> 1: The Event "onmouseup" is available in the SVG file.
>
> 0: The event "onmouseup" is not available in the SVG file (default).

generateOnMouseOver
> 1: The event "onmouseover" is available in the SVG file.
>
> 0: The event "onmouseover" is not available in the SVG file (default).

generateOnMouseMove

>1: The event "onmousemove" is available in the SVG file.

>0: The event "onmousemove" is not available in the SVG file (default).

generateOnMouseOut

>1: The event "onmouseout" is available in the SVG file.

>0: The event "onmouseout" is not available in the SVG file (default).

clickRegionCursorHand

>1: The mouse pointer changes into a hand symbol upon reaching the click region.

>0: The mouse pointer does not change upon reaching the click region (default).

generateConnectorClickRegion

>1: For the connector, its AttrSpots and HotSpots click regions are generated.

>0: No click regions are generated for the connector (default).

<u>**Password restrictions:**</u>

With this option, users changing their password in the ADONIS Business Process Management Toolkit can be forced to enter only passwords of a given type. The language for the password restriction is based on the following syntax:

__PasswordRule :__   **PASSWORD_RULE** *regExpr* **errMsg:***strVal* .

The rule has to be defined as a regular expression (see chap. 4., p. 534). Additionally, an error message has to be specified which is shown if the new password is invalid.

<u>**Example:At least six characters**</u>

```
PASSWORD_RULE ".{6,}"
errMsg:"The password must have a length of at least six characters!"
```

<u>**Set change counter:**</u>

On renaming a model or object, this change takes effect in all INTERREF attributes pointing to the renamed model or object. While the referencing models remain technically unchanged, from the user point of view a change took place. This makes a difference e.g. if using the delta generation function (cf. ADONIS user manual) . In this case, a function has to be set which also marks such models as "changed":

```
INCR_CHANGE_COUNTER_OF_SOURCE_MODELS enable:1
```

>**Hint:** Just to make sure, this function should always be activated, except the computer/network has low capacity and large pool models are renamed on a regular basis.

<u>**Gradient printing:**</u>

Some printer device drivers have problems with printing colour gradients. Sometimes whole gradients or parts of them are missing. If no error-free printer is available, another filling method can be specified using **GRADIENT_PRINTING**. These variants are available:

● **default**: The normal gradient filling is printed.

● **first-color**: Instead of a gradient, a solid filling is printed using the *first* colour of the gradient.

- **avg-color**: Instead of a gradient, a solid filling is printed using the *average* colour of the gradient.

   Hint: These settings apply for printing matters *only* - other displaying forms remain unaffected.

2.1.3.13   Seitenlayouts (Modellierung)

The library attribute "Seitenlayouts" exists in both types of library (BP library and WE library). The page layouts defined here are available for all models of a type defined in the particular library. Usually, the attribute "Seitenlayouts" in an application library has the same value in both the BP and the WE library.

Using the page layout the formats for printing models are defined. One of the layouts defined can be assigned at any time to any model. It will be used when printing the model. The layout can be selected via the "Page layout" option on the printing screen. Among other things, a layout defines the size of the printed page available for the model. The margin resulting from this can be made visible as broken lines on the drawing area so that the ADONIS user can already see the measurements of his model during a modelling session.

Via page layouts, model printing formats are defines. Every model can be assigned one of the defined layouts at any time which is then used for printing. The desired page layout can be selected via the menu entry "Page layout". A layout definition contains e.g. the definition of the printable area on one page. The resulting page borders can be shown as a dashed line in the graphical modelling editor, telling the user if an object would be printed on two pages.

There is no limit for the number of page layouts (except the 3700 characters max length of the STRING attribute type). A new model automatically gets the first defined layout. Three layouts (see p. 206) are defined internally and neither can be deleted nor hidden: The "ADONIS standard page layout", the "ADONIS standard page layout for tables" and "Posters with adhesive tabs". These are always last in the list.

The language for defining page layouts is based on the following syntax:

__PageLayouts :__   { *PageLayout* } .

__PageLayout__ :   **LAYOUT** *layoutName* [**for-models**] [**for-reports**]
          [**orientation:***Orientation*] [**graphrep:***AttrName*]
          { Pen | Fill | HeadArea | BitmapArea | PageArea | FootArea }
          .

   Hint: *HeadArea*, *BitmapArea* and *FootArea* may appear at most once, *PageArea* must appear exactly once per layout.

__Orientation__ :   **portrait** | **landscape** .

__Pen__ :   **PEN** [**w:***width*] [**style:***PenStyle*] [**color:***colorSpec*] .

__PenStyle__ :   **solid** | **dot** | **dash** | **dashdot** | **null** .

__Fill__ :   **FILL** [**style:***BrushStyle*] [**color:***colorSpec*]
          [**fcolor:***colorSpec*] [**transparent**] .

__BrushStyle__ :   **solid** | **horz** | **vert** | **cross** | **diagcross** | **updiag** |
          downdiag | **mix25** | **mix50** | **mix75** | **null** .

__HeadArea__ :   **HEAD** *PosSize*
          { *FixedText* | *ModelAttribute* | *Font* } .

__BitmapArea__ :   **BITMAP** [*bmpFileName*] *PosSize*
          { *FixedText* | *ModelAttribute* | *Font* } .

__PageArea__ :   **PAGE** *PosSize* [**with-flaps**]
          [**left-margin:***width*] [**top-margin:***height*]
          { *FixedText* | *ModelAttribute* | *Font* } .

FootArea :	**FOOT** *PosSize* { *FixedText* | *ModelAttribute* | *Font* } .
FixedText :	**TEXT** *text PosSize* **line-break:***LineBreakMode*] .
ModelAttribute :	**ATTR** *modelAttrName PosSize* [**line-break:***LineBreakMode*].
PosSize :	[*XCoord*] [*YCoord*] [*Width*] [*Height*] .
XCoord :	**x:***xpos* | **x:***XCoordFlags* [**:***xpos*] .
XCoordFlags :	**l** | **c** | **r** .
YCoord :	**y:***ypos* | **y:***YCoordFlags* [**:***ypos*] .
YCoordFlags :	**t** | **c** | **b** .
Width :	**w:***width* | **w:***WidthFlags*[**:***width*] .
WidthFlags :	[**p**] [**l** | **c** | **r**] .
Height :	**h:***height* | **h:***heightFlags*[**:***height*] .
HeightFlags :	[**p**] [**t** | **c** | **b**] .
LineBreakMode :	**off** | **words** | **rigorous**] .
Font :	**FONT** [*fontName*] [**h:***fontHeight*] [**bold**] [**underline**] [**italic**] [**color:***ColorSpec*] .

A layout definition starts with the keyword **LAYOUT**, followed by a name in quotation marks. Since you select the page layout by the name assigned here, the name should somehow characterise the layout.

Two page layouts for two scenarios are distinguished:

for-reports: this parameter defines that the layout is only used for printing tables (browser).

for-models: this parameter defines that the layout is only used for printing model graphics.

If neither **for-reports** nor **for-models** is defined, then the layout can be used for both purposes.

There are four sections in the layout definition which start with the following keywords:

PAGE	Definition of an area on the printed page (model area), available for printing the model. From this, you can also derive the size of the page which may be displayed in the model editor.
HEAD	Definition of position and size of the header. The header can contain texts, model attributes and print attributes such as the printing date.
FOOT	Definition of position and size of the footing. The footing can contain texts, model attributes and print attributes such as the printing date.
BMP	Definition of an area into which a bitmap is inserted.

These sections can have any arbitrary order you wish.

> **Hint:** In any case, **PAGE** must occur once per layout definition, **HEAD**, **FOOT** and **BMP** are optional.

The following elements may be used within the **PAGE**, **HEAD**, **FOOT** and **BMP** sections:

TEXT	Specify text to be displayed in a defined position in the current section. This will usually be the header or the footer.
	The text may contain tokens which are replaced by print attributes during the printing. **%D** is replaced by the printing date, **%P** by the current page number, **%N** by the total number of pages and **%#** by a reference to adjacent pages. In the case of **%#** a directing arrow with the respective page's number is set for each one of the (at most) four adjacent pages.

Please take care that a font which contains arrows as special symbols is specified for this purpose (such as "Symbol" under NT or "Symbol Set" under OS/2).

ATTR   Specify a model attribute to be displayed in the current section at the position specified.

FONT   Change the current font (font type, font height and font style). This effects the following **TEXT** and **ATTR** output.

Using the **FONT** element you can define the font type (*fontName*) by the name following the keyword and the font height (*fontHeight*) by the attribute h. You may also combine the styles **bold**, **underline** and **italic**.

The *XCoordFlags* have the following meaning:

l   Default setting; in the cases of **PAGE**, **HEAD**, **FOOT** and **BMP** the coordinate refers to the left margin of the printed page, in the cases of **TEXT** and **ATTR** to the left margin of the current area.

c   The coordinate refers to the horizontal centre.

r   The coordinate refers to the right margin.

The *YCoordFlags* have the following meaning:

t   Default setting; in the cases of **PAGE**, **HEAD**, **FOOT** and **BMP** the coordinate refers to the upper margin of the printed page, in the cases of **TEXT** and **ATTR** to the upper margin of the current area.

c   The coordinate refers to the vertical centre.

r   The coordinate refers to the bottom margin.

The *WidthFlags* have the following meaning:

p   The printed page's width is added to the width value, i.e. only a width value <= 0 is sensible here. This flag is only effective in the cases of **PAGE**, **HEAD**, **FOOT** and **BMP**. The current page size is taken from the printer's settings.

l   Alignment to the left with regard to the x-coordinate. Default setting for **PAGE**, **HEAD**, **FOOT** and **BMP**.

c   Horizontal centring with regard to the x-coordinate. Default setting for **TEXT** and **ATTR**.

r   Alignment to the right with regard to the x-coordinate.

The *HeightFlags* have the following meaning:

p   The printed page's height is added to the height value, i.e. only a height value <= 0 is sensible here. This flag is only effective in the cases of **PAGE**, **HEAD**, **FOOT** and **BMP**. The current page size is taken from the printer's settings.

t   Alignment to the top with regard to the y-coordinate. Default setting for **PAGE**, **HEAD**, **FOOT** and **BMP**.

c   Vertical centring with regard to the y-coordinate. Default setting for **TEXT** and **ATTR**.

b   Alignment to the bottom with regard to the x-coordinate.

For **BMP** a bitmap file (***bmpFileName***) will be loaded and adjusted in a rectangle through some given coordinates. The BMP file will be loaded from a file system with a name ***fileName***. Please remember to enter a path and a file name of the BMP file in inverted commas.

Should the file specification not contain a path name, the ADONIS directory will be searched for the bitmap file.

   Hint: The backslashes (\) in the path specification of the bitmap file must be masked by '\' as in c:\\bitmaps\\logo.bmp for the file "logo.bmp" in the directory "bitmaps" on the partition "C:").

> **Hint:** For the including of bitmap files which are saved as external files in theADONIS-database (see chap. 15., p. 606)quote as path '**db:**' (e.g. "db:\\logo.bmp" for the in the database saved file "logo.bmp").

> **Hint:** If no bitmap is specified or if the file specified does not exist, the BOC logo will be used.

> **Hint:** The ratio of the sides of the rectangle defined in **BMP** should match the one of the bitmap, otherwise the diagram will be printed in a distorted way.

At the **FONT** element, the font size (font height) will be determined which overrules the assigned name of the font and the attribute **h**.

Furthermore the following styles can be combined as you like:

bold - bold letters.

underline - underlined.

italic - cursiv font.

Over **PEN** and **FILL** analog to the*GraphRep (see chap. 2.1.1.1, p. 131)* - the squaring and filling of the parts **PAGE**, **HEAD**, and **FOOT**can be determined. Default at **HEAD** and **FOOT** is

```
PEN color:black    FILL color:$f0f0f0  # hellgrau, $f0=240
```

but the**PAGE**-square with

```
PEN color:black    FILL style:null
```

.

GraphRep for complex layouts:

It is possible to use GraphRep-Syntax (see p. 134) for defining page layouts. This enables integrating more than one graphic file, using colour gradients and more.

> **Hint:** Complex layouts are only available after a special modification of the application library . Please contact your ADONIS consultant for details.

Internal ADONIS standard page layouts

In addition to the freely definable page layouts of the library attribute "Seitenlayouts" three predefined, fixed layouts are always available:

- The ADONIS Standard page layout

- The ADONIS Standard page layout for tables

- Posters with adhesive tabs

ADONIS Standard page layout

```
LAYOUT "ADONIS standard page layout" for-models
PAGE x:1cm y:2.2cm w:p:-1.5cm h:p:-2.9cm
FILL color:$bfcccc
HEAD x:1cm y:0cm w:p:-1.5cm h:2cm
TEXT "Angelegt am:" x:l:2.5cm y:c w:l:3.5cm
ATTR "Angelegt am" x:l:6.1cm y:c w:l:20cm
TEXT "von:" x:l:9.3cm y:c w:l:20cm
ATTR "Autor" x:l:10.2cm y:c w:l:20cm
TEXT "Letzte Änderung am:" x:l:2.5cm y:b:-0.3cm w:l:3.5cm h:b
```

```
ATTR "Letzte Änderung am" x:l:6.1cm y:b:-0.3cm w:l:20cm h:b
TEXT "von:" x:l:9.3cm y:b:-0.3cm w:l:20cm h:b
ATTR "Letzter Bearbeiter" x:l:10.2cm y:b:-0.3cm w:l:20cm h:b
TEXT "Modelltyp:" x:l:13.5cm y:c w:l:2.9cm
ATTR "Modelltyp" x:l:15.5cm y:c w:l:20cm
TEXT "Status:" x:l:13.5cm y:b:-0.3cm w:l:2.9cm h:b
ATTR "Status" x:l:15.5cm y:b:-0.3cm w:l:20cm h:b
FONT bold underline
ATTR "Name" x:l:2.5cm y:t:0.3cm w:l:15.5cm h:t:.5cm
FONT "Symbol" h:8.0pt
TEXT "%#" x:r:-0.24cm y:t:0.2cm w:r:10cm h:t
BMP x:1.5cm y:0.1cm w:1.34cm h:1.8cm
FOOT x:1cm y:b:0cm w:p:-1.5cm h:b:0.5cm
TEXT "(c) BOC Group, 1996-2006" x:l:0.5cm y:c:-0.03cm w:l:10cm
TEXT "%D" x:c:0cm y:c:-0.03cm w:c:10cm
TEXT "Seite %P / %N" x:r:-0.5cm y:c:-0.03cm w:r:10cm
```

ADONIS Standard page layout for tables:

```
LAYOUT "ADONIS standard page layout for tables" for-reports
FILL color:bfcccc
HEAD x:0cm y:0cm w:p h:2.2cm
TEXT "%T" x:l:2.5cm y:0cm w:l:17cm h:t
FONT "Symbol" h:8.0pt
TEXT "%#" x:r:-0.24cm y:t:0.2cm w:r:10cm h:t
BMP x:0.1cm y:0.1cm w:1.5cm h:2.0cm
FOOT x:0cm y:b:0cm w:p h:b:0.5cm
TEXT "(c) BOC Group, 1996-2006" x:l:0.5cm y:c:-0.03cm w:l:10cm
TEXT "%D" x:c:0cm y:c:-0.03cm w:c:10cm
TEXT "Seite %P / %N" x:r:-0.5cm y:c:-0.03cm w:r:10cm
PEN style:null
FILL style:null
PAGE x:0cm y:2.5cm w:p h:p:-3.3cm
```

Posters with adhesive tabs:

LAYOUT "Posters with adhesive tabs"

PAGE w:p:-1cm h:p:-1cm with-flaps

2.1.3.14 Connector marks - Numerierung / Grafische Darstellung (Modellierung)

The connector mark definition in a library is based on two attributes:

- Numerierung

- Grafische Darstellung

"Numerierung" is of type ENUMERATION and describes, how the connector marks are numbered in the models. Two options are available:

numerical Arabic numerals (1, 2, 3 ...)

alphabetically letters (A, B, C ..., AA, AB, ...)

The graphical representation of the connector marks is defined in the library attribute "Grafische Darstellung". The syntax for it is based on the GraphRep grammar (see p. 134) (graphical representation of classes and relations). A special variable, called **isoutgoing**, is available. With a value of 1, the connector is outgoing, if the value is 0 it is ingoing. This enables a different look for the two connector marks of a pair.

Examples of connector marks:

Circle: The attribute value

```
GRAPHREP ELLIPSE rx:.3cm ry:.3cm
```
creates a circle (e.g.).

Triangle: The attribute value

```
GRAPHREP POLYGON 3 x1:-.3cm y1:.25cm y2:-.35cm x3:.3cm y3:.25cm
```
creates a triangle (e.g.).

Square: The attribute value

```
GRAPHREP RECTANGLE x:-.3cm y:-.3cm w:.6cm h:.6cm
```
creates a square (e.g.).

Pentagon: The attribute value

```
GRAPHREP
FILL color:aliceblue
PEN w:0.03cm
IF (isoutgoing) {
  POLYGON 5 x1:-0.3cm y1:-0.3cm x2:0.3cm y2:-0.3cm
            x3:0.3cm y3:0.25cm x4:0cm y4:0.35cm x5:-0.3cm y5:0.25cm
} ELSE {
  POLYGON 5 x1:-0.3cm y1:-0.25cm x2:0cm y2:-0.35cm x3:0.3cm y3:-0.25cm
            x4:0.3cm y4:0.3cm x5:-0.3cm y5:0.3cm
}
```
creates a pentagon in the shape of a house with different in- and outgoing parts.
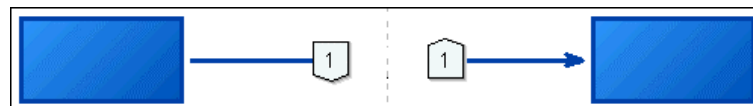
Figure 174: Pair of connector marks

> **Hint:** The numbering of the connector marks is exactly nine pixels high. The breadth depends on the letter(s) or the number displayed.

2.1.3.15 Anordnungsfunktion (Modellierung)

The attribute "Object arrangement" allows you to define the characteristics of one or more object arrangement functions of the process hierarchy function and of the numbering function.

> **Hint:** The way the three function types mentioned above work is documented in the ADONIS user manual.

The language for defining the object arrangement function is based on the following syntax:

__TotalConfiguration :__	[*ArrangeConfiguration*] [*EnumConfiguration*] [*MenuConfiguration*] .
***ArrangeConfiguration* :**	[*Parameterset*] [*Hierarchyset*] .

Parameterset :   `PROFILE "`*Functionname*`"`
                                     [`type:`*Profiletype*]
                                     DefinedModeltypes
                                     Parameters
                                     [*BibParameterset*] .

Profiletype :   `std` | `cust` .

DefinedModeltypes :   `DEFMODELTYPE "`*Modeltypename*`"`
                                     [*DefinedModeltypes*] .

Parameters :   [*BendpointParameters*]
                                     [*MinCrossingParameters*]
                                     [*PendulumParameters*]
                                     [*FlipflyParameters*]
                                     [*DoubleBendpParameters*]
                                     [*ChangesizeParameters*] .

BendpointParameters :   `BEND` [`points` | `edges`] .

MinCrossingParameters :   `MINCROSS`
                                     [`upon`][`upcount:`*countVal*]
                                     [`dwnon`][`dwncount:`*countVal*]
                                     [`pairwise`][`paircount:`*countVal*] .

PendulumParameters :   `PENDULUM`
                                     [`upon`][`upcount:`*countVal*]
                                     [`dwnon`][`dwncount:`*countVal*]
                                     [`rubband`][`rubcount:`*countVal*] .

FlipflyParameters :   `FLIPFLY`
                                     [`mirrhor`]
                                     [`mirrvert`] *FlipflyWay* .

FlipflyWay :   `dwn` | `up` | `toright` | `toleft` .

DoubleBendpParameters :   `DOUBLEBP dist:`*countVal* .

ChangesizeParameters :   `CHNGSIZE vertdist:`*countVal*
                                     `hordist:`*countVal* .

BibParameterset :   { *ClassParameters* }
                                     { *ClassPairParameters* } .

ClassParameters :   `CLASSMODELTYPE "`*Modeltypename*`"`
                                     { *ClassParameterLine* } .

ClassParameterLine :   `CLASSPAR "`*Classname*`"`
                                     `turn:`*boolVal* `space:`*countVal*
                                     `priority:`*countVal* .

ClassPairParameter :   `CLASSPAIRMODELTYPE "`*Modeltypename*`"`
                                     { *ClassPairParameterLine* } .

ClassPairParameterLine :   `CLASSPAIRPAR`
                                     `first:"`*FirstClassname*`"`
                                     `second:"`*SecondClassname*`"`
                                     `xdist:`*countVal*
                                     `ydist:`*countVal* .

Hierarchyset :   `HIERPROFILE` [`"`*HierFunctionname*`"`]
                                     `use:"`*Functionname*`"`
                                     { *HierParameters* } .

HierParameters :   `HIERMODELTYPE "`*Modeltypename*`"`
                                     `usetype:"`*UseModeltypename*`"`

```
                                  { HIERCLASS "Classname" }
                                  HIERATTRIB "Attributename"
                                  HIERUSECLASS "UseClassname"
                                  HIERUSEREL "UseRelname" [ relturn ] .
```

EnumConfiguration :	`ENUMPROFILE [ "EnumFunctionname" ]` `{ Modtypeenumset } .`
Modtypeenumset :	`ENUMMODELTYPE "Modeltypename"` `{ EnumRel | EnumClass } .`
EnumRel :	`ENUMREL "Relname" [ enumturn ] .`

EnumClass :	`ENUMCLASS "Classname"`
	`[ attrib:"UseAttributename" ]`
	`[ inflow ] .`

MenuConfiguration :	`DISABLE [ arrange ][ edit ] [ enum ][ hier ] .`

ArrangeConfiguration defines both the **Object arrangement Function** (see p. 210) and the **Hierarchy function** (see p. 213). The settings of the **Numbering Function** (see p. 214) are specified in *Enum-Configuration*.

> **Hint:** An object arrangement function defined in the application library can be copied in the Business Process Management Toolkit, and can be changed as well as saved as a user-defined object arrangement function in the ADONIS database.

MenuConfiguration provides the possibility to **DISABLE** the menu options for the object arrangement, the model hierarchy and the numbering function in the modelling component of the Business Process Management Toolkit with the following definitions:

arrange   The menu option "Arrange" (menu "Edit") is disabled, i.e. the ADONIS user can neither use the object arrangement functions defined in the application library nor create user-defined object arrangement functions.

edit   The sub-menu option "Arrangement Assistant" (option "Arrange" - menu "Edit") is disabled, i.e. the ADONIS user can use the object arrangement functions defined in the application library but not create any.

enum   The option "Number Objects" (menu "Edit") is disabled, i.e. the ADONIS user can not use the numbering functions defined in the application library.

here   The sub-menu option "Process Hierarchy (graphical)" (option "Search Models" - menu "Model" is disabled, i.e. the ADONIS user can not use the model hierarchy functions defined in the application library.

Settings of the Anordnungsfunktion

When the object arrangement function is executed, the complete graph is traversed to assign all the model's objects to hierarchically structured levels and connect them with relations.

> **Hint:** Internally the arrangement of objects is done from top to bottom, regardless of which modelling direction the corresponding model has. The horizontal and vertical settings also refer to this internal arrangement.

In addition to the **PROFILE element** (see p. 211) and the **DEFMODELTYPE element** (see p. 211) general and class-specific parameters will be defined for every object arrangement function. The following list contains the **general parameters** in the order in which they are processed when the object arrangement function is executed:

1. **BEND** (see p. 211)

2. **MINCROSS** (see p. 211)

3. **PENDULUM** (see p. 212)

4. **FLIPFLY** (see p. 212)

5. **DOUBLEBP** (see p. 213)

6. **CHNGSIZE** (see p. 213)

The **class-specific parameters CLASSPAR** (see p. 213) or **CLASSPAIRPAR** (see p. 213) allow additional settings for the relations and classes of a model type.

PROFILE

The **PROFILE** element introduces the definition of an object arrangement function. An object arrangement function consists of various parameters (*Parameterset*) and is marked by a non-ambiguous name (*Functionname*).

The attribute **type** serves to classify the object arrangement function.

● **type:cust** means that the object arrangement function can be changed by the ADONIS user.

● **type:std** means that the object arrangement function can **not** be changed by the ADONIS user.

   Hint: If the attribute **type** is left undefined the value assumed is **type:cust**.

DEFMODELTYPE

Using **DEFMODELTYPE** you can specify to which model types (*Modeltypename*) the object arrangement function generally apply.

   Hint: For every model type defined you can make further settings for the arrangement of objects and relations (**CLASSPAR**) of this model type.

The parameter **BEND**, **MINCROSS**, **PENDULUM**, **FLIPFLY**, **DOUBLEBP** and **CHNGSIZE** control the general settings of the object arrangement function.

BEND

The bendpoint settings can be switched with the help of the parameter **BEND**. This causes the relations of the model to be supplemented by bendpoints.

BEND points   Connectors which connect objects of non-adjacent layers will be arranged around the objects of the directly following layers. This minimises the probability of objects or connectors intersecting each other. This "diversion" is created by inserting bendpoints.

BEND edges   As far as possible, the connectors between the objects are drawn by horizontal and vertical lines only. This representation of connectors is reached by inserting bendpoints.

MINCROSS

With the help of the parameter **MINCROSS** the objects of one level can be moved onto this level to minimise the number of intersections among connectors.

upon upcount:*countVal*
     Method "Upward Median"; here the minimisation of intersections is carried out from the hierarchically lower to the hierarchically higher levels. *countVal* specifies the maximum number of iterations and serves to restrict the calculating time in case of very large models.

> **Hint:** If the number of iteration steps is too small this can influence the arrangement quality to its disadvantage. Default value: 10.

dwnon dwncount:*countVal*

> Method "Downward Median"; here the minimisations of intersections is carried out from the hierarchically higher to the hierarchically lower levels *countVal* specifies the maximum number of iterations and serves to restrict the calculating time in case of very large models.
> **Hint:** If the number of iteration steps is too small this can influence the arrangement quality to its disadvantage. Default value: 10.

pairwise paircount:*countVal*

> Method "Pairwise Interchange"; here every two adjacent objects of the same level are interchanged. Then it is checked whether the number of intersections could be decreased. *countVal* specifies the maximum number of iterations which helps to restrict the calculating time in case of very large models
> **Hint:** If the number of iteration steps is too small this can influence the arrangement quality to its disadvantage. Default value: 10.

PENDULUM

With the parameter **PENDULUM** objects can - depending on preceding or succeeding objects - be placed on different layers, so that the flow can be represented in one line.

upon upcount:*countVal*

> Method "Upwards Swinging"; here the objects of one level are oscillated according to the position of preceding objects, that means they are arranged in a "line".*countVal* specifies the maximum number of iterations and serves to restrict the calculating time in case of very large models.
> **Hint:** If the number of iteration steps is too small this can influence the arrangement quality to its disadvantage. Default value: 10.

dwnon dwncount:*countVal*

> Method "Downwards Swinging", here the objects of one level are oscillated according to the position of succeeding objects, that means they are arranged in a "line". *countVal* specifies the maximum number of iterations which helps to restrict the calculating time in case of very large models.
> **Hint:** If the number of iteration steps is too small this can influence the arrangement quality to its disadvantage. Default value: 10.

rubband rubcount:*countVal*

> Method "Rubberband", here the objects of one level are oscillated according to the position of the preceding and subsequent objects. *countVal* specifies the maximum number of iterations serves to restrict the calculating time in case of very large models.
> **Hint:** If the number of iteration steps is too small this can influence the arrangement quality to its disadvantage. Default value: 10.

FLIPFLY

The parameter **FLIPFLY** can be used to mirror the objects arranged either horizontally or vertically. Also, the direction of the arrangement can be defined.

mirrhor	The objects arranged are mirrored horizontally.
mirrvert	The objects arranged are mirrored vertically.
dwn	The objects are arranged in the processing direction from top to bottom.
up	The objects are arranged in the processing direction from bottom to top.
toright	The objects are arranged in the processing direction from the left to the right.
toleft	The objects are arranged in the processing direction from the right to the left.

DOUBLEBP

With the parameter **DOUBLEBP** additional connector sections can be inserted to prevent intersections between objects and connectors. **dist:***countVal* specifies the length of the sections to be inserted in cm.

CHNGSIZE

With the parameter **CHNGSIZE** the drawing area can be adjusted to the space needed by the newly arranged model. The size needed is derived from the number of objects and the settings of the column's (**vertdist:***countVal*) and the lines' (**hordist:***countVal*) distances. *countVal* specifies the distance in cm.

> **Hint:** Should the parameter CHNGSIZE not be set, the distances between a model's objects will be calculated in such a way that the model fits into the previous drawing area.
>
> Should the drawing area be too small the objects arranged could intersect each other.

CLASSPAR

The parameter **CLASSPAR** specifies how the relations of a model type are represented. For this purpose you have to enter the name of the model type (*Modeltypename*) into the **CLASSMODELTYPE-Element**.

The parameter **CLASSPAR** contains the following attributes beside the name of the relation class (*Classname*):

space:*countVal*	Indicates the number of levels skipped when inserting bendpoints.
turn:*boolVal*	Specifies whether the relation's direction will be internally reversed (**turn:1**) or not (**turn:0**).
priority:*countVal*	Specifies the priority when a relation is selected.

CLASSPAIRPAR

The parameter **CLASSPAIRPAR** specifies how pairs of objects of a model type are represented. For this purpose, you have to enter the name of the model type (*Modeltypename*) into the **CLASSPAIRMODELTYPE-Element**.

The parameter **CLASSPAIRPAR** contains the following attributes:

first:"*FirstClassname***"**	Specifies the name (*FirstClassname*) of the pair's first class.
second:"*SecondClassname***"**	Specifies the name (*SecondClassname*) of the pair's second class.
xdist:*countVal*	Specifies the horizontal distance between the first and the second class.
ydist:*countVal*	Specifies the vertical distance between the first and the second class.

Settings of the model hierarchy function

The expression **HIERPROFILE** introduces the model hierarchy's definition. A name (*HierFunctionname*) can be entered for the model hierarchy function.

The model hierarchy function is based on an arrangement function already defined. The name (*Functionname*) of the arrangement function (see **PROFILE** (see p. 211)) is specified by the expression **use**.

The parameter **HIERMODELTYPE** specifies the model type's name (*Modeltypename*) originating from which the model hierarchy has to be generated. The name of the model type (*UseModeltypename*) on which the resulting model is based should be entered into the expression **usetype**.

> **Hint:** The model type for the resulting model (*UseModeltypename*) also has to be selected in the arrangement function referenced (*Functionname*).

In **HIERCLASS** the names of those classes (*Classname*) are entered in the inter-model references for which a subordinated model for the model hierarchy are defined. The attribute (*Attributename*) of the class specified in **HIERCLASS**, in which the inter-model reference is entered, is specified in **HIERATTRIB**.

In **HIERUSECLASS** the name of the class (*UseClassname*), and in **HIERUSEREL** the name of the relation (*UseRelname*) which has to be used for the graphic representation of the model hierarchy is specified. By use of **relturn** you can reverse the direction of the relation used (*UseRelname*) in the representation of the model hierarchy.

The name of the model referenced is entered into the attribute (*UseAttributename*) specified in **HIERUSEATTRIB**.

Settings of the numbering function

The expression **ENUMPROFILE** followed by a name (*EnumFunctionname*) introduces the definition of the numbering function.

The parameter **ENUMMODELTYPE** specifies the name of the model type (*Modeltypename*) for the models to be numbered.

For every model type defined, the classes (*Classname*) of the objects to be numbered can be defined in **ENUMCLASS** and the relations (*Relname*) of the connectors linking these objects can be defined in **ENUMREL**. Using the expression **enumturn** the direction of the relation used can be reversed in order to traverse the model graph. Through **attrib** you can specify the attribute (*UseAttributename*) for every class selected (*Classname*) into which the result of the numbering is to be entered. Through **inflow** instead, you can produce a class-independent numbering based on the direction of the process flow.

2.1.3.16   Beziehungsauswertungen (Analyse)

With the help of relation analysis you can create relation tables which show for a model of a given type the existing connectors of one class of relations between two object classes. The term "relation" has a wide meaning in this context. The relation tables are based on AQL, i.e., to determine which relations exist AQL is being used. The relation tables can be selected in the Analysis Component of the Business Process Management Toolkit via the menu "Relation tables".

The language for the definition of relation tables is based on the following syntax:

Tabledefinition : { *Table* } .

Table :   **RELATIONTABLE "***Tablename***"**
             [**modeltype:"***Modeltype-name***"**]
             fromclass:"*FromClass-name***"**
             fromattribute:"*FromAttribute-name***"**
             [**tomodeltype:"***ToModeltype-name***"**]
             toclass:"*ToClass-name***"**
             toattribute:"*ToAttribute-name***"**
             Aql | *Rel* .

***Rel*:**	```relation:"```*Relation-name*```"``` [```attribute:"```*Relationattribute-name*```"] .```
***Aql*:**	{ ```EXPR "```*Aql expression part*```"	FOREACHFROM``` } .

There are two possible definitions for relation tables:

- The relation between two classes (the "Rel" case)

- Via AQL expression (the "Aql" case)

The relation between two classes (the "Rel" case)

Its semantics are as follows:

A relation table presents all existing connectors of the given relation between objects from two different classes. However, it can be done for only one model (due to the defined *fromclass* and *toclass*, which defines in this case a starting and a target class).

RELATIONTABLE element defines the name of the relation table (*Tablename*), which is then shown in menu "Relation tables".

> **Hint:** Names of relation tables must be clearly defined in an application library!

If desired, the system can also display the model type (*Modeltypename*).

> **Hint:** If the model type is not displayed, it means that the standard model type (= on the first place of library attribute "Modus", belonging to the defined modeltype (see chap. 2.1.3.9, p. 191)) was not accepted in the particular library.

The value of the objects which are shown in the ADONIS browser are defined via *fromattribute* or *toattribute*. It allows you to visualise other attributes as object names. The existence of a connector between two objects is represented with an 'x'. If there is an *attribute*, the system will show the value of this attribute.

Example:

```
RELATIONTABLE "Activity-Resource table"
fromclass:"Activity"
fromattribute:"Name"
toclass:"Ressource"
toattribute:"Name"
relation:"uses"
```

This definition shows each connector and the relation "uses", which exists between activities and resources.

Via AQL expression (the "Aql" case)

An AQL expression can be used here in order to present one (possibly transitive and model predominant) relation between any two classes. In this case these two classes can come from two different model types. The AQL expression is dynamically built up from **FOREACHFROM** and **EXPR** and evaluated for each object of the starting class. The part **FOREACHFROM** presents reference to an object of the starting class and is filed with the current object name. **EXPR** elements show those parts of the AQL expression, which are fixed and independent from the changing object names.

The name of the relation table (*Tablename*) is defined in the **RELATIONTABLE element**. This name is shown in Menu "Relation tables".

> **Hint:** Relation table names must be clear in the application library!

If desired, the system can also display the model type (*Modeltypename*).

> **Hint:** If the model type is not defined, it means that the standard model type (= on the first place of library attribute "Mode", belonging to the defined modeltype (see chap. 2.1.3.9, p. 191)) is taken over to the current library.

Similarly to the Rel - case the target class defines the max number of objects of which relations will be shown. If the class is not in the same model type as the starting class, *tomodeltype* must be given. Due to this, the target class should be clearly identified via its name. It is also possible to visualise any attribute of the objects, by using *fromattribute* or *toattribute*.

<u>Ways of defining relation tables via FOREACHFROM and EXPR</u>

In order to define relation tables with **FOREACHFROM** and **EXPR** the user has to define the AQL expression, which will display desired results for the concrete object of the starting class. In order to show e. g., which resources are used by the activity-4711, you need to use the following AQL expression:

```
{"Activity-4711"} -> "uses"
```

or alternatively

```
{"Activity-4711" : "Activity"} -> "uses"
```

.

Each part of the above AQL expression which contains names (eventually with the class) of the objects should be replaced with **FOREACHFROM** construction, because the final relation table should display all objects of the starting class. It will allow the dynamic use of the names of all instances.

Consequently, the AQL expression must be divided: {"Activity-4711"} or {"Activity-4711":"Activity"} will be replaced with **FOREACHFROM** and the rest with **EXPR**. As the other LEO elements (in the opposite to the above example) remain unchanged, the following LEO syntax is used:

```
RELATIONTABLE "Activities-Resource tables"
fromclass:"Activity"
fromattribute:"Name"
toclass:"Resource"
toattribute:"Name"
FOREACHFROM
EXPR "-> \"used\""
```

If a table showing who is the manager of whom should be created (for working environment), it should be started with a concrete performer:

```
{"Performer-1"} -> "is manager" <- "belongs to"
```

The above expression displays all performers of an organisational unit, where performer-1 is the manager. If we presume that the organisational unit is modelled in such a way that the manager of the unit (additionally) belongs to it, it is reasonable to remove this particular performer from the list with results, as nobody is manager of himself:

```
{"Performer-1"} -> "is manager" <- "belongs to" DIFF {"Performer-1"}
```

The dynamic parts must be replaced (i.e. the names of the concrete instances) - here are two examples of such a case. The rest is a part of EXPR:

```
RELATIONTABLE "Manager table"
```

```
fromclass:"Performer"
fromattribute:"Name"
toclass:"Performer"
toattribute:"Name"
FOREACHFROM
EXPR "-> \"is manager\" <- \"belongs to\" DIFF "
FOREACHFROM
```

Other examples:

To define a relation table between variables and activities you can use the following LEO syntax:

```
RELATIONTABLE "Variables-Table with activities"
fromclass:"Variable"
fromattribute:"Name"
toclass:"Activity"
toattribute:"Name"
FOREACHFROM
EXPR "<- \"create variable\" -> \"create\""
```

If the user reduces the relation table to the variables, which are used as objects occupying variables, and which are divided according to the discrete distribution, he can use the following LEOgram:

```
RELATIONTABLE "Variables-Activities table"
fromclass:"Variable"
fromattribute:"Name"
toclass:"Activity"
toattribute:"Name"
FOREACHFROM
EXPR "<- \"created variable\" [?\"Value\" like \"Discrete*\"] -> \"created\""

RELATIONTABLE "Activity-Organisational table"
fromclass:"Activity"
fromattribute:"Name"
tomodeltype:"Working Environmentsmodel"
toclass:"Organisational unit"
toattribute:"Name"
FOREACHFROM
EXPR " --> \"Organisational unit\""
```

2.1.3.17   Simulation definition - Simtext (Simulation)

The attribute **Simtext** contains some user-specific expressions. These are used by ADONIS to label simulation results . The Simtext is used ...

●   ... in the Path Analysis: in the result text

●   ... in the Capacity Analysis, the Workload Analysis and the Analytical Evaluation: for labelling the table columns

    Hint: The attribute "Simtext" is only defined for BP libraries.

The language describing the definition of the Simtext is based on the following syntax:

__Simtext :__   **SIMTEXT undefined** | *Settings* .

Settings **:**   **bp:** "*term for <business process>*"

                cycletime: "*term for <cycle time>*"

                activity: "*term for <activity>*"

> **number:** "*term for <number (count)>*"
>
> **actor:** "*term for <person>*"
>
> **perscost:** "*term for <personnel costs>*"
>
> **resource:** "*term for <resource>*"
>
> **rescost:** "*term for <resource costs>*"

Hint: The term "**undefined**" is automatically assigned, if an invalid simtext is detected during the conversion of an application library from ADONIS Version 3.81 or earlier to ADONIS Version 3.9. This expression causes ADONIS to ignore the complete simtext. In this case, you can neither execute a simulation nor an analytical evaluation unless the simtext is corrected and the expression "**undefined**" removed.

ATTENTION: The Library check (see chap. 2.2.2, p. 280) returns an error, if Simtext contains neither a definition nor "undefined"!

2.1.3.18   Simulation definition - Simmapping (Simulation)

"**Simmapping**" is an attribute of the BP library. In this attribute, input sets for the Simulation and the Analytical Evaluation are defined. Furthermore, a group of classes can be defined which are then used in simulation-related Actions.

The language for the definition of the SimMapping is based on the following syntax:

SimMapping :	{ *SimOption* } [*SimClasses*] .
***SimOption* :**	**SIMOPTION [invalid]**
	name: "*option name***"**
	[{ **name_***<iso639code>***: "***option name in ISO639 language***" }]**
	activity: "*name of activity-class***"**
	[**helptext: "***info/help text for option***"]**
	[{ **helptext_***<iso639code>***: "***info/help text for option in ISO639 language***" }]**
	[**executiontime: "***attribute name of execution time***"]**
	[**waitingtime: "***attribute name of waiting time***"]**
	[**restingtime: "***attribute name of resting time***"]**
	[**transporttime: "***attribute name of transport time***"]**
	[**userattribute-1: "***additional attribute name 1***"]**
	[**userattribute-2: "***additional attribute name 2***"]**
	...
	[**userattribute-n: "***additional attribute name n***"]**
	[*PerformerAssignment (for Subprocesses)*]
	{ *SimActions* }
***PerformerAssignment (for Subprocesses)* :**	**processcall: "** *class name of subprocess call***"**
	subperformerattr: "*attribute name of the default performer assignment for subprocesses***" .**
***SimActions* :**	**ACTION class: "***class name***"**
	attribute: "*attribute name***"**
	[**event: start | interrupt | continue | finish .**
***SimClasses* :**	**SIMCLASSES**
	bp-all | bp-none |
	[**bp-1: "***bp class name***"**
	bp-2: "*bp class name***"**
	...
	bp-n: "*bp class name***"]**
	we-all | we-none |
	[**we-1: "***we class name***"**

```
                    we-2: "we class name"
                    ...
                    we-n: "we class name" ]
```

Hint: The attribute defined with **subperformerattr** must be of type "Text" (STRING) or "Expression" (EXPRESSION).

ACTION

Following this key word you can get process class ("**class**") and attribute of the type "Program call" ("**attribute**"). During simulation each object of the process class, which is processed, will automatically execute the proper program call. Optionally, the **event:** use can indicate when during simulation the process call should be used. However, the following events can be evaluated:

● **start**, when the process object that your dealing with was started,

● **interrupt**, when the process object that your dealing with is cancelled,

● **continue**, when the process object that was cancelled is continued,

● **finish**, when the process object that was cancelled was ended.

"**event:start**" is the standard value.

bp class name

Name of a class from your ADONIS BP library. With the expression **bp-***nr*: "*bp class name*" you may specify as many classes as you wish, where *nr* is a serial number. This means that the additional class has to be specified by **bp-1**, the second class by **bp-2** and so on.

The classes specified in this manner are of importance for the ADONIS evaluation agents: only instances of the classes specified can be assigned as reference objects to the agents (in the ADONIS standard application library these classes are "Activity" and "Subprocess").

The specification of **bp-all** instead of **bp-***nr* is equivalent to specifying all classes from your ADONIS BP library. When an ADONIS BP library is migrated from ADONIS Version 3.81 or earlier to ADONIS Version 3.9 the expression **bp-all** is automatically generated. We recommend that you replace this expression with a sensible selection of classes and do not use it when customising manually.

we class name

Name of a class from your ADONIS WE library. Using the expression **we-***nr*: "*WEClassName*" you can specify as many classes as you wish, where *nr* is a serial number. That means, the first class has to be specified by **we-1**, the second class by **we-2** and so on.

Classes specified in this manner affect the assignment of agents and the result output of the simulation and the agents in the ADONIS Business Process Management Toolkit:

No classes are available for agent assignment which have not been specified in this way.

The result windows of the simulation and of the agents allow the aggregated representation of results which refer to the Working Environment. No class which has not been specified before as described above is available as aggregation criterion.

Hint: Neither in the assignment of agents nor in the respective result windows are necessarily **all** classes specified available for selection.

The specification of **we-all** instead of **we-***nr* is equivalent to the specification of all classes from your ADONIS WE Library. When an ADONIS WE library is migrated from ADONIS Version 3.81 or earlier to ADONIS Version 3.9 the expression **we-all** is automatically generated. We recommend that you replace this expression with a sensible selection of classes and do not use it when customising manually.

Multilinguality:

It is possible to assign values in different languages to **name** and **helptext**. The suffix "_<iso639code>" (e.g. `helptext_en` for English) must then be set. Upon starting ADONIS, the corresponding name/help text is used.

Leaving Simmapping undefined:

To formally correct leaving Simmapping undefined, the attribute must have the value:

```
SIMPOTION invalid
SIMCLASSES bp-none we-none
```

> **Hint:** The expression "**invalid**" is automatically assigned, when an incorrect combination of input parameters is detected while an application library is migrated from ADONIS Version 3.81 or earlier to ADONIS Version 3.9. The expression causes ADONIS to ignore the *SimOption* specified this way.

2.1.3.19 Simulation definition - Simergebnis-Mapping (Simulation)

The attribute "**Simergebnis-Mapping**" defines which simulation results are written back into which attributes of a model when you click on the "Evaluation" button.

The language which defines the Sim result mapping for the **business process library** is based on the following syntax:

__Simresultmapping :__ [**PROCESSSTART "***Name_of_processstart_class***"**
     [**fixedinfo:"***Name_of_info_attribute***"**]
     [**fixedcycletime:"***Name_of_cycletime_attribute***"**]
     [**fixedpersonalcosts:"***Name_of_personalcosts_attribute***"**]
     { **FROMCLASS "***Name_of_fromclass***"**
       fromattribute:"*Name_of_fromattribute***"**
       toattribute:"*Name_of_toattribute***"** }]
     [**ACTIVITY "***Name_of_activity_class***"**
     [**fixedinfo:"***Name_of_info_attribute***"**]
     [**fixednumber:"***Name_of_number_attribute***"**]
     [**Fixedpersonalcosts:"***Name_of_personalcosts_attribute***"**]
     { **FROMCLASS "***Name_of_fromclass***"**
       fromattribute:"*Name_of_fromattribute***"**
       toattribute:"*Name_of_toattribute***"** }] .

The simulation results can only be transferred into the process start class (*ProcessStartClassName*) defined in **PROCESSSTART** or in the activity class (*ActivityClassName*) defined in **ACTIVITY**.

> **Hint:** Should the simulation results be transferable to several instantiable process-start or activity classes, the sim result mapping must be defined on a common basis (parent) class.

The values of the "fixed" attributes listed below do not result directly from other attributes:

fixedinfo   Attribute of the Type STRING (see p. 174), in which information about the origin of the results (e.g. simulation algorithm, type of result, ADONIS user, date and time of the simulation, number of cycletimes and so on) is entered.

fixedcycletime   Attribute of Type TIME (see p. 175), where the total cycletime of the process is stored.

fixedpersonalcosts   Attribute of Type DOUBLE (see p. 174), where the total personnel costs of a process or the personnel costs of an activity are saved.

fixednumber   Attribute of Type INTEGER (see p. 173), where the frequency of occur-
                           rence of an activity is stored.

      ATTENTION: The "fixed" attributes have to be defined for the particular attribute type.

By **FROMCLASS** additional classes (*FromClassname*) may be selected and from the **fromattribute**
attribute values (*FromAttributename*) specified. The selected attributes of this class can be transferred
back through **toattribute** into the respective attribute (*ToAttributename*).

The language which describes the definition of the sim result mapping for the **Working Environment
library** is based on the following syntax:

__Simresultmapping :__ [**PERSON "***Name_of_person_class***"**
                          [**fixedinfo:"***Name_of_info_attribute***"**]
                          [**fixedworkload:"***Name_of_workload_attribute***"**]
                          [**fixedcapacity:"***Name_of_capacity_attribute***"**]
                          [**fixedpersonalcosts:"***Name_of_personalcosts_attribute***"**]
                          { **FROMCLASS "***Name_of_fromclass***"**
                            fromattribute:"*Name_of_fromattribute***"**
                            toattribute:"*Name_of_toattribute***"** }] .

Simulation results can only be transferred into those performer classes (*PersonClassName*) defined in
PERSON.

    Hint: Should the simulation results be transferable to several instantiable performer classes,
        the simresult-mapping must be defined on a common basis (parent) class.

The values of the "fixed" attributes listed below do not directly result from other attributes:

fixedinfo   Attribute of the type STRING (see p. 174), where information about the ori-
                          gin of the results (e.g. simulation algorithm, type of result, ADONIS user,
                          date and time of the simulation, number of cycletimes and so on) is stored.

fixedworkload   Attribute of type DOUBLE (see p. 174), where the workload of a performer is
                          stored.

fixedcapacity   Attribute of type DOUBLE (see p. 174), into which the capacity of a per-
                          former is stored.

fixedpersonalcosts   Attribute of type DOUBLE (see p. 174), into which the personnel costs of a
                          performer are entered.

      ATTENTION: The "fixed" attributes must be defined for the corresponding attribute type.

By **FROMCLASS** additional classes (*FromClassname*) may be selected and by **fromattribute** attrib-
ute values (*FromAttributename*) specified. The selected attributes of this class can be transferred back
through **toattribute** into the respective attribute (*ToAttributename*).

2.1.3.20   Simulation definition - Variablenprüfung (Simulation)

The attribute "**Variablenprüfung**" defines whether the models have to undergo an **extended consis-
tency check** before the simulation.

Selecting **"on"** (default setting) causes an appropriate error statement to occur, if uninitialised vari-
ables are used in the models to be simulated.

When the variable check is switched off (**"off"**), the simulation can also run if not initialised variables
exist.

2.1.3.21 Agenten-Definition (Simulation)

Knowledge of the agent evaluation concept is a necessary pre-condition for understanding the description of agent customising. For this, read the general introduction to the ADONIS evaluation agents in the ADONIS user manual.

With the help of the attribute "Agent definition" evaluation agents may be predefined in the application library - in part or in total. These predefined agents are then available in the window "Agents overview" (see fig. 175) in the Simulation Component of the Business Process Management Toolkit. They serve as templates ("Agent Types") for the creation of other agents.
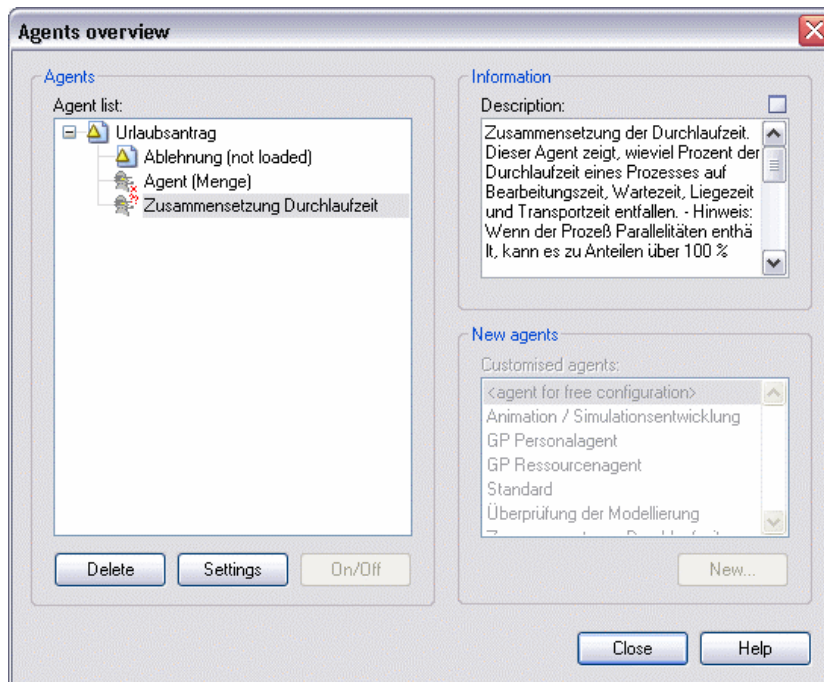
Figure 175: Definition of agents (Simulation)

Agent types are defined by entering text in the agent definition language described below (see p. 222). Each type of agent is described by a separate paragraph. When defining several agents a number of paragraphs may be linked to each other.

> **Hint:** The maximum length allowed for the definition of an agent is 32,000 symbols.

Basically, the definition of an agent types always consists of two parts:

1. Specification of general setting concerning the agent as a whole.

2. Selection and configuration of the results to be determined. Each agent can calculate as many results as you wish. Specific settings have to be made during the customising for each of these results.

An example of the agent definition language (see p. 237) with comments explains the connections and definition.

Syntax for the agent definition

The language for agent definition is based on the following syntax:

Agenten-Definition :	`{ AgentDefinition } .`
AgentDefinition :	`AGENT "AgentTypeName"`

```
                              [ SimAlgorithms ]
                              [ agent-class: "AgentClass" ]
                              [ lock-select-results ]
                              [ hide-select-results ]
                              [ auto-showextremum ]
                              [ lock-extremum ]
                              [ hide-extremum ]
                              [ auto-buildsum ]
                              [ lock-buildsum ]
                              [ hide-buildsum ]
                              [ auto-group ]
                              [ grouping-class: "ClassName"
                                grouping-relation: "RelationName" ]
                              [ lock-group ]
                              [ hide-group ]
                              [ calendar: "CalendarString" ]
                              [ lock-calendar ]
                              [ hide-calendar ]
                              [ objects: "AQLTerm" ]
                              [ lock-objects ]
                              [ hide-objects ]
                              [ infotext: "Text" ]
                              [ lock-infotext ]
                              [ deactivate ]
                              [ allowed-modeltype-1 ... n: "ModelTypeName" ]
                              { Results } .
```

SimAlgorithms :
```
                              [ path-analysis ]
                              [ volume-analysis ]
                              [ wlstatic-analysis ]
                              [ wlnonstatic-analysis ] .
```

Results :
```
                              { Animation | ExecutionCosts |
                              ExecutionCosts (limited) | Workload |
                              Strain | CycleTime |
                              Frequency | Formula |
                              SumOverAttributes (AVG Sum) | NumberOfChangesInRespon-
                              sibility (RespChange) } .
```

Animation :
```
                              ANIMATION
                              name: "ResultName = Title of the Animation Window) "
                              [ hide-name ]
                              [ lock-name ]
                              [ SimAlgorithms ]
                              [ hide-animated ]
                              [ lock-animated ]
                              { ReferencedResult } .
```

ExecutionCosts :
```
                              WWECOST
                              BasicResultSettings
                              [ class: "ClassName" ]
                              [ hide-class ]
                              [ lock-class ]
                              [ attribute: "AttributeName" ]
                              [ hide-attribute ]
                              [ lock-attribute ] .
```

ExecutionCosts (limited) :
```
                              UWWECOST
                              BasicResultSettings
                              [ class: "ClassName" ]
                              [ hide-class ]
                              [ lock-class ]
```

```
                              [ attribute: "AttributeName" ]
                              [ hide-attribute ]
                              [ lock-attribute ] .
```

Workload :

```
                              WORKLOAD
                              BasicResultSettings
                              [ class: "ClassName" ]
                              [ hide-class ]
                              [ lock-class ] .
```

Capacity :

```
                              STRAIN
                              name: "ResultName"
                              [ hide-name ]
                              [ lock-name ]
                              resultscope:DetailSpec
                              [ hide-resultscope ]
                              [ lock-resultscope ]
                              [ dont-show-result ]
                              [ class: "ClassName" ]
                              [ hide-class ]
                              [ lock-class ] .
```

CycleTime :

```
                              CYCLETIME
                              BasicResultSettings .
```

Frequency :

```
                              FREQUENCY
                              BasicResultSettings
                              absolute | relative
                              [ hide-mode ]
                              [ lock-mode ]
                              [ ignore-not-executed-objects ]
                              [ hide-ignore ]
                              [ lock-ignore ] .
```

Formula :

```
                              FORMULA
                              name: "ResultName"
                              [ hide-name ]
                              [ lock-name ]
                              [ SimAlgorithms ]
                              [ expression: "CalcFormula"
                                [ hide-expression ]
                                [ lock-expression ]
                                { ReferencedResult } ] .
```

SumOverAttributes :

```
                              AVGSUM
                              BasicResultSettings
                              [ class: "ClassName" ]
                              [ lock-class ]
                              [ hide-class ]
                              [ attribute: "AttributeName"
                                attribute-type: numeric | time ]
                              [ lock-attribute ]
                              [ hide-attribute ] .
```

NumberOfChangesInResponsibility :

```
                              RESPCHANGE
                              BasicResultSettings
                              [ class: "ClassName" ]
                              [ lock-class ]
                              [ hide-class ] .
```

BasicResultSettings :

```
                              name: "ResultName"
                              [ hide-name ]
```

```
                                    [ lock-name ]
                                    [ SimAlgorithms ]
                                    [ hide-algorithms ]
                                      resultscope: DetailSpec
                                    [ hide-resultscope ]
                                    [ lock-resultscope ]
                                    [ dont-show-result ]
                                    [ hide-show-result ]
                                    [ lock-show-result ]
                                    [ history-mem | history-disk ]
                                    [ hide-history-mem ]
                                    [ lock-history-mem ]
                                    [ hide-history-disk ]
                                    [ lock-history-disk ]
                                    [ history-file: "FileName" ]
                                    [ dont-show-history ]
                                    [ hide-show-history ]
                                    [ lock-show-history ] .
```

***DetailSpec* :**	**process** │ **mainobjects** │ **allobjects** .
***ReferencedResult* :**	**OBSERVED "**ResultName**"** .
***CalcFormula* :**	*FormulaElement* [*BasicOperator CalcFormula*] .
***FormulaElement* :**	\"*ResultName*\" │ *Number* │ *AdvancedOperator1*(*CalcFormula*) │ *AdvancedOperator2*(*CalcFormula*, *CalcFormula*) │ *cond*(*Condition*, *CalcFormula*, *CalcFormula*) .
***BasicOperator* :**	**+** │ **-** │ * │ **/** .
***AdvancedOperator1* :**	**abs** │ **acos** │ **asin** │ **atan** │ **cos** │ **cosh** │ **exp** │ **log** │ **log10** │ **sin** │ **sinh** │ **tan** │ **tanh** │ **sqrt** .
***AdvancedOperator2* :**	**max** │ **min** │ **pow** .

All expressions occurring in agent definition (see p. 225) are explained In the following. It is noted for each expression, whether it is obligatory or optional in an agent definition.

Lock/Hide Parameters

For many settings there are **lock** and **hide** parameters (as in **lock-select-results**, **lock-class** and others). Using a **lock** parameter prevents among other things the ADONIS user from later changing a particular setting.

Using the **hide** parameter prevents among other things the particular control elements needed for changing these settings from being displayed in the windows of the Business Process Management Toolkit.

> **Hint:** In many cases control elements not only serve to change a setting but also display the current setting. Therefore, it can sometimes be more sensible to use a **lock** parameter instead of a **hide** parameter.

AGENT

Every predefined agent type has to start with the keyword **AGENT**. This is followed by the name of the agent type (*AgentTypeName*) which you can select according to your preferences and which is followed by the actual agent definition.

Example:

```
AGENT "Agent for cycletime and costs"
```

By entering the simulation algorithms for the agent you define among which algorithms the agent will be available. You may of course specify several simulation algorithms here.

> **Hint:** If no algorithm is specified, the agent will - as a default - be available within all simulation algorithms.

- **path-analysis** indicates the path analysis.

- **volume-analysis** indicates the capacity analysis

- **wlstatic-analysis** indicates the workload analysis (steady state).

- **wlnonstatic-analysis** indicates the workload analysis (fixed time period).

> **Hint:** The results determined by the agent may have further implications on the selection of the simulation algorithm.

Example:

A basic rule is that the result "Capacity" can only be calculated by the simulation algorithm "Capacity analysis". If you use an agent to calculate the "Capacity" and specify that it is valid for all simulation algorithms, the capacity can't then be calculated by the "Path analysis" simulation.

```
AGENT "Agent for cycletime and costs"
 path-analysis
 volume-analysis
 ...
```

Should the ADONIS application library contain several types of agents, you can specify in **agent-class** to which agent class (*AgentClass*) this agent will be assigned. If no agent class is specified, the ADONIS user must select one when he is creating the agent.

The parameter **lock-select-results** prevents predefined results from being removed from this agent and new results from being added by ADONIS users.

The parameter **hide-select-results** has the same effect as "lock-select-results". In addition, the button needed for adding and removing results is not displayed in the "Basic configuration" window of the particular agent.

The expression **auto-showextremum** concerns the representation of the agent's result. When this expression is specified, the result window of the agent lists the minimum and the maximum value of each result in addition to the different evaluation results.

> **Hint:** By default the results are represented without the explicit minimum and maximum values.

The lock/hide parameters (see p. 225) **lock-extremum** and **hide-extremum** control the output of the minimum and maximum values.

When the expression **auto-buildsum** is specified, the result window of the agent shows - in addition to the single evaluation results - the sum of each result.

> **Hint:** If this expression is not specified, the results are by default represented without the sums.

The lock/hide parameters (see p. 225) **lock-buildsum** and **hide-buildsum** control the output of the sums.

By specifying the expression **auto-group** the results produced by the agent are arranged in groups by default (e.g. there are results per "performer" which relate to results per "organisational unit"). In addition to this key-word the expressions **grouping-class** and **grouping-relation** should also be specified.

The expression **grouping-class** defines which class (*ClassName*) is to be used by default for grouping this agent's results.

Example:

For the results to be put out per "organisational unit" the following has to be defined:

```
grouping-class: "Organizational unit"
```

The expression **grouping-relation** defines which relation (*RelationName*) is to be used by default for grouping this agent's results.

> **Hint:** If you also specified a class (*ClassName*) using the expression **grouping-class**, the relation specified must be a relation to or from this class.

Example:

You want the results of this agent to be grouped according to "roles" by default. Thus, you insert one of the following expressions into the agent definition:

```
auto-group grouping-class: "Role"
grouping-relation: "Has role"
```

Example:

You want the results of this agent **not** to be grouped by default. However, should the ADONIS user later explicitly select a grouped representation, this is to be by default for the managers of organisational units. Therefore, you insert the following expressions into the agent definition:

```
grouping-class: "Organizational unit"
grouping-relation: "Is manager"
```

lock-group and **hide-group** control the grouped output of results.

The expression **calendar** specifies the observation period of the agent. The calendar description (*CalendarString*) has to meet the requirements of the usual calendar syntax.

> **Hint:** If no calendar description is specified, an observation period of 24 hours a day and 365 days a year will be assumed.

> **Hint:** The observation period of the agent is only of importance in the simulation algorithms "Workload Analysis (steady state)" and "Workload Analysis (fixed time period)".

lock-calendar and **hide-calendar** control the input of the agent's observation period.

By use of the expression **objects** the objects to be observed (Activities, Performers and so on) can be defined. The *AQLTerm* has to meet the requirements of the usual AQL syntax.

lock-objects and **hide-objects** control the objects to be observed.

The expression **infotext** permits you to enter an information (*Text*) on the respective agent. The length of the text is restricted to 3700 symbols.

Example:

```
AGENT "Agent for cycletime and costs"
 infotext: "This agent calculates the average cycletime for the
complete process. The average costs are split according to activities."
```

The lock parameter (see p. 225) **lock-infotext** controls the input of the information text.

> **Hint:** There is no **hide** parameter for the information text.

When the expression **deactivate** is specified, newly created agents of this agent type are by default deactivated.

By the expression **allowed-modeltype-***nr* you can restrict the agent's application area to certain model types (*ModelTypeName*). In any case, the agents can only be created for model types which can be simulated.

> **Hint:** *nr* represents a serial number, i.e. for the first model type **allowed-modeltype-1** has to be specified, for the second model type **allowed-modeltype-2** and so on.

> **Hint:** Some result functions exclude certain model types in advance. Agents which contain the result function "Sum over Attributes" cannot be created in working environment models. Even the expression **allowed-modeltype-***nr* may not skip these restrictions.

<u>**Example:**</u>

```
 allowed-modeltype-1:"Business Process Model"
 allowed-modeltype-2:"Working Environment model"
```

In addition, the results each agent has to calculate can be specified. The following types of results are available:

- **Animation - ANIMATION** (see p. 228)
- **Costs of execution - WWECOST** (see p. 229)
- **Costs of execution (conditional) - UWWECOST** (see p. 230)
- **Workload - WORKLOAD** (see p. 230)
- **Capacity - STRAIN** (see p. 231)
- **Cycle time - CYCLETIME** (see p. 231)
- **Frequency - FREQUENCY** (see p. 232)
- **Special formula - FORMULA** (see p. 232)
- **Sum over attributes - AVGSUM** (see p. 233)
- **Number of changes in responsibility - RESPCHANGE** (see p. 234)

Animation

> **Hint:** The result function "Animation" does <u>not</u> restrict the agent with regard to model type.

The result function "Animation" has to be introduced by the expression **ANIMATION**. The result function is then named (*ResultName*) by **name**.

> **ATTENTION:** The names of results functions must be non-ambiguous within one agent.

<u>**Example:**</u>

```
 ANIMATION
 name: "Animation of the cost development"
```

The lock/hide parameters (see p. 225) **lock-name** and **hide-name** control the naming of results.

By specifying the simulation algorithms for the result function you can define during which algorithms the animation is executed. You may of course enter several simulation algorithms following each other.

> **Hint:** When no algorithm is specified, the animation is by default executed during <u>all</u> simulation algorithms.

- **path-analysis** indicates the path analysis.

- **volume-analysis** indicates the capacity analysis.

- **wlstatic-analysis** indicates the workload analysis (steady state).

- **wlnonstatic-analysis** indicates the workload analysis (fixed time period).

The hide parameter (see p. 225) **hide-algorithms** controls the application concerning the algorithms.

> **Hint:** There is no **lock-** parameter for the simulation algorithms.

The lock/hide parameters (see p. 225) **lock-animated** and **hide-animated** control the changes of the results to be animated.

The results to be animated are defined by specifying the Referenced results (see p. 234).

WWECOST - Costs of execution

> **Hint:** The result function "Costs of execution" does <u>not</u> restrict the agent with regard to the model type.

The result function "Costs of execution" must be introduced by the expression **WWECOST**. This result function can be used for calculating personnel costs, resource costs and others.

The **basic result settings** (see p. 235) of this result function can be defined apart from the following exception:

Specified simulation algorithms are ignored as far as this result function is concerned, since the result "Costs of execution" can only be calculated in the capacity analysis.

By the expression **class** you can define the class (*ClassName*) for which the execution costs are to be calculated for. The classes available for definition are all those derived from the meta-model classes "Performer" and "Resource".

> **Hint:** If this expression is not specified, the ADONIS user himself has to select the class.

<u>Example:</u>

```
class: "Performers"
```

The lock/hide parameters (see p. 225) **lock-class** and **hide-class** control the selection of the class specified.

By the expression **attribute** you can define an attribute (*AttributeName*) of the class defined in **class** which represents the hourly wages.

> **Hint:** If this expression is not specified, the ADONIS user himself has to select the attribute.

The lock/hide parameters (see p. 225) **lock-attribute** and **hide-attribute** control the selection of the attribute specified.

UWWECOST - Costs of execution (conditional)

> **Hint:** The result function "Costs of execution (conditional)" does <u>not</u> restrict the agent with regard to the model type.

The result function "Costs of execution (conditional)" must be introduced by the expression **UWWECOST**. This result function can be used for calculating personnel costs, resource costs and others.

The **basic result settings** (see p. 235) of this result function can be defined apart from the following exception:

The Path Analysis must not be used as simulation algorithm for this result function.

By the expression **class** you can define the class (*ClassName*) for which the execution costs are to be calculated for. The classes available for definition are all those derived from the meta-model classes "Performer" and "Resource".

> **Hint:** If this expression is not specified, the ADONIS user himself has to select the class.

<u>**Example:**</u>

```
class: "Performers"
```

The lock/hide parameters (see p. 225) **lock-class** and **hide-class** control the selection of the class specified.

By the expression **attribute** you can define an attribute (*AttributeName*) of the class defined in **class** which represents the hourly wages.

> **Hint:** If this expression is not specified, the ADONIS user himself has to select the attribute.

The lock/hide parameters (see p. 225) **lock-attribute** and **hide-attribute** control the selection of the attribute specified

Workload

> **Hint:** The result function "Workload" does <u>not</u> restrict the agent with regard to the model type.

The result function "Workload" must be introduced by the expression **WORKLOAD**. This result function can then be used to calculate the workload of performers and resources.

The **basic result settings** (see p. 235) of this result function can be defined apart from the following exception:

The path analysis and the capacity analysis must not be specified as simulation algorithms for this result function.

By the expression **class** you can define the class (*ClassName*) for which the Workload Analysis is to be carried out on. The classes available for definition are all those derived from the meta-model classes "Performer" and "Resource".

> **Hint:** If this expression is not specified, the ADONIS user himself has to select the class.

<u>**Example:**</u>

When an agent should calculate the workload of performers, the following expression has to be defined:

```
class: "Performer"
```

The lock/hide parameters (see p. 225) **lock-class** and **hide-class** control the selection of the class specified.

STRAIN - Capacity

> **Hint:** The result function "Capacity" does <u>not</u> restrict the agent with regard to the model type.

The result function "Capacity" must be introduced by the expression **STRAIN**. This result function can then be used both to calculate the capacity of performers and resources and to calculate the personnel and the resource capacities needed. The result function is named (*ResultName*) in **name**.

> **ATTENTION:** The names of result functions must be non-ambiguous within an agent.

The lock /hide parameters (see p. 225) **lock-name** and **hide-name** control the result naming.

The expression **resultscope** specifies to which degree of detail the result is to be calculated. The following types are available:

process   The agent calculates a result in total with regard to the complete business process evaluated. ("Results for whole process" in the Business Process Management Toolkit).

mainobjects   The agent calculates a result each with regard to all objects of the business process examined. For each subprocess called an aggregated result is calculated ("Results for whole process (detailed)" in the Business Process Management Toolkit).

allobjects   The agent calculates a result, each with regard to all objects of the business process examined and with regard to all subprocesses ("Results for process hierarchy" in the Business Process Management Toolkit).

The lock/hide parameters (see p. 225) **lock-resultscope** and **hide-resultscope** control the selection of the degree of detail.

When the expression **dont-show-result** is specified the result calculated is not displayed after the simulation. That makes sense in cases where the result is only a provisional result which is further processed by the result function "Special formula" (see p. 232).

> **Hint:** By default (**dont-show-result** is not defined) it is possible after the simulation to have a look at the result.

The expression **class** defines the class (*ClassName*) for which the capacity is to be calculated for. The classes available are all those derived from the meta-model classes "Performer" and "Resource".

> **Hint:** If this expression is not specified, the ADONIS user himself has to select the class.

Example:

When an agent is to calculate the capacity of performers, the following expression has to be defined:

```
class: "Performers"
```

The lock/hide parameters (see p. 225) **lock-class** and **hide-class** control the selection of the class specified.

CYCLETIME

> **Hint:** The result function "Cycle time" restricts the agent to model types from the business process library.

The result function "Cycle time" must be introduced by the expression **CYCLETIME**. This result function can then be employed to calculate the cycle time of business processes.

The **basic result settings** (see p. 235) for this result function can be made.

FREQUENCY

> **Hint:** The result function "Frequency" restricts the agent to model types from the business process library.

The result function "Frequency" must be introduced by the expression **FREQUENCY**. This result function can then be employed to calculate the frequency of occurrence of business processes.

The **basic result settings** (see p. 235) for this result function can be made:

By specifying the expressions **absolute** or **relative**, you can indicate, whether the absolute or the relative frequency has to be calculated. The <u>absolute frequency</u> tells you during how many simulation runs an object (or model) has been executed. The <u>relative frequency</u> tells you proportionally (as a number between 0 and 1), how often an object (or model) has been executed.

The lock/hide parameters (see p. 225) **lock-mode** and **hide-mode** control the frequency selection.

When you specify the expression **ignore-not-executed-objects** the agent ignores, in its calculation, the frequency of objects which are never executed (i.e. the frequency of which is = 0).

The lock/hide parameters (see p. 225) **lock-ignore** and **hide-ignore** control the settings of **ignore-not-executed-objects**.

FORMULA

> **Hint:** The result function "Special formula" does <u>not</u> restrict the agent with regard to the model type.

The result function "Special formula" must be introduced by the expression **FORMULA**. This result function can then define formulas which can be calculated by the respective agent. The result function is named (*ResultName*) by **name**.

> **ATTENTION:** The names of result functions must be non-ambiguous within an agent.

The lock/hide parameters (see p. 225) **lock-name** and **hide-name** control the naming of the result.

Specifying the simulation algorithm for the result function defines during which algorithms the formula is calculated. You may of course specify several simulation algorithms following each other.

> **Hint:** When no algorithm is specified, the animation is by default executed during <u>all</u> the simulation algorithms.

> **ATTENTION:** The special formula is only calculated in those simulation algorithms in which underlying results are available.

- **path-analysis** indicates the path analysis.

- **volume-analysis** indicates the capacity analysis.

- **wlstatic-analysis** indicates the Workload analysis (static view).

- **wlnonstatic-analysis** indicates the Workload analysis (dynamic view).

The expression **expression** specifies the calculation formula according to which the formula is being calculated.

> **ATTENTION:** Inverted commas (**"**) and back slashes (****) inside the formula must be "masked" by a back slash ****.

Example:

```
expression: "\"Execution Time\" + \"Waiting Time\""
```

A calculation formula always contain one or more result names (under "masked" inverted commas) which are connected by the operators (*BasicOperator*, *AdvancedOperator1* or *AdvancedOperator2*).

All result names used in the formula must correspond to results which have been calculated by the same agent.

> **Hint:** Long calculation formulas can be broken up over several lines.

> **Hint:** If the expression **expression** is not specified, the ADONIS user himself has to specify the calculation formula.

The lock/hide parameters (see p. 225) **lock-expression** and **hide-expression** control the definition of the calculation formula.

By specifying the Referenced Results (see p. 234) the results used in the calculation formula are explicitly defined.

AVGSUM - Sum over attribute

> **Hint:** The result function "Sum over attribute" restricts the agent to model types from the business process library.

The result function "Sum over attribute" must be introduced by the expression **AVGSUM**. This result function calculates the expected value of an arbitrary numerical attribute.

The **basic result settings** (see p. 235) for this result function can be made.

The expression **class** defines the class (*ClassName*) which is to be examined by this result function. The classes available are all those derived from the meta-model class "__Activity__" .

> **Hint:** If this expression is not specified, the ADONIS user himself has to select the class.

Examples:

When an agent is to calculate the average costs of a business process and the attribute "Costs" is defined in the class "Activity", the following expression has to be defined:

```
class: "Activity"
attribute: "Costs"
attribute-type: numeric
```

The lock/hide parameters (see p. 225) **lock-class** and **hide-class** control the selection of the class specified.

With the expression **attribute** an attribute (*AttributeName*) can be defined which will then be summed up.

> **Hint:** if this expression is not specified, the ADONIS user has to select the attribute.

When using the expression **attribute** the attribute type **attribute-type** must also be defined where:

- **numeric** must be specified for attributes of types INTEGER (see p. 173) and DOUBLE (floating number) (see p. 174) and

- **time** must be specified for attributes of type TIME (ADONIS time format) (see p. 175)

Example:

To calculate the average execution time of a business process based on the attribute "Execution time" in the class "Activity", the following expression has to be defined:

```
class: "Activity"
attribute: "Execution time"
attribute-type: time
```

The lock/hide parameters (see p. 225) **lock-attribute** and **hide-attribute** control the selection of the attribute specified.

RESPCHANGE - Changes in responsibility

> **Hint:** The result function "Number of changes in responsibility" restricts the agent to model types from the business process library.

The result function "Number of changes in responsibility" must be introduced by the expression **RESPCHANGE**. This result function calculates how often the performers and/or resources are involved in a business process change during its execution.

The **basic result settings** (see p. 235) for this result function can be made.

The expression **class** defines the class (*ClassName*) which will be examined by this result function. The classes available are all those derived from the meta-model classes "Performer" and "Resource".

> **Hint:** If this expression is not specified, the ADONIS user himself has to select the class

Example:

When an agent is to calculate the changes of responsibility with regard to performers, the following expression has to be defined:

```
class: "Performer"
```

The lock/hide parameters (see p. 225) **lock-class** and **hide-class** control the selection of the class specified.

Referenced result - OBSERVED

The referenced result (*ReferencedResult*) must be introduced by the expression **OBSERVED** and be accomplished by the result (*ResultName*) to be animated (**Animation** (see p. 228)) or by the result (*ResultName*) used in the calculation formula (**Special formula** (see p. 232)), respectively.

Example:

When two results named "Cycletime" and "Costs" should be animated, the animation function has to be defined as follows:

```
OBSERVED "Cycletime"
OBSERVED "Costs"
```

Example:

When two results named "Execution Time" and "Waiting Time" are needed to calculate a specific formula, the function has to be defined as follows:

```
expression: "\"Execution Time\" + \"Waiting Time\""
OBSERVED "Execution Time"
OBSERVED "Waiting Time"
```

Basic result settings of the result functions

For the following result functions the basic result settings (*BasicResultSettings*) listed below can be defined:

- **Costs of execution - WWECOST** (see p. 229)

- **Costs of execution (conditional) - UWWECOST** (see p. 230)

- **Workload - WORKLOAD** (see p. 230)

- **Cycletime - CYCLETIME** (see p. 231)

- **Frequency - FREQUENCY** (see p. 232)

- **Sum over Attribute - AVGSUM** (see p. 233)

- **Number of changes in responsibility - RESPCHANGE** (see p. 234)

The result function (*ResultName*) is named by **name**.

> **ATTENTION:** The names of result functions must be unique within an agent.

The lock/hide parameters (see p. 225) **lock-name** and **hide-name** control the naming of result functions.

Specifying the simulation algorithms for the result function, you define which algorithms are executed to calculate the respective result. You may of course enter several simulation algorithms.

> **Hint:** When no algorithm is specified, the animation is by default executed during <u>all</u> the simulation algorithms.

- **path-analysis** indicates the path analysis.

- **volume-analysis** indicates the Capacity Analysis.

- **wlstatic-analysis** indicates the Workload Analysis (static view).

- **wlnonstatic-analysis** indicates the Workload Analysis (dynamic view).

> **Hint:** Some results may only be calculated by certain simulation algorithms - independent of which algorithms are listed here.

<u>**Example:**</u>

A basic rule is that the result "Costs of execution" can only be calculated by the simulation algorithm "Capacity analysis". If you charge an agent to calculate the "Costs of execution", during Path Analysis, this result will not be calculated.

The hide parameter (see p. 225) **hide-algorithms** controls the selection of the simulation algorithms.

> **Hint:** There are no **lock** parameters for the simulation algorithms.

The expression **resultscope** specifies to which degree of detail the result will be calculated. The following types are available:

process	The agent calculates a result in total with regard to the complete business process evaluated. ("Results for whole process" in the business Business Process Management Toolkit).
mainobjects	The agent calculates a result each with regard to all objects of the business process examined. For each subprocess called an aggregated result is calculated ("Results for whole process (detailed)" in the Business Process Management Toolkit).

allobjects The agent calculates a result, each with regard to all objects of the business process examined and with regard to all subprocesses ("Results for process hierarchy" in the Business Process Management Toolkit).

lock-resultscope and **hide-resultscope** control the selection of the degree of detail.

When the expression **dont-show-result** is specified the calculated result is not displayed after the simulation. This would be used for cases where the result is only an intermediate result which is further processed by the result function "Special formula" .

> **Hint:** By default (**dont-show-result** is not defined) it is possible after the simulation to view the result.

lock-show-result and **hide-show-result** control the setting of **dont-show-result**.

When the expression **history-mem** is specified, the development of the results is temporarily stored in the main memory.

When the expression **history-disk** is specified, the development of results is stored on a data medium such as a floppy disk

> **Hint:** If neither of the two expressions are specified, the development of results is, by default, not stored.

By the expression **history-file** you can specify a file (*FileName*) including its path into which the development of results is to be saved by default. Usually, this expression is used in combination with **history-disk**.

> **ATTENTION:** Back slashes (****) inside the file name must be "masked" by '****'.

> **Hint:** When the expression **history-file** is specified in combination with **history-mem**, usually this expression will be ignored. However, if the ADONIS user switches manually to saving the development of results on a data medium such as a floppy disk, the default file name (*FileName*) will be the one specified.

Example:

The development of the result currently defined shall by default be stored in the main memory during the simulation. Should an ADONIS user later on decide that the development of results should be stored on a floppy disk, for example, this will by default be done in the file "C:\TEMP\AGENT. TXT". For this, the following expression has to be included into the definition of the current result:

```
history-mem history-file: "C:\\TEMP\\AGENT.TXT"
```

Example:

The development of the result currently defined shall - by default - be stored during the simulation into the file "C:\TEMP\AGENT. TXT". For this, the following expression has to be included into the definition of the current result:

```
history-disk history-file: "C:\\TEMP\\AGENT.TXT"
```

lock-history-mem and **hide-history-mem** control the setting of **history-mem**.

lock-history-disk and **hide-history-disk** control the setting of **history-disk**.

When the expression **dont-show-history** is specified, the saved development of results will not be shown after the simulation.

The lock/hide parameters (see p. 225) **lock-show-history** and **hide-show-history** control **dont-show-history**.

Example of the agent definition language

The following example shows a complete agent definition, based on the ADONIS standard application library. The example contains a single agent which calculates four different results during the simulation.

If you wish (and if you work with the ADONIS standard application library), you can just copy this agent definition into the attribute "Agent definition". In this way, you will have your first predefined agent which you can then employ in the ADONIS Business Process Management Toolkit.

After the agent definition, every expression used in the example is explained. The reasons why certain expressions have **not** been used are also explained.

```
AGENT "ExampleAgent"
volume-analysis
lock-select-results
hide-extremum
auto-buildsum
hide-buildsum
infotext: "This agent is an ExampleAgent."
modeltype-1: "businessprocessmodel"
FORMULA
name: "Totalcosts"
expression: " \"PersonnelCosts\" + \"ResourceCosts\" + \"ProcessCosts\"
"
lock-expression
OBSERVED "PersonnelCosts"
OBSERVED "ResourceCosts"
OBSERVED "ProcessCosts"
WWECOST
name: "PersonnelCosts"
lock-name
resultscope: process
hide-history-mem
hide-history-disk
class: "Performer"
lock-class
attribute: "Hourly wages"
lock-attribute
WWECOST
name: "ResourceCosts"
lock-name
resultscope: process
hide-history-mem
hide-history-disk
class: "Resource"
lock-class
attribute: "Hourly wages"
lock-attribute
AVGSUM
name: "ProcessCosts"
lock-name
resultscope: process
hide-history-mem
hide-history-disk
class: "Activity"
lock-class
```

```
attribute: "Costs"
attribute-type: numeric
lock-attribute
```

In the following all the expressions used in the example above are explained in the order of their oc-currence:

AGENT "ExampleAgent"

An agent definition is always introduced by the expression **AGENT**. This expression must therefore be in the first position. In addition, a name must be assigned to the agent type defined. The name can be chosen according to your preferences. In the example above the name chosen is "ExampleAgent".

volume-analysis

This expression specifies that this agent is only active during the simulation algorithm "Capacity Analysis".

As defined below in the agent definition, this agent contains among others the result function "Costs of execution" (see expression **WWECOST**). This result can only be calculated during a capacity analysis. If the agent is employed in any other simulation algorithm, it will not be able to calculate this result function.

The result of the result function "Special formula" (see expression **FORMULA**) is based on the "Costs of execution" and thus could also not be calculated within a path or workload simulation. Only the re-sult function "Sum over attribute" (see expression **AVGSUM**) could be calculated in this case.

The agent's area of application is restricted to the capacity analysis by the expression **volume-analysis**. This prevents the agent from being employed in simulation algorithms where its functionality could not be fully used.

lock-select-results

This expression specifies that the ADONIS user can neither remove result functions from nor add re-sult functions to the agent.

In the case of predefined agents it is in general sensible to specify this expression, since the prede-fined agent usually contains all the result functions desired. If an ADONIS user wants an agent with other result functions, he should create a new agent and configure it according to his wishes rather than carry out extensive changes to an existing predefined agent.

However, in the example above the expression **lock-select-results** has been specified for a further reason. This agent contains the result function "Special formula" (see expression **FORMULA**). This is based on other result functions of this agent. If ADONIS users were allowed to remove result functions from this agent, they might accidentally remove result functions which are needed to calculate the "Special formula". To prevent this from happening, the expression **lock-select-results** is specified.

hide-extremum

In addition to an agent's results, the minimum and maximum values of the results can also be dis-played, if desired. The expression **hide-extremum** specifies that the control elements which control the display of the minimum and maximum values are hidden from the ADONIS user.

The expression **auto-showextremum** causes the extreme values to be shown by default. In our ex-ample, extreme values are - by default - not shown.

auto-buildsum

This expression specifies that for every result of this agent the total sum is also shown by default. However, only a default value is specified here which can be changed by the ADONIS user at any time. To prevent the ADONIS user from changing the default settings the following expression can be specified:

hide-buildsum

This expression causes the control elements controlling the display of the sums to be hidden from the ADONIS user. So the ADONIS user no longer has the possibility to change the default value specified above.

`infotext: "This agent is an ExampleAgent."`

An information text is specified here. We recommend that you enter a short but meaningful information text for each predefined agent as this helps the ADONIS user to select the agent he desires - especially useful when there are many predefined agents.

`modeltype-1: "Businessprocessmodel"`

This expression ensure that this type of agent may only be created in business process models.

The specification of model type is not necessarily required in this example. The agent defined contains the result function "Sum over attribute (see below; expression "AVGSum"), which can only be assigned to agents in business process models. This result function already restricts this agent to business process models. Nevertheless to give a complete picture, this expression has been specified in this example.

`FORMULA`

This expression assigns the "Special formula" result function to the agent. All the following expressions (up to the definition of the next result function) refer to this result function.

`name: "TotalCosts"`This expression names the "Special formula" result function. The name can be freely chosen but must be unique within the agent.

`expression: " \"PersonnelCosts\" + \"ResourceCosts\" + \"ProcessCosts\" "`

This expression defines the calculation formula for the result. Since the calculation formula itself has to contain inverted commas, all other inverted commas inside the calculation formula are *masked*, i.e. instead of **`"`** **`\"`** has to be written.

> **Hint:** The elements "PersonnelCosts", "ResourceCosts" and "ProcessCosts" are the names of the other result functions in this agent (see below).

`lock-expression`

This expression prevents the ADONIS user from changing the calculation formula on his own. However, he may still view the calculation formula.

> **Hint:** The expression **hide-expression** would *in addition* hide the calculation formula from the ADONIS user. However in order to understand a certain result it can be useful when the corresponding calculation formula is displayed.

When the ADONIS user may not change the calculation formula, it might also be important, to prevent him from removing result functions from this agent. Otherwise, the calculation formula could refer to results which this agent cannot calculate any longer. The ADONIS user would have no possibility to adjust the calculation formula accordingly. For this reason, the expression **lock-select-results** (see above) has been included in this agent's definition.

`OBSERVED "PersonnelCosts"`
`OBSERVED "ResourceCosts"`
`OBSERVED "ProcessCosts"`

These expressions specify all those result functions on which the "Special formula" is based. It is in any case required that the names of all result types which occur in the calculation formula are specified.

`WWECOST`

This expression assigns the result function "Costs of execution" to the agent. All the following expressions (up to the definition of the next result function) refer to this result function.

`name: "PersonnelCosts"`

The name of this result. The name can be chosen freely, but must be unique within the agent.

In this example the result function "Special formula" (see expression **FORMULA**) refers to the name of this result. If the result's name is changed, the definition of the formula must be adjusted accordingly.

lock-name

This expression specifies that the ADONIS user may not change the result name of this result function.

This is necessary because the special formula result function (see expression **FORMULA**) refers to the name of this result function ("PersonnelCosts"). If the ADONIS user changed the name of the result, the calculation formula of the "Formula" would have to be adjusted accordingly. In the definition of the "Formula", however, it has been specified that the ADONIS user may not change the calculation formula (see above; expression **lock-expression**). Thus, he cannot adjust this formula. Therefore, it is necessary to specify that he also cannot change the current name of the result ("PersonnelCosts").

resultscope: process

This expression specifies that the result function calculates a single number by default as result with regard to the whole process evaluated.

> **Hint:** Since neither the expression **lock-resultscope** nor the expression **hide-resultscope** were specified, the ADONIS user may, if necessary, change the degree of detail of the result calculation.

hide-history-mem
hide-history-disk

These expressions cause all control elements in the configuration window of the result function to be hidden. Therefore the ADONIS user has no access to these elements. Since the default setting is that the development of results is not stored (and nothing to the contrary has been specified in this ExampleAgent definition), the development of results for this specific result will not be recorded.

class: "Performer"

This expression causes the execution costs to be calculated with regard to the class "Performer". This means, the personnel costs will be calculated.

lock-class

This expression prevents the ADONIS user from changing the class selection. Since the expression **lock-class** but not the expression **hide-class** has been specified, the ADONIS user still has the possibility to see for which class the execution costs are being calculated.

attribute: "Hourly wages"

This expression specifies the attribute which contains the hourly wages based on which the execution costs are to be calculated. The attribute "Hourly wages" is defined in the class "Performer" in the ADONIS standard application library.

lock-attribute

This expression prevents the ADONIS user from changing the attribute selected. Since the expression **lock-attribute** but not **hide-attribute** has been specified, the ADONIS user may still see on which attribute the calculation is based.

WWECOST

This expression assigns the result function "Costs of execution" a second time to the agent. This time, however, the result function is configured differently so that this time the "ResourceCosts" can be calculated (see below).

> **Hint:** All the following expressions (up to the definition of the next result function) refer to this result function.

name: "ResourceCosts"

Definition of a further result name.

lock-name

The explanations made under "PersonnelCosts" (see above) also apply here.

resultscope: process

The explanations made under "PersonnelCosts" (see above) also apply here.

hide-history-mem
hide-history-disk

The explanations made under "PersonnelCosts" (see above) also apply here.

class: "Resource"

This expression determines that the execution costs are calculated with regard to the class "Resource" . Thus it is the resource costs which are calculated here (which is why the result has been named that way).

lock-class

The explanations made under "PersonnelCosts" (see above) also apply here.

attribute: "Hourly wages"

This expression specifies the attribute containing the hourly wages based on which the resource costs are calculated. The attribute "Hourly wages" is defined in the class "Resource" of the ADONIS standard application library.

lock-attribute

The explanations made under "PersonnelCosts" (see above) also apply here.

AVGSUM

This expression assigns the result function "Sum over Attributes" to the agent. All the following expressions now refer to this result function.

name: "ProcessCosts"

The explanations made under "PersonnelCosts" (see above) also apply here.

lock-name

The explanations made under "PersonnelCosts" (see above) also apply here.

resultscope: process

The explanations made under "PersonnelCosts" (see above) also apply here.

hide-history-mem
hide-history-disk

The explanations made under "PersonnelCosts" (see above) also apply here.

class: "Activity

This expression defines in which class the attribute to be summed up is defined. It must be a class that can be instantiated and which has been derived from the meta-model activity class. In the ADONIS standard application library this is the class "Activity".

lock-class

This expression prevents the ADONIS user from changing the class selection.

attribute: "Costs" attribute-type: numeric

This expression specifies that the expected value of the sum of the attribute "Costs" is to be calculated. Since this attribute is numerical, the expression **numeric** must be specified for "**attribute-type:**".

```
lock-attribute
```
This expression prevents the ADONIS user from changing the attribute selection.

2.1.3.22   Enterprise Time - Tage pro Jahr (Simulation)

An attribute where the basic settings for the working days per year can be specified.

The attribute value "Tage pro Jahr" is used in combination with the attribute value "Stunden pro Tag" (see chap. 2.1.3.23, p. 242) in the Simulation Component to convert the times into enterprise times (see user manual, chapter "Simulation"). This library attribute holds the default value.

   Hint: The attribute "Tage pro Jahr" is only defined for BP libraries.

2.1.3.23   Enterprise Time - Stunden pro Tag (Simulation)

An attribute where the basic settings of the hours per working day can be defined.

The attribute value "Stunden pro Tag" is used together with the attribute value "Tage pro Jahr" (see chap. 2.1.3.22, p. 242) in the Simulation Component to convert the times into enterprise times (see user manual, chapter "Simulation"). This library attribute contains the default value.

   Hint: The attribute "Stunden pro Tag" is only defined for BP libraries.

2.1.3.24   Activity based costing - CCC-Mapping (Evaluation)

The classes, relations and attributes of a user-defined application which are necessary for the Process Cost Analysis component are specified in the attribute "CCC-Mapping".

   Hint: The attribute "CCC-Mapping" is only defined for BP libraries.

The language in which the default settings of ADONIS-3c are specified, is based on the following syntax

<u>***CCC-Mapping :***</u>   *CcClass Synonyms* .

***CcClass* :**   **CCCLASS**
            costcenter: "*Name of CostCenterClass***"**
            relcount: "*Name Of IsChargedTo-RelationClass***"**
            relchef: "*Name of IsCostCenterManager-RelationClass***"** .

***Synonyms* :**   **SYNONYMS**
            budget: "*CostCenterBudgetName***"**
            lmntime: "*OutputNeutralTimeName"***
            lmnproc: "*OutputNeutralProcessName***"**
            lmnfix: "*OutputNeutralFixCostName***"**
            ccmanager: "*CostCenterManagerName***"**
            cccap: "*CostCenterCapacity***"**
            lmiproc: "*OutputInducedProcessName***"**
            costdriver: "*CostDriverName***"**
            cdquantity: "*CostDriverQuantityName***"**
            quantity: "*QuantityName***"**
            idlecap: "*IdleCapacityName***"**
            executioncost: "*ExecutionCostName***"**
            stuffcost: "*StuffCostName***"**
            totalfixcost: "*TotalFixCostName***"**

```
          lmncost:   "OutputNeutralCostName"
          idlecost:  "IdleCostName"
          totalcost: "TotalCostName"
          lmipcs:    "OutputInducedProcessCostRatioName"
          pcs:       "ProcessCostRatioName" .
```

The CCC mapping is introduced by the key-word **CCCLASS**. The name of the class (*Name of Cost-CenterClass*) which is defined in the library as the cost centre class is specified in the expression **costcenter**.

The name of the relation (*Name Of IsChargedTo-RelationClass*) which defines the relation "Is charged to" (from performer to cost centre) is specified in the expression **relcount**.

The name of the relation (*Name of IsCostCenterManager-RelationClass*) which defines the relation "Is cost centre manager" (from performer to cost centre) is specified in the expression **relchef**.

The **SYNONYMS** element adapts the process costing terms used in ADONIS-3c to those used in your particular company.

2.1.3.25   Activity based costing - CCC-Grundeinstellung (Evaluation)

The classes, relations and attributes of a user-defined application library which are necessary for the process costs analysis are specified in the attribute "CCC-Grundeinstellung".

> **Hint:** The attribute "CCC-Grundeinstellung" is only defined for BP libraries.

The language in which the default settings of ADONIS 3c are specified, is based on the following syntax:

<u>***CCC Basic/Default Setting :***</u>   `Currency { FixCost } .`

***Currency* :**   `CURRENCY "Currency" .`

***FixCost* :**   `FIXCOST "FixCostName" .`

In the **CURRENCY** element the currency and - if necessary - a measuring value (*Currency*) is defined. The text entered is displayed in the result representation of the process costs analysis. The default value is "Euro".

> **Hint:** The specification of the currency should not exceed 20 symbols, since that is the maximum number of symbols that will be shown.

Example:

If the sums calculated by the process costs analysis are to be represented in 1000 Euro, the following has to be specified:

```
CURRENCY "TEUR"
```

or

```
CURRENCY "1000 EUR"
```

In the **FIXCOST** element the names (*FixCostName*) of those attributes which are defined in the activity classes of the library as costs of the type "Activity fixed Costs" are specified. Through this, the costs are evaluated properly.

2.1.3.26   Dynamische Evaluationsmodule (Evaluation)

The attribute "Dynamische Evaluationsmodule" contains the information which menu options should be seen in the ADONIS Evaluation Component. Together with each menu option its calculation instructions for the evaluation module are set.

The language for the dynamic evaluation component is based on the following syntax:

<u>Dynamic Evaluation Modules:</u> `EVALUATIONMODULE "`*Modulename*`"`
                 `simoption: "`*SimOptionName*`"`
                    `[ `*VolumeReportMatrixDefinition*` ]`
                    `{ `*Resultdefinition*` } .`

VolumeReportMatrixDefinition : `vmatrix-class: "`*MatrixClass*`"`
                    `[ vmatrix-apref: "`*AttributProfileReference attribute*`" ]`
                    `vmatrix-recordattr: "`*MatrixAttribute*`"`
                    `vmatrix-perfattr: "`*PerformerAttribute*`"`
                    `vmatrix-volumeattr: "`*VolumeAttribute*`"`
                    `[ vmatrix-factorattr: "`*FactorAttribute*`" ] .`

Resultdefinition : `CALC "`*Resultname*`" | REDEF "`*Function*`"`
                    `[ per: run | period ]`
                    `[ `*Visibility*` ]`
                    `[ BP `*StructureSpec*` ]`
                    `[ ACT `*StructureSpec*` ]`
                    `[ WE `*StructureSpec*` ]`
                    `[ PER `*StructureSpec*` ] .`

StructureSpec : *Formula* | *Aggregation*
                    `[ subcalc: main2sub | sub2main | sub ]`
                    `[ `*AQL constraint*` ]`
                    `[ `*Visibility*` ] .`

Formula : `formula:(`*Expression*`) as: time | numeric |`
                 `addfrm: (`*Expression*`) as: time | numeric .`

Expression : A calculation command that has to be based on the EXPRESSION attribute language syntax (see chap. 10., p. 561).

In addition to the EXPRESSION attribute language, the following special functions are available: **aval(...)**, **maval(...)**, **fre()**, **vol()**, **rvol()**, **cyc()**, **hpd()**, **dpy()**, **value("**Resultname**")**.

Aggregation : `recalc | add | wgt | no .`

AQL constraint : `constraint:"`*AQL expression*`" .`

Visibility : `show:off | standard | always .`

The key word **EVALUATIONMODULE** is followed by the name of the dynamic evaluation module. This name is shown as a menu option in the ADONIS evaluation component.

The key word **simoption** gives a combination of the input parameters, which should be used during calculation of the cycle time. This combination of the input parameters is not important for other results.

> **Hint: simpotion** must be defined in the library attribute "**Simmapping**" (see chap. 2.1.3.18, p. 218).

The key word **vmatrix-class** shows, in which class the quantity matrix is defined. The quantity matrix is represented by an attribute of type RECORD (key word **vmatrix-recordattr**) and consists of at least 2 columns: One column is for the organisational unit or performer who reports quantity (key word **vmatrix-perfattr**), and the second one for the reported quantity (key word **vmatrix-volumeattr**). Each model can get from 0 to n objects of the class.

The key word **vmatrix-apref** is optional. If the quantity matrix is closed in an attribute, the key word gives an attribute profile reference to the attribute name.

The key word **vmatrix-factorattr** is optional. There are names of the attribute used to fix the reported quantities. All the reported quantities will be multiplied together with its value before analysis.

The key word **CALC** starts the definition of new evaluation results. If one of the existing function (fre(), vol(), rvol(), cyc(), hpd() or dpy()) should be redefined with the solution definition, the key word **REDEF** should be used instead of CALC.

Each result is by default analysed both per one flow and per defined period of time (e. g. per month). Giving **per:run** means that the result will concern only process flow. Giving **per:period** means that the result will concern only the defined period of time.

The key word **BP** defines the starting point of results definition for **object groups** in Business Process Models.

> **Hint:** A *group* is everything which can be seen as aggregation of other instances (a model, an organisational unit ...).

The key word **ACT** defines the starting point of results definition for activities.

The key word **WE** defines the starting point of results definition for **object groups** in WE models.

The key word **PER** defines the starting point of results definition for a performer.

> **Hint:** If the key word BP, ACT, WE or PER is not given, the actual results for current model structures will not be calculated.

The key word **formula** comes from the calculation formula of current results. If, instead of the formula, the key word **addfrm** is given, the calculation formula will be used for all objects directly connected with current model structure. The sum of the results builds up the real result of the current model structure.

> **Hint:** The following structures are seen as "directly connected": Activities are "directly connected" with objects groups in Business Process Models. Performers are "directly connected" with activities. Performers are also "directly connected" with objects groups in WE models. Activities are "directly connected" with performers.

The data type of results must be given with **as:time** or **as:numeric**.

The result function **fre()** gives the frequency of appearance of activities and Business Process Models. This function can only be used together with the key words **BP** and **ACT**.

The result function **vol()** gives the appearance number of the current Business Process Models. This function can only be used together with the key words BP and ACT. Please notice that the appearance number for an activity is **fre() * vol()**!

The result function **rvol()** gives the quantity reported by a performer. If this function is called together with the key word ACT or BP, the system will inform you about the total reported quantity for each structure.

The result function **cyc()** gives the cycle time of a Business Process Model. This function can only be used together with the key word BP.

The result function **hpd()** gives working hours per day.

The result function **dpy()** gives working days per year.

The result function **value("***Resultname***")** gives the result value together with the passed name. The result must be defined for the current dynamic evaluation module.

The key word **recalc** results in the same calculation formula for all current structures as for the "directly connected" structure.

The key word **add** gives the situation, where the result for the current structure is the same as the sum of results of all "directly connected" structures.

The key word **wgt** gives the situation, where the result for the current structure is the average of the result divided by all "directly connected" structures.

The key word **no** results in no result calculated for the current structure.

Thanks to the key word **subcalc** the calculation of results are influenced, especially for the structures, which are subordinated to other structures. (For example, activities are normally subordinated by their business process models;.) The parameter **main2sub** allows the user to use the same formula both for "subordinated" structures and superordinated structures. However, the result will be adopted to a subordinated structure. (It is a standard behaviour.) The parameter **sub2main** allows the user to calculate the "subordinated" structure due to its formula, but the result will be adopted to the superordinated structure. The parameter **sub** allows the user to calculate the "subordinated" structure due to its formula and the result will not be adopted for other structures.

The key word **constraint** allows the user to calculate the result only for the chosen objects.

The key word **show** defines under which conditions the results could be displayed. The parameter **off** deactivates the complete presentation of results. Using the parameter **standard** means that the result will not be shown, when the subordinated structure is subordinated in its presentation by an activity or a performer. The parameter **always** allows the result to be always presented.

2.1.3.27   Dokumentations-Konfiguration (Dokumentation)

This attribute defines the settings for documentation purposes and consists of the following parts:

-
-
-

Defining attribute modes

An attribute mode quotes which attributes of the modelling objects will be taken over into the documentation. After the key word **ATTRIBUTEMODI** follows a list of the available attribute modes.

> **Hint:** The attribute modes of an application library can not be changed online. If you need a new mode or changes to an existing one please contact your ADONIS consultant.

Example:

```
ATTRIBUTEMODI  "@Standard@@StdDocu@@Documentation@@Docu@@Documentation  with  simula-
tiondata@@SimDocu@"
```

In the example there are three attribute modes with the name "standard", "documentation" "documentation with simulation data" and the included keys "StdDocu", "Docu", "SimDocu" are defined.

Defining the settings dialogue

In the settings dialogue settings can be made which have effects on the generated documentation (e.g. should graphics be generated). The key word **DIALOG** is used to define which settings are available for the user.

> **Hint:** You can group or remove the existing settings from the settings dialogue. New settings can not be added online. If you want to add new settings ask your ADONIS consultant.

The settings dialogue will be determined via the notebook definition "AttrRep" (see chap. 2.1.1.2, p. 171):

Example:

```
DIALOG
notebook:"NOTEBOOK
CHAPTER \"General settings\"
ATTR \"Library specific settings\" ctrltype:check
GROUP \"General settings\"
ATTR \"Modus\" ctrltype:dropdown
ATTR \"Attribute modus\" ctrltype:dropdown
ATTR \"Generate graphics\" ctrltype:check
ATTR \"Orientation\" ctrltype:dropdown
ATTR \"Modus for graphic files\" ctrltype:dropdown
ATTR \"Page layout\" ctrltype:dropdown
ENDGROUP
..."
```

A **CHAPTER element** defines a new chapter in the settings dialogue. The chapter headline will be quoted after the keyword CHAPTER.

An **ATTR element** defines an insertion field for a setting possibility (e.g. group graphics, modus, page layout) in the according chapter of the setting dialogue. The name of the insertion field will be quoted after the key word ATTR.

The two elements **GROUP** and **ENDGROUP** group more settings. The setting dialogue will be framed around the insertion fields which are situated between the two elements. The frame gets a headline which is quoted after the key word GROUP.

> **ATTENTION:** As the notebook definition is situated between quotation marks of the elements **DIALOG notebook:**all quotation marks (**"**) for the naming of chapter groups or insertion fields are masked i.e. replaced through **\"**.

If you want to prevent that a user can make determined settings himself delete the according ATTR element from the notebook definition.

Defining the menu entries

For every menu item you can do settings and reference to settings from other setting dialogues. From the setting dialogue referenced settings will be quoted through the key word **attribute:** followed by the name of the setting in the setting dialogue.

> **Hint:** Which of the following settings are available depends on the application library.

You can assign to all attributes which have been assigned to the settings dialogue also fixed settings. Therefore you can make fixed settings and forbid the user to make changes.

To assign attributes to fixed settings change the according setting and remove the insertion from the notebook definition. If you like to e.g. generate only graphics with the orientation "steady", remove the setting

```
gfxorientation:attribute:"orientation"
```

through the setting

```
gfxorientation:"steady"
```

Now the graphics will be generated with the characteristic "steady", independently of the definition given by the user in the setting dialogue.

Then delete the line

```
ATTR \"orientation\"
```

from the notebook definition to remove the insertion field for the orientation from the setting dialogue.

Each menu insertion in the menu "documentation" represents a possibility for the user to generate the documentation. With the keyword **EXPORT** such a menu item will be created.

Example:

```
EXPORT "HTML - Generation"
   visible:1
   smarticon:html
   menuname:"~HTML Generation"
   filename:attribute:"filename"
   filedescription:"HTML files"
   fileextension:"*.htm"
   copy1:"boclogo.bmp"
   copy2:"db:\\icon.gif"
   requirefile1:"%Hboclogo.gif"
   temp1:"sgmlfilename"
```

In the element **EXPORT** the window title for the model selection for the documentation generation will be determined. The following attributes will be defined for this element:

Is the attribute **visible:** assigned to the value 1, the menu "documentation" in the import/export component will be displayed in the insertion through which the export will be started. If the value 0 is assigned to the attribute, no insertion will be added to the menu.

The name, which will be displayed in the "documentation" (import/export component) will be defined in the attribute **menuname:**.

The attribute **smart icon:html** and the **smart icon:rtf** are allowed to be defined in one EXPORT element of the documentation configuration. Over them the menu insertion will be determined, which will be called over the SmartIcons 🗐 (smarticon:html) and 🗐 (smarticon:rtf).

File type and file extension will be determined in **filedescription:** and **fileextension:** respectively.

The filename of the target file will be set in the **filename:**. This attribute will be set with the file name, which the user inserted in the export dialogue.

The attribute **temp*Number*:** will be set at the running time with the file name of a temporary file. Further temporary files can be generated through continuous counting.

The attribute **copy*Number*:** allows to quote the names of graphic files (*fileName*) that are included in the documentation. *Number* stands for a number to be entered (starting from one).

The attribute **requirefile** allows to attach the selected external files to the generated documentation. Before starting the documentation generation, the system checks whether these attached files exist or not. If there are files that are missing, an error message will be shown and the documentation generation will be cancelled. The key word **%H** will be replaced with the installation directory of ADONIS.

Hint: The file name (incl. path) must be quoted in inverted commas. The back slashes ("\") in the path quotation of the file must be masked through '\' (e.g. "c:\\gfx\\logo.gif" for the file "logo.gif" in the directory "gfx" on the "C:" drive).

Hint: In order to include files, which are saved asexternal files in the ADONIS database (see chap. 15., p. 606),please quote as the path '**db:**' (e.g. "db:\\logo.gif" for the in the database saved file "logo.gif").

Settings for the SGML model export are in the element **SOURCE "Model2SGML"**. The following attributes are defined:

Example:

```
SOURCE "Model2SGML"
   filename:attribute:"sgmlfilename"
   basename:attribute:"filename"
   sortmode:"deep search"
   modeltypes:"Business Process Model;process landscape"
   copydocuments:"docs\\"
   libraryspecific:0
```

The name for the SGML file will be quoted in the attribute **filename:**. The attribute will be saved with the name of the temporary file.

The file name for the target documentation file will be set in **basename:**. The attribute will be saved with the name given by the user.

The attribute **subprocesses:** contains information, whether the referenced models should also be exported. The values 1 or 0 will be assigned to the attribute, accordingly to the user decision (1= yes, 0=no).

The order of models contained in the generated documentation can be defined in **sortmode:**. Possible values are "deep search" "alphabetical" and "width search".

The names of the documentation attributes can be translated by the settings of **translation:**. Therefore, the user should assign to the settings a chain of values in the following form: @*Attribute name*@ @*Translation*@. *Attribute name* is a name of an attribute in ToolName;. *Translation* contains the translated name of the attribute, which should be displayed instead of the attribute name from ToolName;.

modeltypes: permits the user to limit the group of models to be contained in the generated documentation to the defined model types. Please enter all the selected model types that should be contained in the documentation using as a separator ";".

copydocuments: allows the user to give a name of an absolute or relative catalogue, where all the external files should be copied into during documentation generation.

Hint: The external files will be copied during documentation generation, provided they are system files. In case an attribute "Programcall" (PROGRAMCALL) references to, for example, an Internet address, the file will not be copied (but link to the web page will be taken over to the generated documentation). If the files cannot be copied (e.g. in case of Internet address, where the file no longer exists), the window with adequate information will display (during documentation generation).

Switching on **checkexternfilenames:** means, that all referenced external files wil be carefully checked (value 1). The user will read warnings during documentation generation each time when a name of the external file contains things other than numbers and lower-case letters.

The attribute **libraryspecific:** can get the value 0 or 1. The value 1 means that the user has chosen (in the window with settings) "Settings specific for model types", on the other hand the value 0 means that the user has not chosen it.

The element SOURCE "Model2SGML" needs definitions for **LIBRARY** element of each model type as well as for a general LIBRARY element. LIBRARY elements contain attributes, which influences documentation generation (its content and way of presentation).

Selecting "Library specific settings" (in the settings dialogue window) means, that during documentation generation each model will be generated according to the settings defined in the LIBRARY element of the respective model type. If the option is not active, all the models are generated according to general settings. The following attributes are defined for each LIBRARY element:

Example:

```
LIBRARY "Business Process Model"
   gfxformat:"bmp\
   gfxdpi:"70"
   notebookattr:"Documentation"
   graphics:"1"
   gfxorientation:"Upright"
   gfxlayout:"Do not split graphic file"
   gfxmode:"unchanged"
   mode:"Standard"
   gfxdozoom:"1"
   gfxzoomlevels:"10;40;100"
```

The following list describes settings for all library elements:

graphics (INTEGER)

has value 1 when graphics should be generated, value 0 if not.

[mode] (ENUMERATION)

The Mode (see chap. 2.1.3.9, p. 191), in which the documentaion should be generated. Classes will be shown and exported according to the mode settings.

[notebookattr] (ENUMERATION)

The attribute modi for an export of a document. The name of the attribute modi must be specified (not an attribute name / key).

[gfxformat] (ENUMERATION)

The file format for an export of a graphic. *gfxformat* must be entered, in case *graphics* has the value 1. Possible formats are bmp1, bmp24, pcx8, pcx24, png, jpg, emf und svg.

[gfxdpi] (DOUBLE)

The dpi (dots per inch) for graphics. *gfxdpi* must be entered when *graphics* has the value 1.

[gfxmode] (ENUMERATION)

The dpi (dots per inch) for graphics. *gfxmode* must be entered, in case *graphics* has the value 1.

[gfxorientation] (ENUMERATION)

Decide on graphic orientation during export. The possibilities are "do not change", "rotate right (clockwise)", "rotate left (counter clockwise)" or "rotate by 180". *gfxorientation* must be entered, in case *graphics* has the value 1.

ATTENTION: Graphics *emf* cannot be rotated!

[gfxlayout] (ENUMERATION)

> The Page layout (see chap. 2.1.3.13, p. 203), which should be used during export. *gfxlayout* must be entered, in case *graphics* has the value 1.

[gfxscale] (INTEGER)

> The zoom factor for graphics. *gfxscale* must be entered, in case *graphics* has value the 1.

[gfxdozoom] (INTEGER)

> In case this setting has value 0, there will be exactly one graphic generated for each model. If *gfxdozoom* equals 1, there will be exactly one graphic generated for each entry in *gfxzoomlevels*.

[imagemaps] (INTEGER)

> The user can decide, whether graphics in HTML documentation should have hyperlinks (value: 1) or not (value: 0).

ATTENTION: The page layout and modes defined here apply only to generated graphics files, which are contained in the process documentation and not in the documentation layout itself.

A post processor will be started in **SOURCE "AdoScript"** to convert the SGML file into a file having the target format. There will also be a DSSSL file entered here, which should be used for documentation generation. The following attributes will be defined for the element:

Example:

```
SOURCE "AdoScript"
   name:"Jade Converter"
   var1:attribute:"tempfilename"
   var2:attribute:"filename"
   var3:attribute:"homedir"
   script:"SYSTEM (\"\" + homedir +
\"\\\\jade.exe -t html -d \\\"\" + homedir + \"\\\\std2htm3.dsl\\\" -o  \\\"\" +
filename + \"\\\" \\\"\" + tempfilename + \"\\\"\")"
```

name: defines the name of a script.

var*Number***:** allows to define AdoScript variables, which will be used in the actual script. *Number* is used to enter the right number starting from one. The first variable in the example defines the name of the temporary file, the second variable defines the name, which was entered by the user in the export dialogue window and the third one defines the installation directory of ADONIS.

script: defines AdoScript, which calls the post processor.

SOURCE "ModelGroups" contains model group structure and model references for the purpose of export.

Example:

```
SOURCE "AdoScript"
   filename:attribute:"tempfilename"
   exportall:1
```

The name of the SGML file is entered in the attribute **filename:**. The attribute contains also the model group structure. The attribute will be created with the filename of the temporary file.

exportall: says, whether all model groups should be exported (value 1), or only those model groups, which have model references to the exported models (value 0).

SOURCE "UserVariable" allows the export of all the selected user settings from the settings dialogue window.

Example:

```
SOURCE \"UserVariable\"
   filename:attribute: \"tempfilename\"
   var1:attribute: \"Header\"
   var2:attribute: \"Footer\"
   var3:attribute: \"Project\"
```

The name of the SGML file is entered in the attribute **filename:**. This file contains user settings. The attribute will be created with the filename of the temporary file.

var*Number*: permits the user to define settings, which should be exported. *Number* is used to enter the right number starting from one.

> **Hint:** The available settings depend on the application library. If you want to define or change current settings, please contact your ADONIS consultant.

Dokumentations-Konfiguration - Grammar

The language of settings concerning documentation configuration bases on the following syntax:

***DocuConfig*:**	*AttrModeDef* { *ExportDef* } *DialogDef*
***AttrModeDef*:**	**ATTRIBUTEMODI "***AttrModeEntry* { *AttrModeEntry* }**"**
AttrModeEntry	**@***attrModeName***@@***attrModeKey***@**

***ExportDef*:**	**EXPORT "***exportDialogTitle***"**
	    menuname:"*exportMenuName***"**
	    filedescription:"*exportFileTypeDescription***"**
	    fileextension:"*exportFileType***"**
	    filename:attribute:"*attributeName***"**
	    smarticon:" (**html** | **rtf**) **"**
	    visible:*0or1*
	    [*ExportRequireFiles*]
	    [*ExportCopyFiles*]
	    [*ExportTempFiles*]
	    { *SourceDef* }

***ExportRequireFiles*:**	**requirefile1:"***fileName***"** [**requirefile2:"***fileName***"** [...]]
***ExportCopyFiles*:**	**copy1:"***fileName***"** [**copy2:"***fileName***"** [...]]
***ExportTempFiles*:**	**temp1:attribute:"***attributeName***"** [**temp2:attribute:"***attributeName***"** [...]]
***SourceDef*:**	*SourceModel2SGML* | *SourceAdoScript* | *SourceModelGroups* | *SourceUserVariable*

***SourceModel2SGML*:**	`SOURCE "Model2SGML"`
	`filename:attribute:"attributeName"`
	`basename:attribute:"attributeName"`
	`subprocesses:attribute:"attributeName"`
	`[ sortmode:"SortMode"]`
	`[ translation:" TranslationTableEntry`
	`{ TranslationTableEntry } " ]`
	`[ copydocuments:"absoluteOrRelativePath" ]`
	`[ modeltypes:"modelTypesName{;modelTypesName}" ]`
	`[ acfilter:attribute:"attributeName" ]`
	`((libraryspecific:0 GeneralLibDef ) |`
	`(libraryspecific:1 GeneralLibDef SpecificLibDef`
	`{ SpecificLibDef } ) )`
***SortMode*:**	`Search for spread	Search for low	Alphabetical .`
***TranslationTableEntry*:**	`@SourceAttribName@@TranslatedAttribName@`
***GeneralLibDef*:**	`LIBRARY LibDefAttrs`
***SpecificLibDef*:**	`LIBRARY "modelTypeName" LibDefAttrs`
***LibDefAttrs*:**	`mode:"modelTypeModeName"`
	`notebookattr:"attrModeName"`
	`graphics:0or1`
	`[ gfxformat:"GfxFormat" ]`
	`[ gfxdpi:gfxSize ]`
	`[ gfxmode:" ( unchanged | modelTypeModeName ) " ]`
	`[ gfxorientation:"GfxOrientation" ]`
	`[ gfxlayout:"layoutName" ]`
	`[ gfxdozoom:0or1`
	`gfxzoomlevels:" gfxSize { ; gfxSize } " ]`
***GfxFormat*:**	`bmp	bmp1	bmp24	jpg	pcx8	pcx24	png	emf`
***GfxOrientation*:**	`unchanged |`
	`rotate left (counter clockwise) |`
	`rotate right (clockwise) |`
	`rotate by 180°`
***SourceAdoScript*:**	`SOURCE "AdoScript"`
	`name:"scriptName"`
	`script: { AdoScript }`
	`VarDeclList`
***SourceModelGroups*:**	`SOURCE "ModelGroups"`
	`filename:attribute:"attributeName"`
	`exportall:0or1`
***SourceUserVariable*:**	`SOURCE "UserVariable"`
	`filename:attribute:"attributeName"`
	`VarDeclList`
***VarDeclList*:**	`var1:attribute:"attributeName"`
	`[ var2:attribute:"attributeName" [...]]`
***DialogDef*:**	`DIALOG { Notebook }`
***0or1*:**	`0	1 .`

2.1.4   Predefined Analysis queries

The predefined analysis queries can be user-specifically defined and be made available to the ADONIS users in the Analysis Component of the ADONIS Business Process Management Toolkit. Using pre-defined analysis queries, the attribute values of objects and connectors (which have usually been input by ADONIS users) can be analysed according to defined criteria and the appropriate results can be rendered.

If you wish to create, edit or delete the predefined analysis queries of a library, select this library in the window **"Library management - library configuration"** (see fig. 114) and then click on the button **"Predefined analysis queries"**.

The window **"<Library Name> - Edit queries"** (see fig. 176) will be displayed, where instead of <Library Name> the name of the previously selected library is shown.
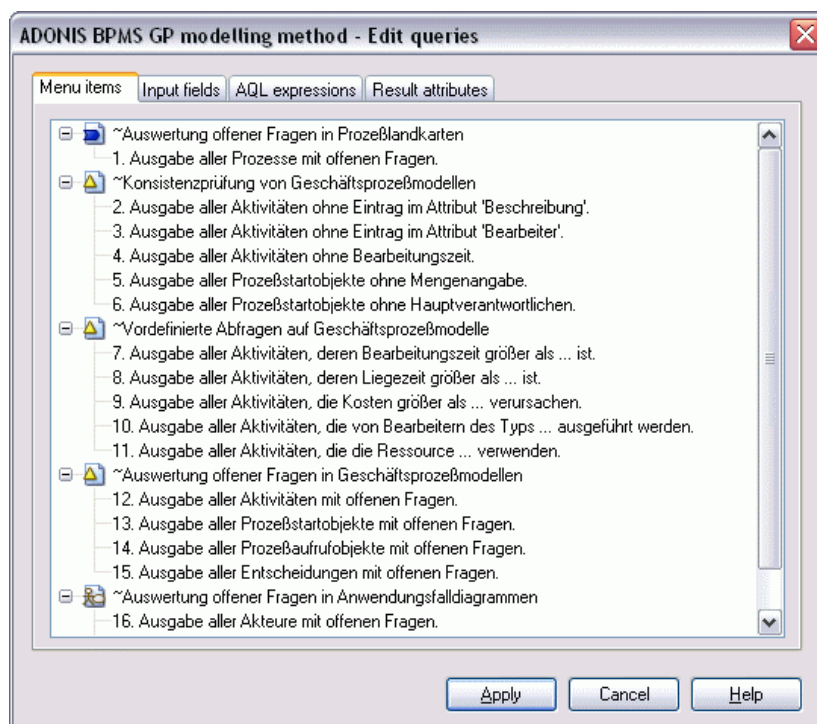
Figure 176: Predefined analysis queries

The predefined queries consist of four parts (attributes) represented by four tabs:

● Menu items (see chap. 2.1.4.1, p. 255)

● Input fields (see chap. 2.1.4.2, p. 259)

● AQL expressions (see chap. 2.1.4.3, p. 264)

● Result attributes (see chap. 2.1.4.4, p. 268)

For every new or existing query, these parts can be edited independently of each other by clicking on the respective tab.

After you have edited the pre-defined analysis queries, click on the button **"Assign"** in order to save changes in the library.

The procedure for predefining an analysis query can best be conceived with an example (see chap. 2.1.4.5, p. 269) from the ADONIS BPMS Application Library (see chap. 20., p. 693).

2.1.4.1 Menu Items

The menu items defined in the tab **"Menu items"** are displayed in the ADONIS Business Process Management Toolkit among the predefined queries available for selection:
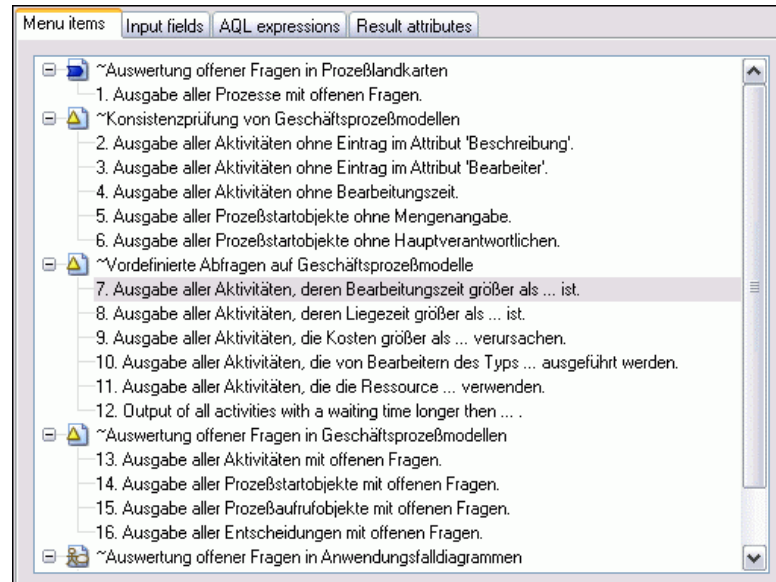
Figure 177: Menu items

A menu item refers to a model type. The model type's symbol is displayed infront of each menu option in the "Menu items" window. Each menu option may include one or more queries.

Create menu item

If you wish to add a new menu option, select that of the existing menu option above or below the position you want to insert the new menu option. Then open the **pop-up menu** by clicking on the right mouse-button and select the option **"New"**.

In the window "Insert new element" (see fig. 178) select the entry "Menu item" from the list "New element", then specify the insertion position and click on the **"OK"** button.

Within the window "Edit text field" (see fig. 180) you can enter text for the menu option and select the model type to which the query refers.

> **Hint:** If you would like to define a short-cut you have to enter a tilde (~) prior to the letter desired in the text field of that menu option.
>
> Make sure that the short cuts defined are unique.

After entering the text, the short cut's definition and the model type, click on the "OK" button to create the new menu option.

> **Hint:** If you wish to add the new menu option in the last position, you do not need to previously select an existing menu option.

Create new query

A new query is created in a similar manner to a new menu option. You select either the menu option within which you want to create a new query (the new query will be inserted as the last query) or if you

wish to place the new query in a certain position, select one of the queries belonging to the menu option desired.

Then open the **pop-up menu** by clicking on the right mouse-button, select the menu option **"New"**, select "Query" from the "New element" listbox and define - if necessary - the insertion position in the window "Insert new element" (see fig. 178). Click on the "OK" button and enter a text for the query within the window "Edit text field" (see fig. 179)

After entering the text, click on the **"OK"** button to create the new query.

Change menu item/query

You can change the text of queries or menu options or the model type to which a menu option has been assigned at any time. To do so select the element (query or menu option) you wish to alter, open the **pop-up menu** (right mouse-button) and select the menu option **"Edit"**. The window "Edit text field" in which you can change the text and the model type of the menu option (see fig. 178) or the text of the query (see fig. 179) appears. Then click on the "OK" button and the changes will be performed and displayed.

Delete menu item/query

If you wish to delete single queries or menu options (including all their queries), select the particular element (query or menu option), open the **pop-up menu** (right mouse-button) and select the menu option **"Delete"**. The query or menu option will be deleted after a confirmation query.

Edit attributes of a query

In order to edit one of the attributes (Input fields, AQL expressions or Result attributes (see chap. 2.1.4.4, p. 268)) select the particular query and click on the tab.

> **Hint:** When you first create a new query, the attributes "Input fields", "AQL expressions" and "Result attributes" are empty.

Add new element

A new element is inserted by selecting an existing element (if the new element is to be inserted above or below the element selected) and selecting the menu option **"New"** from the **popup-menu** (right mouse button).

> **Hint:** If the window is empty (e.g. "input fields" after creating a new query) simply right-click into the empty window.

The window "Insert New Element" appears (see fig. 178).
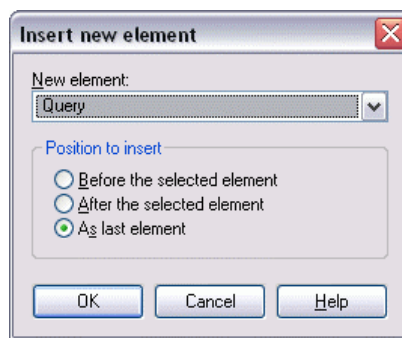
Figure 178: Add new element

Select one of the following entries from the list "New Element":

Menu Items	Tab "Menu items" (see chap. 2.1.4.1, p. 255); adds a new menu option for queries on a certain model type.
Query	Tab "Menu items" (see chap. 2.1.4.1, p. 255); adds a new query to an existing menu option.
Attribute field	Tab "Input fields" (see chap. 2.1.4.2, p. 259); adds a new selection field for the attributes of a class to the query.
Attribute value field	Tab "Input fields" (see chap. 2.1.4.2, p. 259); adds a new selection field for the concrete value of a class' attribute to the query.
Enumeration value field	Tab "Input fields" (see chap. 2.1.4.2, p. 259); adds a new selection field, the values of which have to be defined user-specifically to the query.
Attribute enumeration value field	Tab "Input fields" (see chap. 2.1.4.2, p. 259); adds a new selection field for the possible value forms of an attribute in a query.
Edit field	Tab "Input fields" (see chap. 2.1.4.2, p. 259); adds a new input field where the user of the Business Process Management Toolkit can add text to the query.
Class field	Tab "Input fields" (see chap. 2.1.4.2, p. 259); adds a new selection field for classes to the query.
Text	Tab "Input fields" (see chap. 2.1.4.2, p. 259); adds new text to the query.
AQL part	Tab "AQL expressions" (see chap. 2.1.4.3, p. 264); adds a new AQL expression for performing the query to the query.
Plan	Tab "AQL expressions" (see chap. 2.1.4.3, p. 264); adds a new definition for the generation of plans.
	Hint: only one plan element can be generated in the AQL expressions of a query.
Reference	Tab "AQL expressions" (see chap. 2.1.4.3, p. 264); adds a new reference to a selection or an input field used when running the query.
Attribute	Tab "Result attributes" (see chap. 2.1.4.4, p. 268); adds a new attribute which will be displayed in the query's representation of results.

Should you have selected an existing element, you may in addition specify the position where the new element is inserted. The default setting is "As last element".

> **Hint:** If you did not select an existing element before inserting the new element, the new one will be inserted in the last position.

Confirm your input by clicking on the "OK" button and define - if necessary - the parts of the new elements in the respective windows.

Element	Window
Menu items	Edit text field (see fig. 180)
Query	Edit text field (see fig. 179)
Attribute fields	Class selection (see fig. 182)
Attribute value field	Attribute enumeration value field (see fig. 183)
Enumeration value field	Edit enumeration values (see fig. 184)
Enumeration attribute value field	Attribute selection (Enumeration)
Edit field	Edit field length (see fig. 185)
Class Field	-
Text	Edit text field (see fig. 179)
AQL part	Edit query part (see fig. 188)
Plan	Edit plan definition (see fig. 191)
Reference	-
Attribute	Attribute selection (see fig. 183)

Edit Text Field

You can edit the text in the elements "Menu item", "Query" or "Text" by selecting the respective element and selecting the menu option **"Edit"** in the **pop-up menu** (right mouse button).

> **Hint:** When inserting a new element (see p. 256) of one of the above types, the window "Edit text field" is automatically displayed as soon as the new element has been defined.

In the window "Edit Text Field" (see fig. 179) you can enter or change the text of the element "Query" (Tab "Menu items" (see chap. 2.1.4.1, p. 255)) or the text of the element "Text" (Tab "Input fields" (see chap. 2.1.4.2, p. 259)).
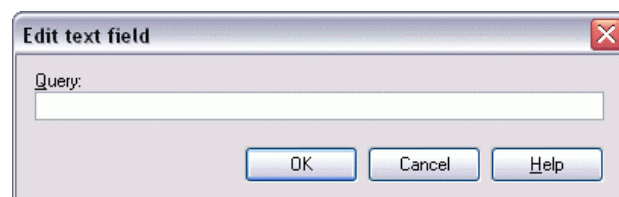
Figure 179: Edit text field

In the window "Edit text field" (see fig. 180) you can enter or change the text of the element "Menu item" (Tab "Menu items" (see chap. 2.1.4.1, p. 255)) and in addition select the model type on which the query is to be executed.
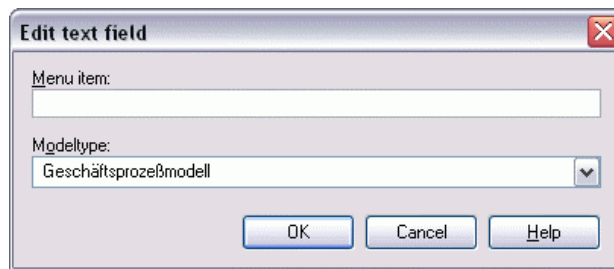
Figure 180: Edit text field and model type

Confirm your input by clicking on the **"OK"** button.

2.1.4.2 Input fields

In the tab "Input fields" those input fields which must be filled in by the ADONIS user when a query is being executed are defined.
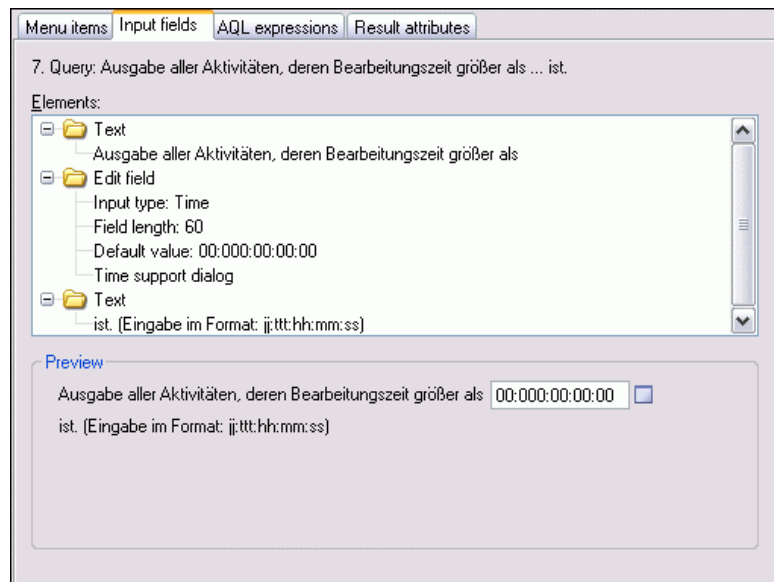
Figure 181: Input fields

Input fields serve the purpose of describing the query. The following elements are available for defining input fields:

- **Text field** (constant text, which describes the query or explains the text and selection fields),

- **Edit field** (field, where the ADONIS user may enter any text),

- **Attribute field** (field, where the ADONIS user can select an attribute of a certain class).

- **Attribute value field** (field, where the ADONIS user can select a concrete value of a certain class' attribute).

- **Attribute enumeration value field** (field, where the ADONIS user can select a concrete value for an attribute).

- **Enumeration value field** (field, where the ADONIS user can select a value from a set of prede-fined values).

- **Class field** (field, where the ADONIS user can select a class of the respective library).

The current elements selected (field "Elements") are displayed in the order of their definition in the field "Preview". This corresponds to the display of the query in the ADONIS Business Process Management Toolkit. Each change to this list - be it the insertion of a new element or the editing or deletion of an existing element - is immediately shown in the field "Preview".

Enter a new element for the input field

In order to insert a new element into the list, select the element in front of or behind the position in which the new element should be inserted and select the menu option "New" in the popup-menu (right mouse-button).

Select the element in the window "Insert New Element" (see fig. 178) from the list "New Element", then specify the insertion position and click on the "OK" button.

When selecting the elements "Attribute field" (see p. 260), "Attribute value field" (see p. 261), "Enumeration value field" (see p. 262), "Edit field" (see p. 262) and "Text" (see p. 258), you have to make additional entries. After making an entry, always click on the "OK" button in the window to insert the new element into the list.

> **Hint:** If you do not select an existing element, the new element will be inserted into the last position in the list.

Change existing element

To edit an existing element, select the respective element and select the menu option "Edit" in the popup-menu (right mouse-button). Depending on what you have selected, the editing will take place in one of the windows listed below:

Element	Window
Attribute Field	Class Selection (see fig. 182)
Attribute Value Field	Attribute Selection (see fig. 183)
Enumeration Attribute Field	Enumeration Attribute Selection (see fig. 186)
Enumeration Value Field	Edit Enumeration Values (see fig. 184)
Input Value	Edit Field Length (see fig. 185)
Text	Edit Text Field (see fig. 179)

Table 3: Editing elements - window titles

Delete Existing Element

If you wish to delete an existing element, select that element, open the popup-menu (right mouse-button) and select the menu-option "Delete". The element you selected before is deleted after a confirmation query.

Class selection

You can specify the class for the attribute selection of the element "Attribute Field" by selecting the element and choosing the menu option "Edit" from the popup-menu (right mouse-button).

> **Hint:** When inserting a new element (see p. 256) "Attribute Field" the window "Class Selection" is automatically displayed as soon as the new element has been defined.

In the window "Class Selection" (see fig. 182) you determine by selecting a class those attributes which are to be shown in the element "Attribute Field" (Tab "Input Fields" (see chap. 2.1.4.2, p. 259)) within the query.

Figure 182: Class selection

Confirm the collection by clicking on the "OK" button.

Attribute selection

You can specify the attribute for the value selection of the elements "Attribute Value Field" or "Attribute" by selecting the respective element and choosing the menu option "Edit" in the pop-up menu (right mouse button).

>   **Hint:** When inserting a new element (see p. 256) of the types mentioned, above the window, "Attribute Selection" is automatically displayed as soon as the new element has been defined.

In the window "Attribute Selection" (see fig. 183) you specify by selecting a class' attribute which values are to be displayed in the element "Attribute Value Field" (Tab "Input Fields" (see chap. 2.1.4.2, p. 259)) in the query.
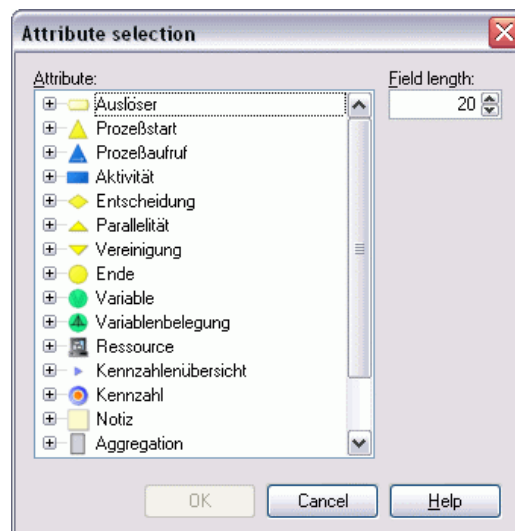
Figure 183: Attribute selection-attribute value field

The attribute values shown are concrete values which are stored in the objects of the models selected for the query. Using the field length you specify the size of the selection field (the preview option supports you when defining the size).

Confirm your selection by clicking on the "OK" button.

Edit enumeration values

You can specify the values for the selection of the element "Attribute Value Field" (Tab "Input Fields") by selecting the element and choosing the menu option "Edit" from the popup-menu (right mouse-button).

> **Hint:** When inserting a new element (see p. 256) of the type "Attribute Value Field" the window "Edit enumeration values" is automatically displayed as soon as the new element has been defined.

In the window "Edit enumeration values" (see fig. 184) you define a value by entering it into the field "Value" which you then add to the list by clicking the button "Add".
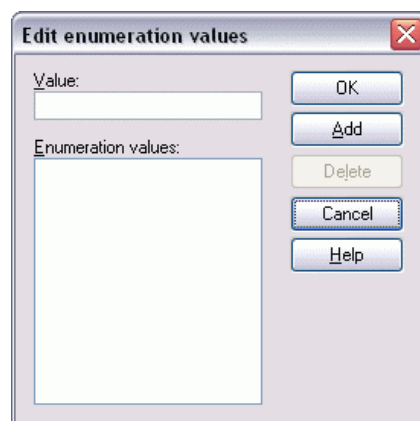
Figure 184: Edit enumeration values

From the list of enumeration values the ADONIS user can select a concrete value when he is about to execute the query.

To change an existing entry in the list "Enumeration values", double-click on it. The entry is then shown in the field "Value" and can be edited.

If you wish to remove an entry from the list "Enumeration values", select it and then click on the button "Delete".

Finish the editing session by clicking on the "OK" button.

Edit field length

You can define the field length of the element "Edit field" (Tab "Input Fields") by selecting the element and choosing the menu option "Edit" in the popup-menu (right mouse-button).

> **Hint:** When inserting a new element (see p. 256) of the type "Edit field" the window "Edit field length" is automatically displayed as soon as the new element has been defined.

In window "To edit Edit field box" (see fig. 185) the user defines features of field (text, time, integer), the length of the edit field box and standard values. Additionally, it is possible to activate the option "Time support" for pre-defined queries, which will results in help when entering data.
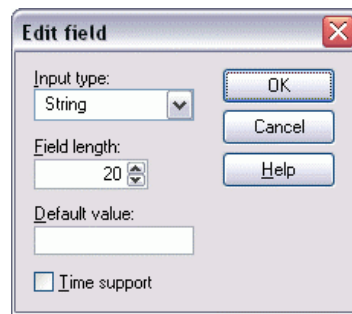
Figure 185: Edit box length

Finish your input by clicking on the "OK" button.

>   **Hint:** The time format entered will be checked for correctness during the execution of the predefined query in the Business Process Management Toolkit.

Enumeration attribute field

You can define the attribute by selecting values for "Enumeration attribute field" (tab "Entering field"). It is done by the selection of the elements and choosing in the context menu the menu option (with right mouse button) "Edit".

>   **Hint:** By adding a new element to (see p. 256) "Enumeration attribute field", the window "Selection of the enumeration attribute" is shown automatically.

In the window "Selection of the enumeration attribute" (see fig. 186) you decide, via selection an attribute from one class, that values defined for this attribute in an application library should be displayed in the "Enumeration attribute field" (tab "Entering field").
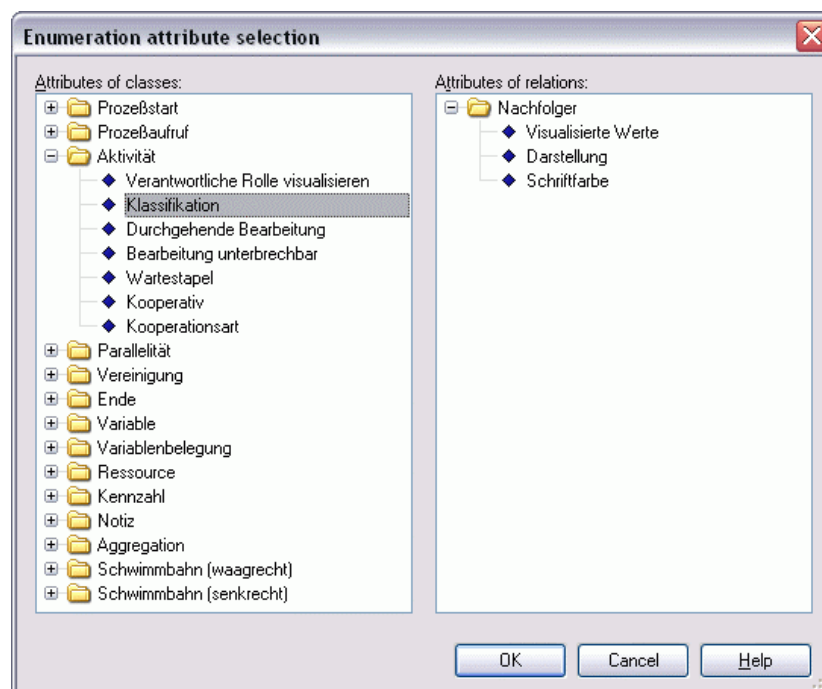
Figure 186: Enumeration attribute selection

Please confirm your choice by clicking on the OK button.

2.1.4.3 AQL expressions

In the tab "AQL expressions" (see fig. 187) you define those AQL expressions (AQL= **A**DONIS **Q**uery **L**anguage (see chap. 12., p. 593)) which perform the query.
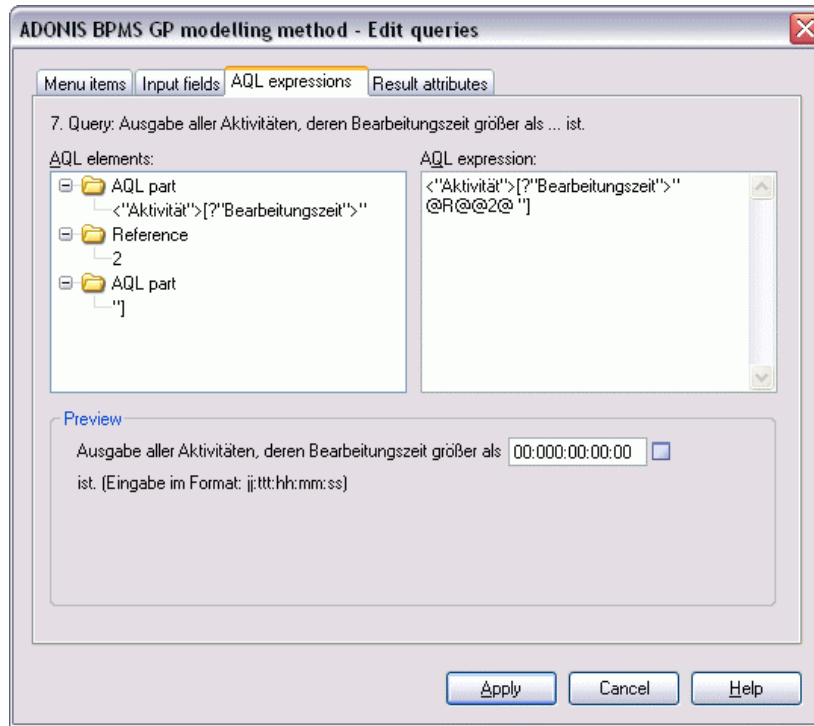
Figure 187: AQL expressions

The following AQL elements are available for defining AQL expressions

- **AQL part** (part of the AQL command text),

- **Reference** (number, which serves to reference the optionally defined text, input and selection field in the AQL expression),

- **Plan** (specific output format of a query; here the number of columns generated and a generic title may be defined).

The latest AQL statement is displayed in the field "AQL expression" and combines the different AQL elements (AQL parts and reference numbers) to a complete and syntactically correct AQL expression.

Add AQL expression

In order to insert a new AQL element into the list, select the AQL element in front of or behind the position you wish to insert the new AQL element and select in the popup-menu "New".

Select the AQL element in the window "Insert new element" (see fig. 178) from the list "New element", specify the insertion position and then click on the "OK" button.

When selecting the AQL elements "AQL Part" (see p. 265) and "Plan" (see p. 266) you have to make additional entries. Click on the "OK" button in the respective window after having finished your entries to add the new element to the list.

> **Hint:** If you do not select an existing AQL element, the new AQL element is added at the end of the list.

Add reference

With the help of the AQL element "Reference" the entries made by the ADONIS user are handled via reference numbers to the AQL expression. A reference number is assigned to each input field (see chap. 2.1.4.2, p. 259). The numbering is serial and starts with one (1).

If you wish to insert or change an AQL element "Reference", select the AQL element in front of or behind the position the reference is to be inserted and select the menu option "New" in the popup-menu (right mouse-button).

The window "AQL - Create reference" (see fig. 188) showing the current AQL expression is shown.
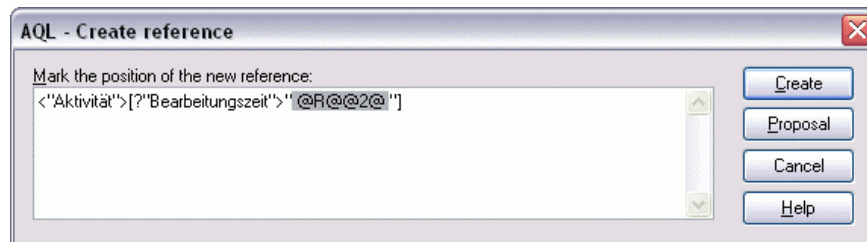
Figure 188: Create reference

Please highlight all the dynamic parts of AQL expressions, which should be presented as a reference. Through repeated clicking on the button **"Suggestion"**, all the reasonable positions in the AQL expression will be highlighted.

Click on the button **"Generate"**, after selecting all the desired positions. Your chosen reference will be taken over and the AQL expression automatically divided in such a way, that the reference will be shown in the previously selected place.

> **Hint:** In case there is only one dynamic element in the entering field (entering field or field with positions to choose), the reference number is automatically assigned to this field.

If there are **many dynamic elements** in the entering field, the reference number will be assigned normally to the first dynamic element and a window with a question whether the reference should be assigned to any other element, will be displayed (see fig. 189).
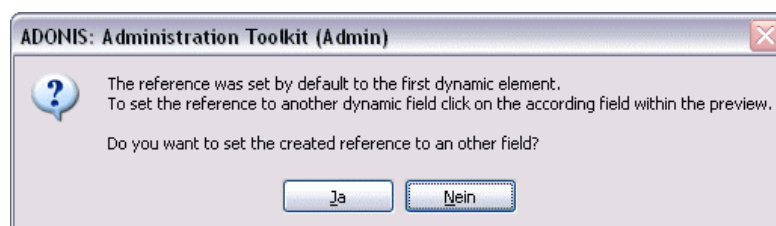
Figure 189: Automatic reference assignment to the first dynamic element

Click on the button **No**, to keep the assigned reference as it is. It is possible to change (see p. 267) these settings later on.

If you click on the **Yes** button, you can choose another field (dynamic element). If the mouse pointer is

moved during a **preview** on this field - the mouse pointer will change its appearance into and at any time when it is moved onto the reference-able field. The proper reference number will be automatically entered into the AQL element "Reference".

Edit AQL element

In order to edit the existing AQL expression, please select the concerned AQL element and choose the menu option "Edit" from the context menu (right mouse button). In dependence of the selected AQL element, the editing of the element takes place (in the next windows) (see Table 4).

AQL element	Window
AQL part	Edit part of the query (see fig. 190)
Plan	Edit plan definition (see fig. 191)

Table 4: Editing of the AQL element - window title

Delete AQL element/reference

If you want to delete the existing AQL element or the exisitng reference, please select the proper AQL element/reference, open contex menu (right mouse button) and choose the option "Delete". The selected AQL element/reference will be (after your confirmation) deleted.

Edit query part

You can edit the query section of the element "AQL part" (Tab "AQL expressions") by selecting the element and then select the menu option "Edit" in the popup-menu (right mouse button).

> **Hint:** When inserting a new element (see p. 256) "AQL part" the window "Edit query section" is automatically displayed as soon as the new element has been defined.

In the window "Edit query section" (see fig. 188) an AQL expression is defined for the query to be performed.
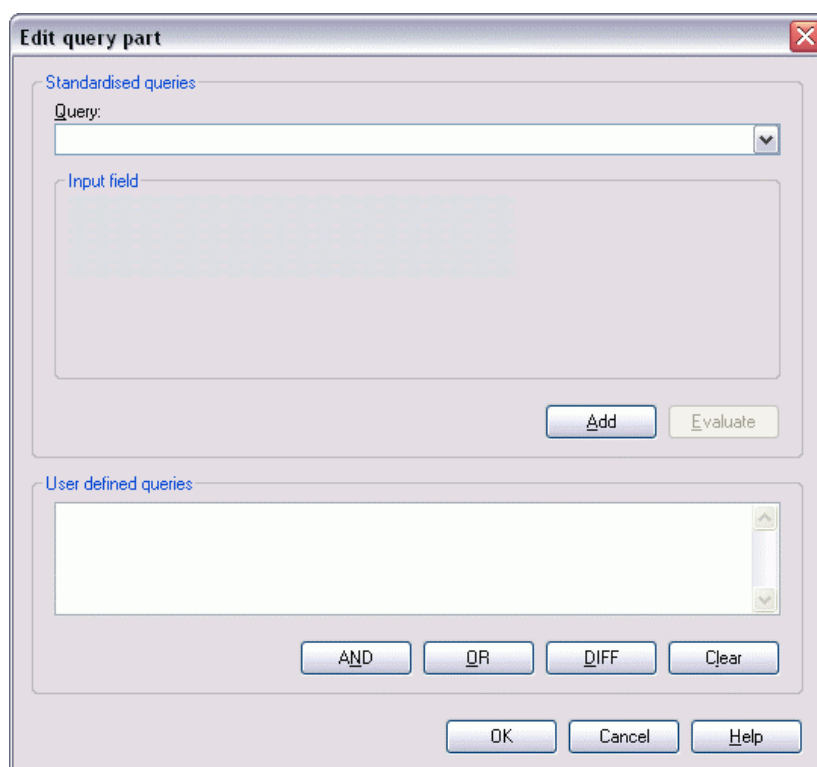
Figure 190: Edit query part

ADONIS Administration Toolkit

To define the AQL expression you can select a query from the set of standard queries and - if neces-sary - complete the input fields. Alternatively you can specify a user-defined query in AQL directly.

Note:

- The buttons "AND", "OR" and "DIFF" support you when defining a user-defined query.

- By clicking on the button "Reset" any user-defined query which has already been entered will be deleted.

- By clicking on the button "Add" you may transfer a complete standard query into the field of the user-defined query and continue editing it there.

Finish your input session by clicking on the "OK" button.

> **Hint:** Should you have created both a standard and a user-defined query, the AQL expres-sion taken when clicking on the OK button will be that of the user-defined query.

> **ATTENTION:** When using references during selection or input fields (Tab "Input fields"), several AQL parts are necessary to create a valid AQL expression. The di-vision into AQL parts automatically occurs when adding a reference.

Edit plan definition

You can edit the plan definition of the element "Plan" (Tab "AQL expressions") by selecting the ele-ment and choosing the menu option "Edit" in the popup-menu (right mouse button).

> **Hint:** When inserting a new element (see p. 256) "Plan" the window "Edit plan definition" is automatically displayed as soon as the new element has been defined.

You can enter a column heading into the window "Edit plan definition" (see fig. 191) and specify the number of columns to be displayed when a plan is being generated.
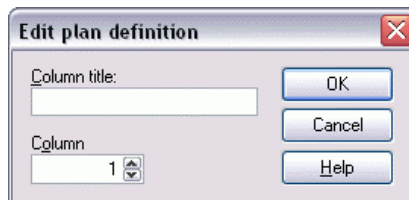
Figure 191: Edit plan definition

> **Hint:** The element "Plan" must occur only once per AQL element. Also, this element is al-ways entered as the first of the AQL elements.

Finish your input session by clicking on the "OK" button.

Edit reference

The AQL element "Reference" allows the user to copy over input entered by other ADONIS users and concerning reference numbers to the AQL expression. Every element in the input field (see chap. 2.1.4.2, p. 259) is given its reference number. The numbering occurs in ascending order, starting with one (1).

To change the reference being created (i.e. to choose different elements in the input field), please choose, after selecting the proper AQL element "Reference", menu option "Edit" from the context

menu (right mouse button). The mouse pointer will be changed into   . Please move the mouse pointer in the **preview** onto each input field, or field with positions to choose, to which you want to

have reference to - the mouse pointer will change its appearance into ![icon] if it will be moved onto the reference-able field and the user will click on this field. The reference number will be automatically entered into the AQL element "Reference".

2.1.4.4   Result attributes

In the tab "Result attributes" (see fig. 192) you specify which objects and which attributes are to be represented as query results.

> **Hint: ADONIS Version 3.9** also allows queries on relations. For this reason, the result attributes are represented separately as attributes of classes and attributes of relations.
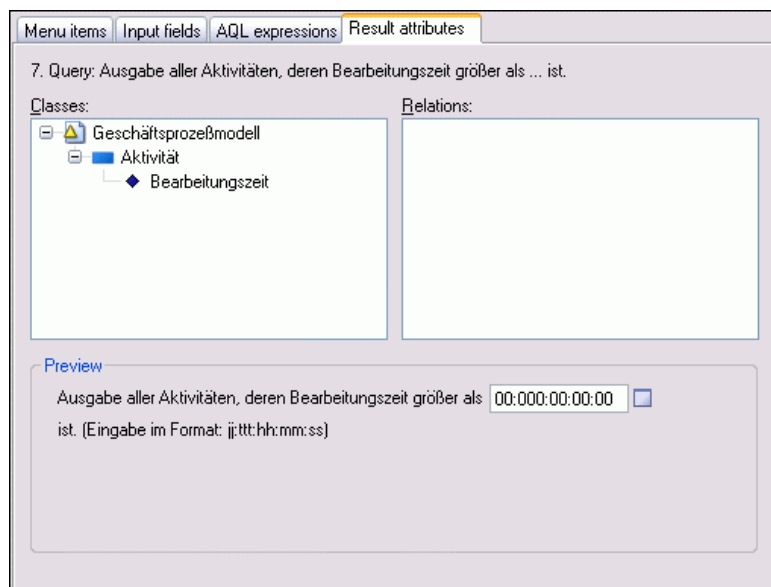
Figure 192: Result attributes

The element "Attribute" is available for use in defining result attributes. This specifies the attributes of a class and/or relation which shall be displayed.

Insert new element

To insert a new element into the list, select the element in front of or behind the position you wish to insert the new element and then select the menu option "New" in the popup-menu (right mouse-button). Click on the "OK" button and select the attribute desired for the result representation in the window "Attribute Selection" (see fig. 183).

> **Hint:** When you have selected an attribute, the window "Attribute selection" (see fig. 183) only shows the corresponding class. To insert an attribute into a different class, you must select and expand the class so that the attributes (in their Notebook chapters) can be seen and selected. Select an attribute by clicking on it.

Following the attribute selection click on the "OK" button to create the new element.

> **Hint:** If you do not select an existing element, the new element is added at the end of the list.

Change existing elements

To edit an existing element, select the respective element and choose the menu option "Edit" from the popup-menu (right mouse button). The window "Attribute selection" (see fig. 183) appears, where you may change the attribute selection. Following the selection, click on the "OK" button and the changes will be displayed.

Delete existing elements

If you wish to delete an existing element, select the respective element and select the menu option "Delete" from the popup-menu (right mouse button). The AQL element will be deleted after a confirmation query.

2.1.4.5   Example of definition of a pre-defined query

This example shows the recommendations of how to define a predefined query. In our example a "Predefined query on business process models", which allows the ADONIS user to display all activities with waiting time grater than the time given by the user will be added.

1. **Add query**

    -   Select in the window "<Library name> - edit queries" the already defined one "Display all activities with resting time greater than..." (see fig. 193).
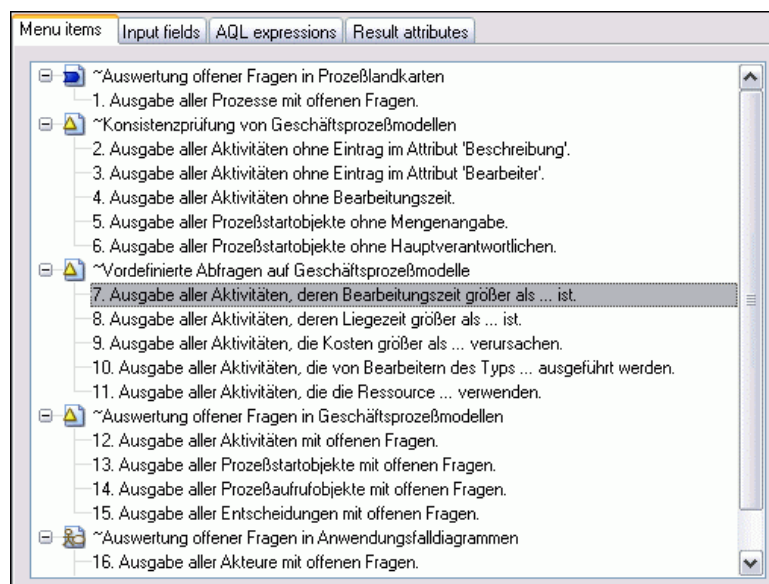
Figure 193: Add query (example)

    -   Open the context menu (with the right mouse button) and choose menu option "New". In the window "Insert new element" the user can define where the query should be inserted (e.g. "After the selected element") (see fig. 194).
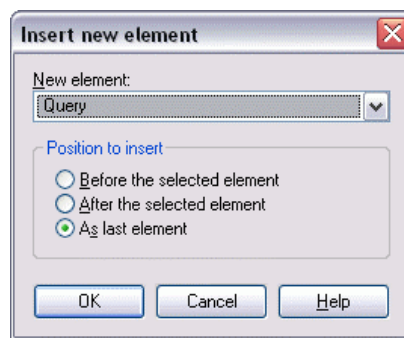
Figure 194: Add new element (example)

- In the window "Edit text field" (see fig. 195) enter text for the new query.
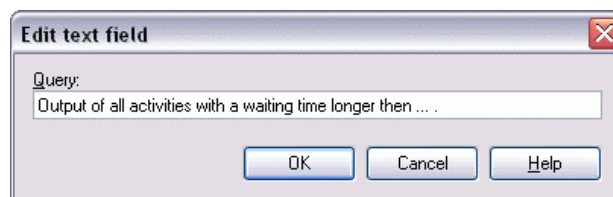
Figure 195: Edit text field (example)

- The new query will be created. Select the new query to define edit fields, AQL expressions and result attributes (see fig. 196).
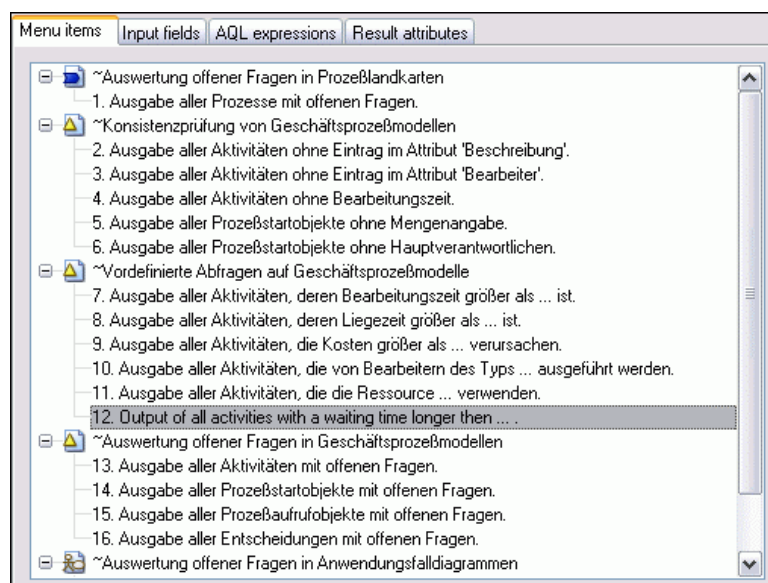
Figure 196: Select new query (example)

2. **To define input fields**

- Please go to the tab "Input fields", where the text for the selected (new) query will be shown (inclusive ordering number) (see fig. 197).
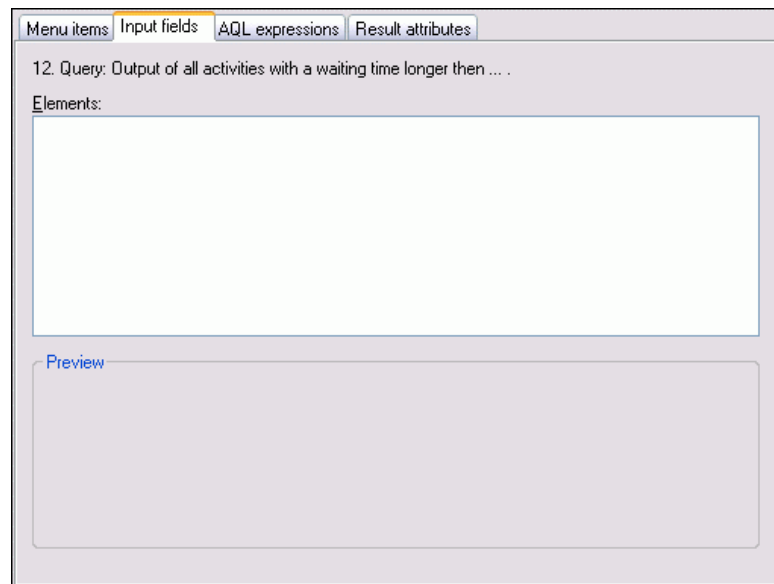
Figure 197: To define input fields (example)

- Click with the right mouse button on the field "Elements" and choose from the context menu the option "New". As a result, in the window "Insert new element" (see fig. 198) you should choose "Text" as a new element and define its "Position to insert".
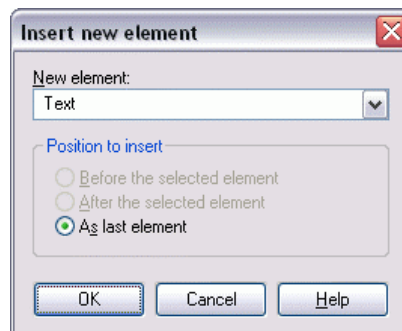
Figure 198: Insert the element "Text" (example)

- In window "Edit text field" give the first element, i.e. the first part of the query, (see fig. 199).
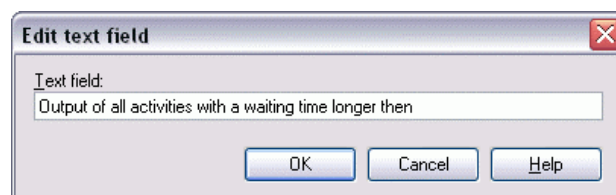
Figure 199: To enter text for the first element (example)

- Click with the right mouse button on the field "Elements" and choose from the context menu the option "New". As a result, in the window "Insert new element" (see fig. 200) you should choose "Input field" as a new element and define its "Position to insert".
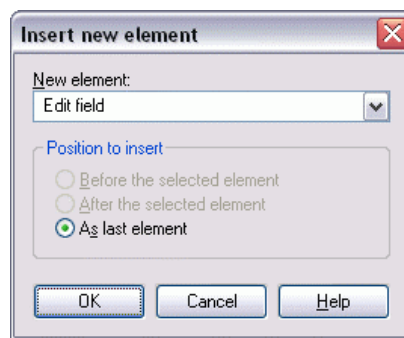
Figure 200: To add element to "Input field" (example)

- In the window "Edit field" please define input type "Time" and the field length "60". In the field "Default value" enter the value "00:000:00:00:00" (zero in ADONIS time format) and activate the option "Time support" (see fig. 201).
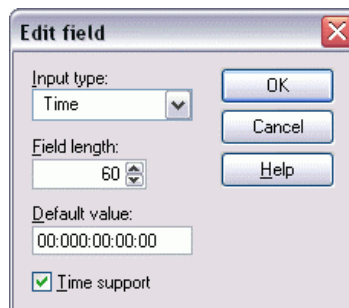
Figure 201: To edit settings of an input field (example)

- Click with the right mouse button on the field "Elements" and choose from the context menu the option "New". As a result, in the window "Insert new element" (see fig. 198) you should choose "Text" as a new element and define its "Position to insert".

- In the window "Edit text field" enter the text of the third element (see fig. 202).
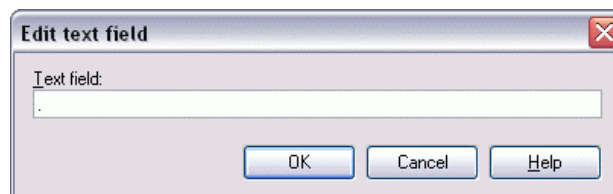
Figure 202: To enter text for the third element(example)

- Click with the right mouse button on the field "Elements" and choose from the context menu the option "New". As a result, in the window "Insert new element" (see fig. 198) you should choose "Text" as a new element and define its "Position to insert".

- In window "Edit text field" enter the text for the fourth element (see fig. 203).
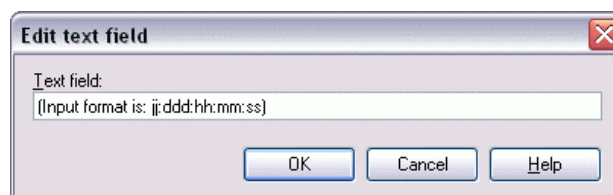
Figure 203: To enter text for the fourth element (example)

- The query is shown in the field Preview in the way it is displayed to the ADONIS user (see fig. 204).
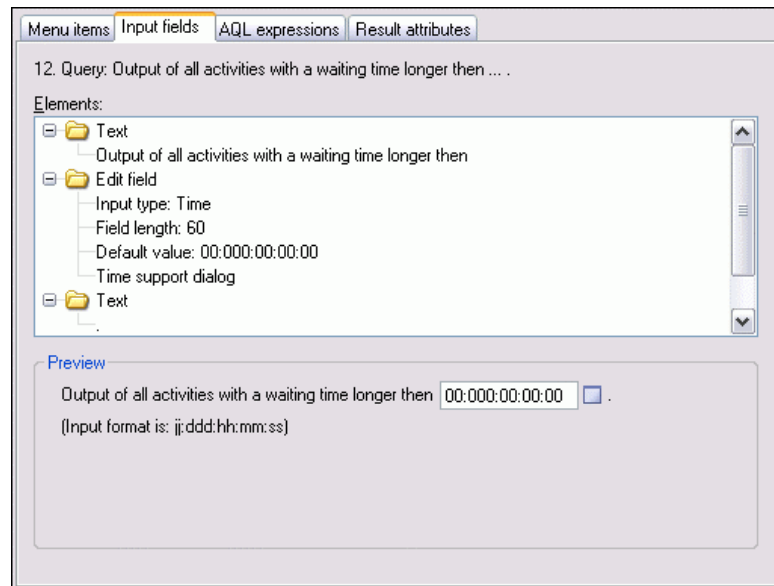
Figure 204: Preview of the input fields (example)

Due to the active option "Time support", the input field for entering the time is followed by the button ☐, which calls a support window.

3. **To define AQL expressions**

- Change the text of the query in the "AQL expression" tab, so that it is the same as seen in the preview (see fig. 205).
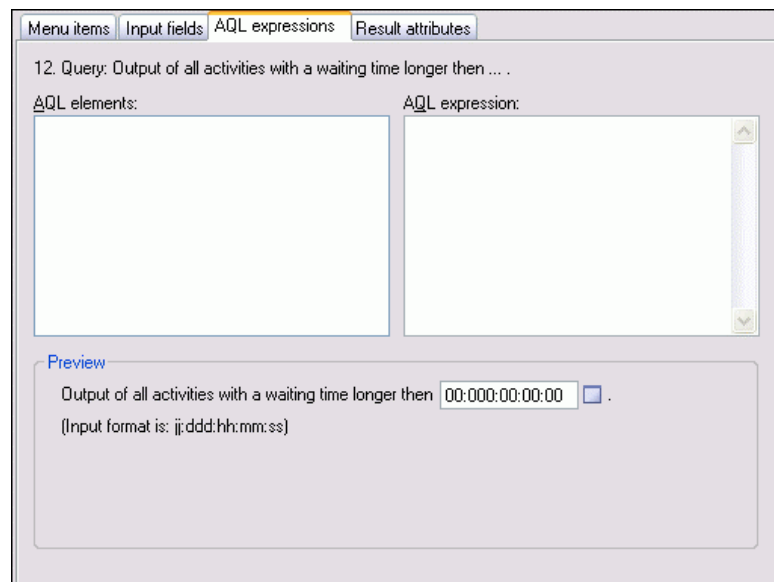
Figure 205: To define AQL expressions (example)

- Click with the right mouse button on the field "AQL elements" and choose from the context menu the option "New". Then choose in the window "Insert new element" (see fig. 206) the new "AQL part" to be inserted. The position to insert is set in advance.
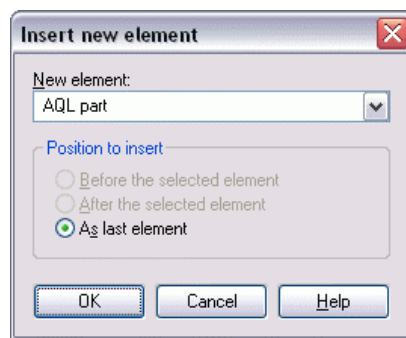
Figure 206: Insert a new AQL element (example)

- In the window "Edit part of a query" you can define a query, which allows you to choose the standard query "All objects of a class... with an attribute ..." and input data into input field (see fig. 207), i.e. class "activity", attribute "waiting time", logical operator ">" and time value "00:000:00:00:00".
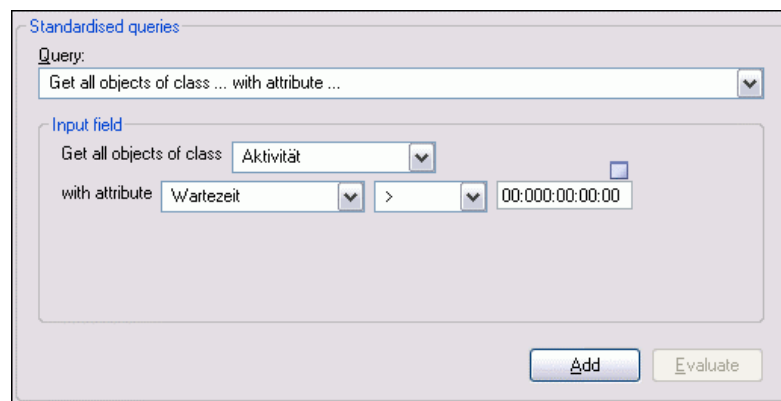
Figure 207: Define a query (example)

Finally, click on the OK button, to turn the defined query into the AQL syntax.

- Select the created element "AQL part", open the context menu (with the right mouse button) and choose the option "New". In the window "Add new element" (see fig. 208) select "Reference" as a new element. In this case, the insert position is irrelevant.
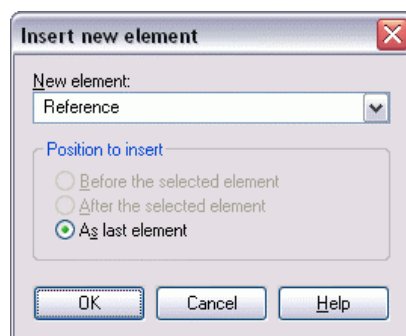
Figure 208: Insert reference into input field (example)

- In the window "AQL - create reference" the AQL expression will be displayed (see fig. 209).
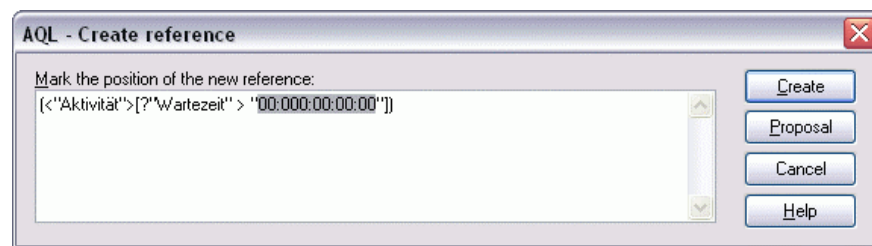
Figure 209: Create reference (example)

Click on the button "Suggestion" (if necessary many times one after another), to highlight the position of the reference (for example time giving "00:000:00:00:00"), and then click on the button "Create".

- In the window "AQL elements" the previously entered AQL expression is divided and widened with the second reference to the second input element. The correct AQL expression is shown in the window "AQL expression" (see fig. 210).

Figure 210: To show AQL expression (example)

4. **To define result attributes**

- Please change in the tab "result attributes" the text of the query, so that it is the same as during preview (see fig. 211)
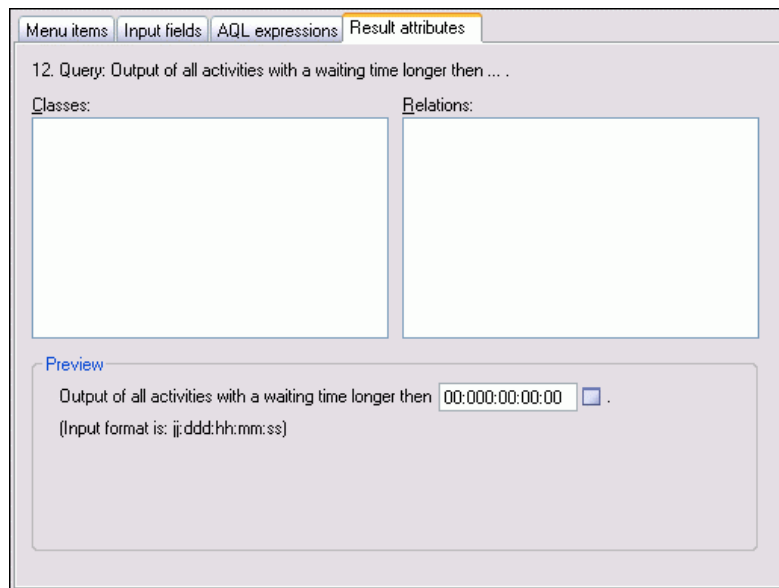
Figure 211: To define result attributes (example)

- Click with the right mouse button on the field "Classes" and choose from the context menu the option "New". In the window "Insert new element" (see fig. 212) select "Attribute" as a new element. In this case, the insert position is irrelevant.
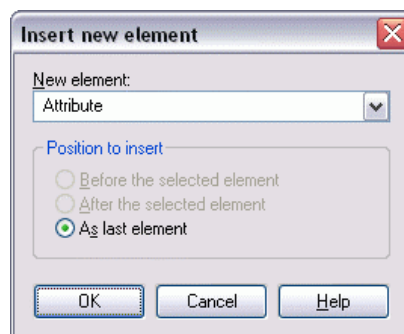
Figure 212: Insert new attribute (example)

- The attributes available to choose will be seen in the "Attribute Selection" window (see fig. 213).
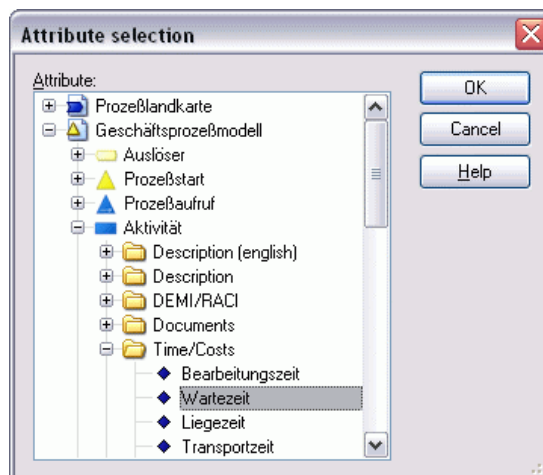
Figure 213: To choose attribute (example)

The structuring of the attribute selection is as follows:

- Model type of an application library

- Classes of the model type

- Notebook chapter of a class

- Attribute of the chapter in the notebook

e.g. choose the attribute "waiting time" in the model type "Business process model", in the class "Activity", in the chapter "Time/costs".

- The attribute "Waiting time" will be displayed in the window "Classes" (as well as a class and model type) (see fig. 214).
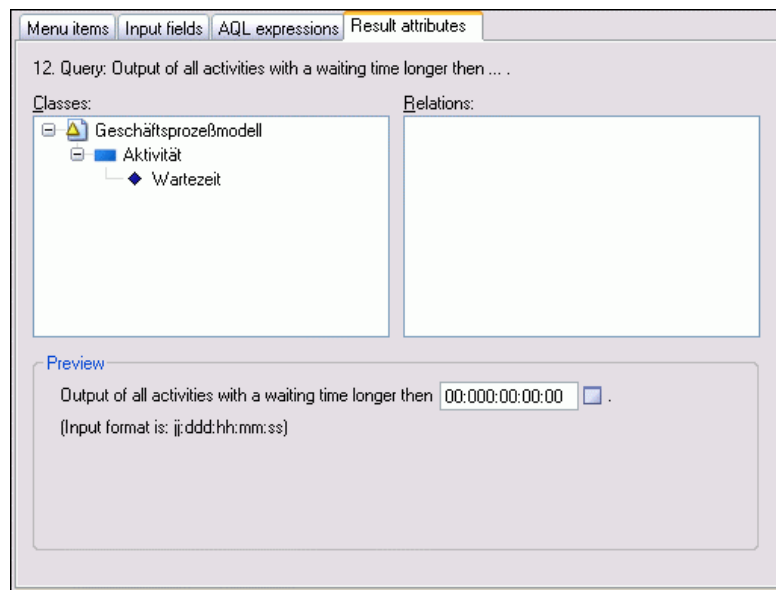
Figure 214: To display result attributes (example)

2.1.5 Predefined evaluation queries

The predefined evaluation queries can be defined user-specifically and are made available to the ADONIS users in the evaluation component of the ADONIS Business Process Management Toolkit. With the help of the predefined evaluation queries you can evaluate the attribute values of objects and connectors (which have usually been calculated by the simulation or the analytical evaluation and been saved in the models) according to defined criteria and the respective results will then be generated.

If you wish to create, edit or delete predefined evaluation queries for a library, select the respective library in the window "Library Management - Library Configuration" (see fig. 114) and then click on the button "Pre-defined Evaluation Queries".

The window "<Library Name> - Edit Evaluation Queries" (see fig. 217) will be displayed where instead of *<Library Name>* the name of the library previously selected will be shown.
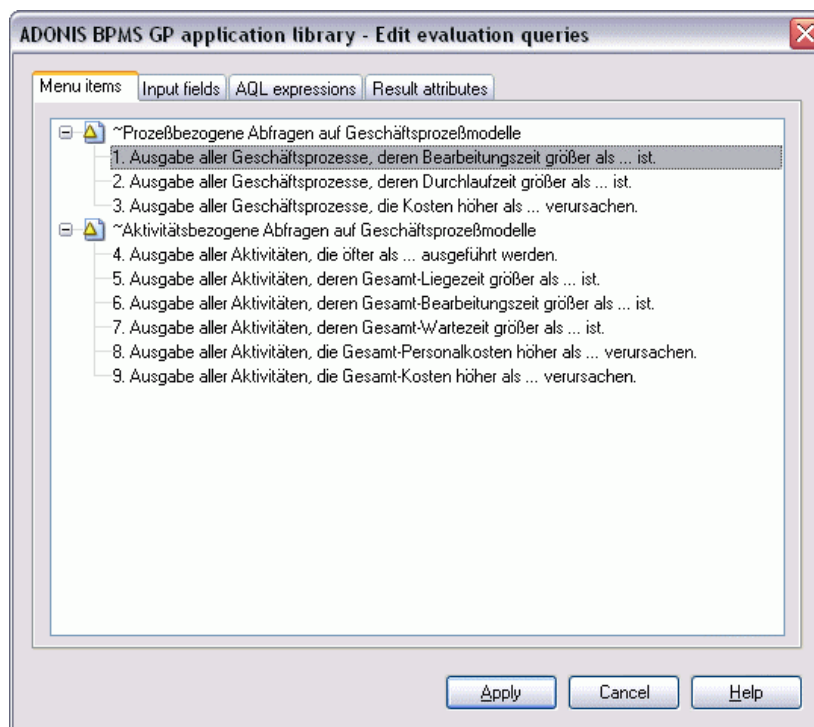
Figure 215: Predefined evaluation queries

> **Hint:** Predefined evaluation queries are created, edited and deleted in the same way as predefined analysis queries. Therefore, we refer you to the appropriate chapters of the "Predefined Analysis Queries" (see chap. 2.1.4, p. 254) section.

The predefined queries consist of four parts (attributes) represented by four tabs:

● Menu options (see chap. 2.1.4.1, p. 255)

● Input fields (see chap. 2.1.4.2, p. 259)

● AQL expressions (see chap. 2.1.4.3, p. 264)

● Result attributes (see chap. 2.1.4.4, p. 268)

For every new or existing query, these parts can be edited independently of each other by clicking on the respective tab.

After editing the predefined evaluation queries, click on the button "Assign" to save changes in the library.

2.2 Checks

This function is used as a support for checking application libraries saved in the ADONIS database.

Please choose the menu option "Check" from the menu "Library management", or click on the adequate Smart-Icon in the quick-Access bar (see chap. 3.3, p. 33).

As a result, you will see the window "Library Management" with the tab "Checks" (see fig. 216). The user can view the full list of all ADONIS application libraries, which are stored in the ADONIS database.
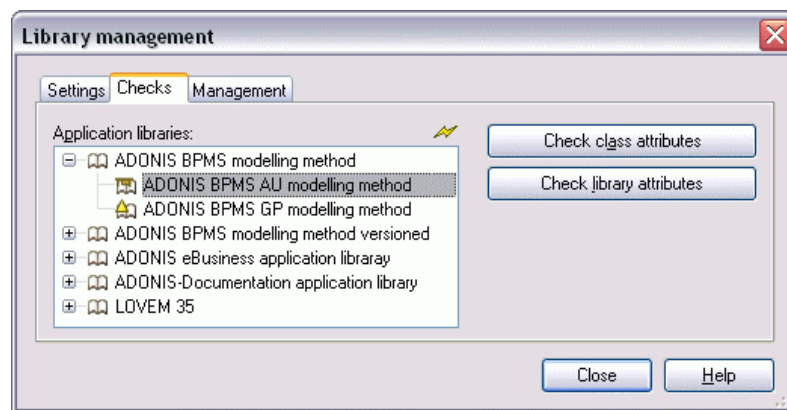
Figure 216: Library management - checks

Click twice on the edited application library, to display both the BP library and the WE library. Please select the desired library through clicking and selecting one of the following options:

● Check class attributes (see chap. 2.2.1, p. 279)

● Check library attributes (see chap. 2.2.2, p. 280)

2.2.1 Check class attributes

When checking class attributes, the definitions of the class attributes are checked for their syntactical accuracy.

> **Hint:** Once you have edited the class attributes listed above, you should check them. This avoids runtime errors when working with this library in the Business Process Management Toolkit (see chap. 4., p. 18).

If you wish to check a library's class attributes, select the respective library in the window "Library management - library configuration" (see fig. 114) and then click on the button "Check class attributes".

If the class attributes are syntactically correct, an appropriate information window will appear (see fig. 216).
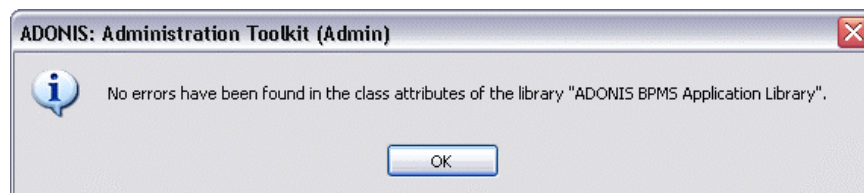
Figure 217: Check class attributes

Should the class attributes be incorrect, the error statement [alibmgt-26] is displayed. Once you have closed the error statement, a protocol (see fig. 218) will be displayed which lists the errors in the definitions.
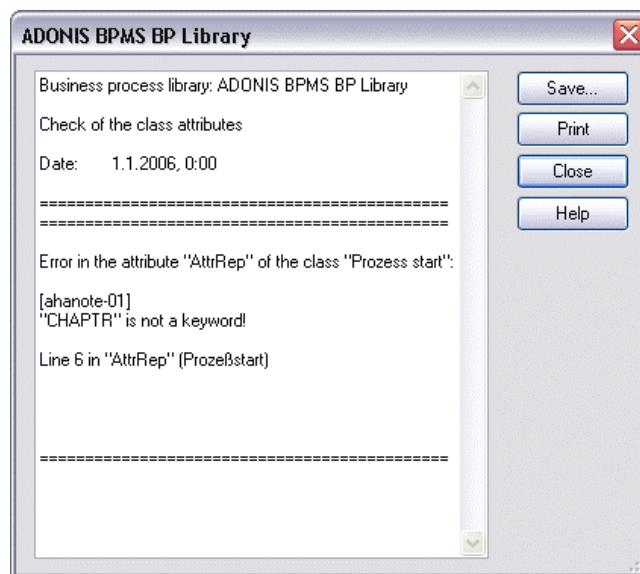
Figure 218: Check class attributes - Protocol

Please remove the errors listed (you can find additional information in the error documentation under the error number listed) or save or print this protocol and hand it over to your ADONIS administrator.

2.2.2   Check library attributes

During the check of the library attributes the definitions of all library attributes are checked for syntactical correctness

> **Hint:** Once you have edited the library attributes (see chap. 2.1.3, p. 188), you should check them. This way, you can avoid runtime errors when working with this library in the Business Process Management Toolkit (see chap. 4., p. 18).

If you wish to check the library attributes of a library, select the library in the window "Library management - library configuration" (see fig. 114) and then click on the button "Check library attributes".

If the library attributes are syntactically correct, an appropriate information window will appear (see fig. 215).
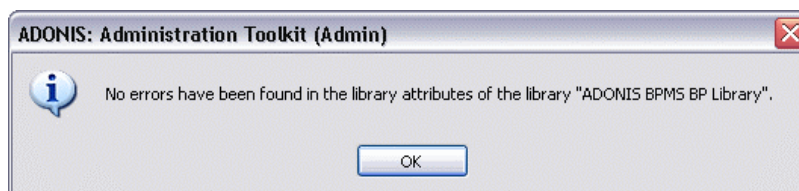
Figure 219: Check library attributes

Should the library attributes be incorrect, the error statement [alibmgt-28] is displayed. Once you have closed the error statement, a protocol window (see fig. 220) will be displayed which describes the errors in the definitions in greater detail.
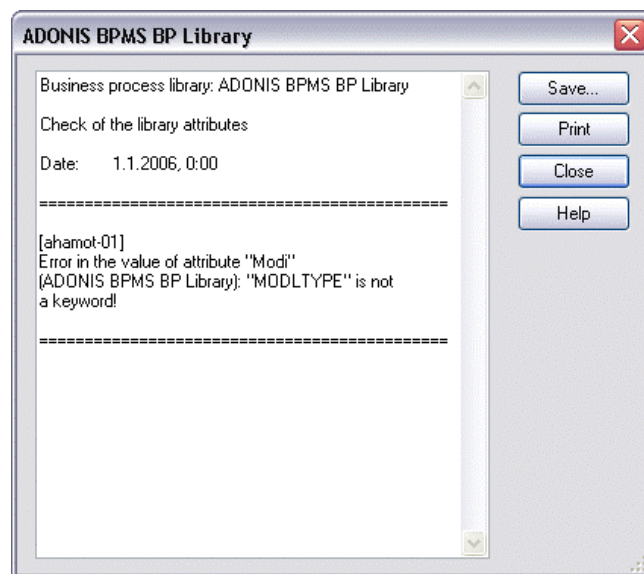
Figure 220: Check library attributes - Protocol

Please remove the errors listed (you can find additional information in the error documentation under the error number listed) or save or print this protocol and hand it over to your ADONIS administrator.

2.3   Administration

This function is used as support during management of classes and library attributes.

Please choose the option "Management" from the menu "Libraries", or click on the corresponding smart-icon 📙 in the quick-access bar.

The window "Library Management" (see fig. 221) will appear, which lists any ADONIS application libraries currently stored in the ADONIS database.
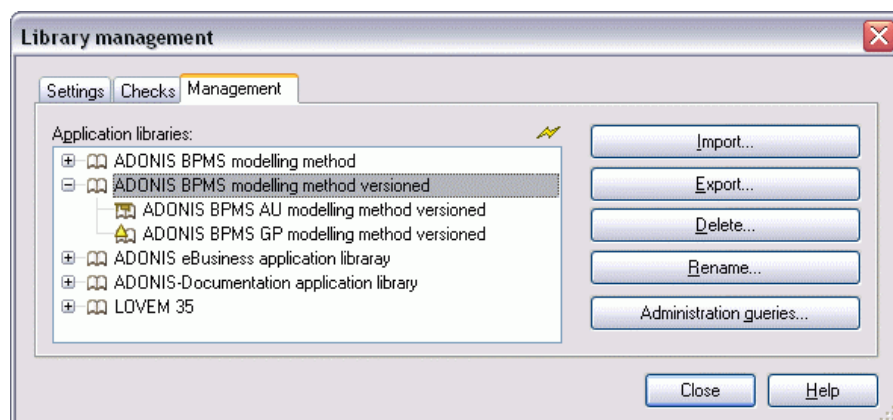
Figure 221: Library management - Management

The following functions are available within the "Library management - Management" window:

"Import"   import (see chap. 2.3.1, p. 282) an application library, which exists as an ABL file (see chap. 9.1, p. 557) in the file system, either on a CD-ROM or a hard disk;

"Export"	export (see chap. 2.3.2, p. 284) one of the application libraries listed, that is save as the ABL files to a disk (this serves backup purposes, in case the ADONIS database is to be deleted);
"Delete"	delete (see chap. 2.3.3, p. 285) one of the application libraries listed from the ADONIS database ;
"Rename"	rename one of the displayed libraries, together with the BP library and WE one (see chap. 2.3.4, p. 286).
"Administration queries"	are used for querying all users along with their models, model groups and user groups (see chap. 2.3.5, p. 286).

2.3.1 Import Application Libraries

Application libraries are imported using ABL files (see chap. 9.1, p. 557).

> **ATTENTION:** The library in the ABL file must have been generated for ADONIS Version 3.9 or 3.81 , i.e. it must have been exported from **ADONIS Version 3.9 (3.81)** or it must have been user-specifically defined for **ADONIS Version 3.9 (3.81)**. The ABL file should only contain application libraries which are not yet stored in the ADONIS database and thus not listed in the application library list. Otherwise, you will be asked to rename the library as it is imported!

Open the window "Application Libraries" (see fig. 221) (menu "Libraries", option "Application library list") and click on the button "Import".

The window "Library import" will appear (see fig. 222).
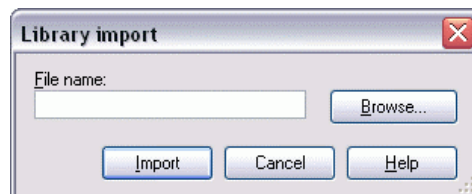
Figure 222: Library import

Enter the path and the file name of the ABL file containing the application library to be imported in the field "File name" and click on the button "Import" to start importing the application library. Please note that clicking the "Search" button will allow you to search the file system for the application library you wish to import.

> **Hint:** In case there is already an application library in the ADONIS database that have the same name as the one saved as the ABL file, you can rename the library to be imported (see chap. 2.3.1.1, p. 283).

While the library is imported, a status window will inform you about the state of the import process.

Before ending the import of an application library the query will be shown, asking the user if the standard model group "Models" and the attribute profile group "Attribute profiles" should be created.
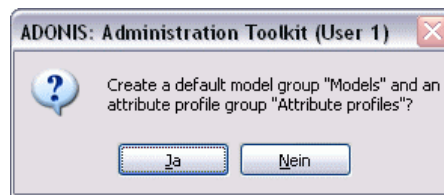
Figure 223: Create model group and attribute profile group automatically

Click on the "Yes" button to automatically import the model and attribute profile groups and create an application library. If you click the "No" button, the system will neither create a model group, nor an attribute profile group.

> **Hint:** The automatically created model group "Models" will possess write and read access rights for the "ADONIS" standard user group.

> **Hint:** Model and attribute profile groups must be defined before the models, or attribute profiles are imported (see chap. 3.4, p. 302).

When the application library has been imported successfully, the window "Library import - Result" (see fig. 224) will appear listing the libraries imported.
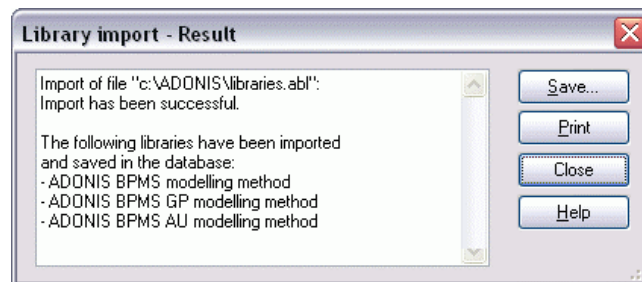
Figure 224: Library import - Result

Close the window by clicking the "Close" button or by pressing the "Enter" key. The application library imported will then be stored in the ADONIS database.

> **Hint:** Before you assign the imported application library to any ADONIS users, you should check it. You can do this within Library configuration (see chap. 2.1, p. 128).

> You should in any case check the definitions of the Class attributes (see chap. 2.2.1, p. 279) and the Library attributes (see chap. 2.2.2, p. 280) of the libraries imported to make sure that they will not cause problems when ADONIS is in use.

2.3.1.1 Rename Libraries

The names of the application libraries in the ADONIS database must be clear.

If you import an application library, which has the same name as the one already stored in ADONIS the user will see, **for each library**, (e.g. application library, BP library and WE library) the following advice (see fig. 225).
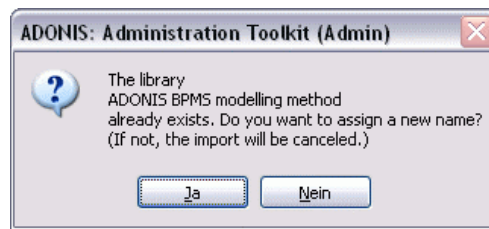
Figure 225: Import of an application library - name conflict

Click on the "Yes" button, to assign a new name to the imported library.

> **Hint:** If you click on the "No" button, the library import will be cancelled.

In the window "ADL import - rename library" (see fig. 226) the user will see the fields "Old name" and "New name". The text in the field "New name" is selected and can easily be changed.
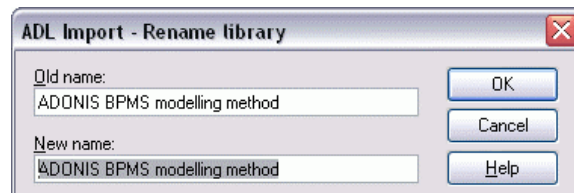
Figure 226: ADL import - rename library

Please change the name accordingly and then click the OK button to close the window and continue the library import.

2.3.2 Export application libraries

Exporting application libraries enables you to save your libraries in an ABL File (see chap. 9.1, p. 557) on a disk. This way you can import ADONIS application libraries into other ADONIS Version 3.9 databases and use them there. In addition, this feature serves for backup purposes.

Export an application library by opening the window "Application libraries" (see fig. 221) (menu "Libraries", option "Application library list"), select the application library to be exported and click on the button "Export".

The window "Library Export" will appear (see fig. 227).
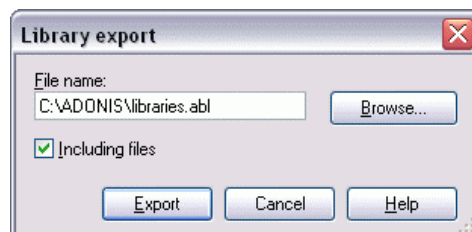
Figure 227: Library export

Enter the path and file name of the ABL file containing the application librariy to be exported in the field "File name" and click on the button "Export" to start the export process. Please note that the "Search" button can be used to enable you to select the location and filename. Activate the option **"Including files"**, if you also want to simultaneously export external files stored in the ADONIS database.

Hint: The external files will be exported as an ABL file together with the library definition. To export single external files, please use the export function from data management (see chap. 15., p. 606).

Finally, click on the button "Export", to start export of an application library.

When the application libraries have been exported successfully, an information window will inform you that the export process is finished.

Close the window by clicking on the "Close" button or by pressing the "Enter" key.

2.3.3 Delete application libraries

To delete an application library open the window "Library Administration - Application library list" (see fig. 221) (menu "Libraries", option "Application Library List"), select the application library you wish to delete and click on the button "Delete".

In the window "delete application libary" (see fig. 228) you can activate the options, which manage users and models that are connected to the application library.
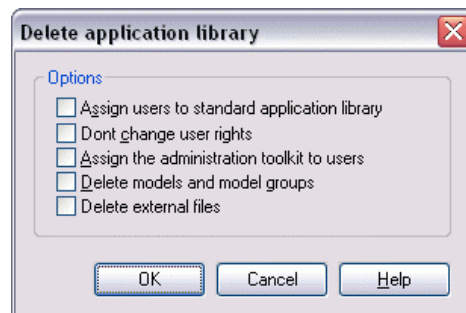
Figure 228: Delete application library - options

Option "Assign user to standard application library"
    This option allows to assign all the users, which were assigned to the deleted library to the ADONIS BPMS Application Library.

Option "Don't change user rights"
    This option allows user rights (see chap. 1.4.3.3, p. 93) to remain unchanged.

Option "Assign Administration Toolkit to users"
    This option allows the assigning of all users (coming from the deleted application library) access rights (see chap. 1.4.3.5, p. 94) to the Administration Toolkit.

    The option is available only when the option "Don't change user rights" is deactivated.

The Option "Delete models and model groups"
    Activation of this option means that all the models and model groups, which are based on the deleted application library, will be deleted.

The option "Delete external files"
    Activation of this option means, that all the external files stored in the ADONIS database and assigned to the deleted application library will also be deleted.

After activation of all the desired options, please click on the OK button to delete the earlier selected library. Then an information window appears, listing the name of the application library deleted.

    ATTENTION: The application library is deleted without a confirmation query! If the option "Delete models and model groups" is activated, these are also deleted without a security check! Therefore it is necessary to export (see chap. 3.5,

p. 325) all models still needed using the Model management (see chap. 3., p. 292) component before deleting the application library.

ATTENTION: If you activated the option "Delete external files", they will be **removed** from the ADONIS database **without** a confirmation!

Hint: The application library which is created during the installation of ADONIS can **not** be deleted!

2.3.4   Rename application libraries

In order to rename an application library, you should open the window "Library management - management" (see fig. 221) (Menu "Libraries", menu option "Management"), select the application library, which should be renamed and click on the button "Rename".

In the window "Rename libray" (see fig. 229) the current names of the application libraries are shown.
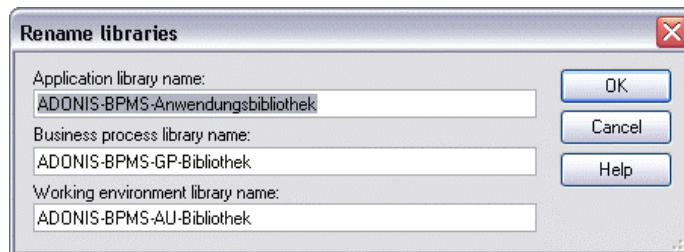
Figure 229: Rename library

Change the name and click on the OK button to close the window and allow the changes.

2.3.5   Administration queries

Using administration queries you can carry out queries concerning databases (user, user groups, model groups, and/or models).

In the window "<*Library name*> - Administration queries" (see fig. 230), the results of a query are displayed in the ADONIS browser.
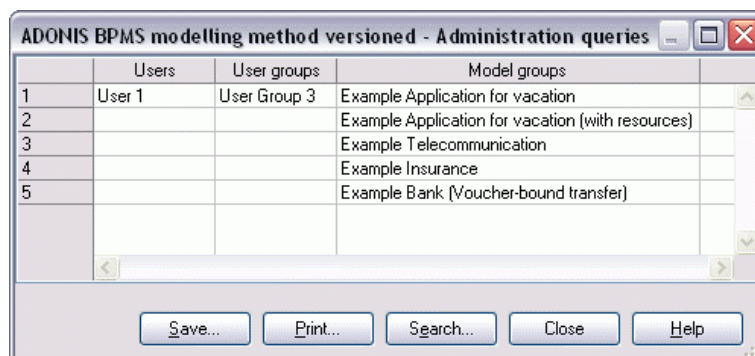
Figure 230: Administration query (example)

You can use the following functions from the context menu (right mouse button):

● Users (see chap. 1.4, p. 89)

● User groups (see chap. 1.5, p. 106)

- Model groups (see chap. 3.3.1, p. 295)

- Options (see chap. 2.3.5.1, p. 287)

In addition to this, the results of the query can also be saved, printed or copied into the deposit station for data transfer.

> **ATTENTION:** Upon performing very extensive queries (e.g. with result sets of more than one million lines), determining the results can take a considerable amount of time and during scrolling or column width adjustments, busy times can occur. Thus we recommend to reduce such queries to two or three colums and delimitate the result using the filter options provided.

2.3.5.1   Settings

In the window "Administration analysis - settings" (see fig. 231) the user can define query composition, i.e. which columns and with what kind of information should be displayed.
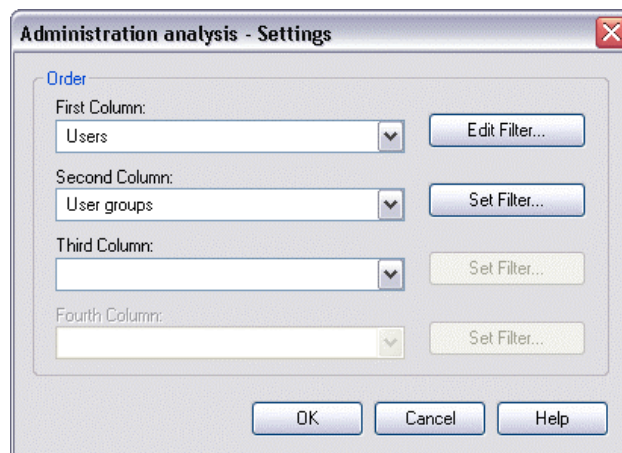
Figure 231: Administration queries - settings

If you click on the button "Filter", you can display the entries of each column by defining the seach term.

2.4   Import Migration Assistant

The import migration assistant supports you when transforming ADONIS Version 3.9 libraries and their corresponding models and users. In combination with the export migration assistant (see chap. 2.5, p. 289), you can transfer the desired data from one ADONIS database to another ADONIS database.

> **ATTENTION:** The migration of libraries, models and users from ADONIS Version 1.x or 2.x to ADONIS Version 3.9 is <u>not</u> possible.

> **Hint:** The functionality of the library, user and model import are already documented in earlier chapters. Thus we will refer to the appropriate sections.

To start the import migration assistant, select the menu option "Import migration assistant" from the "Migration" menu or click on the corresponding smart-icon in the quick-access panel.

The window "Import migration assistant" appears which shows the **tab "Library"**.
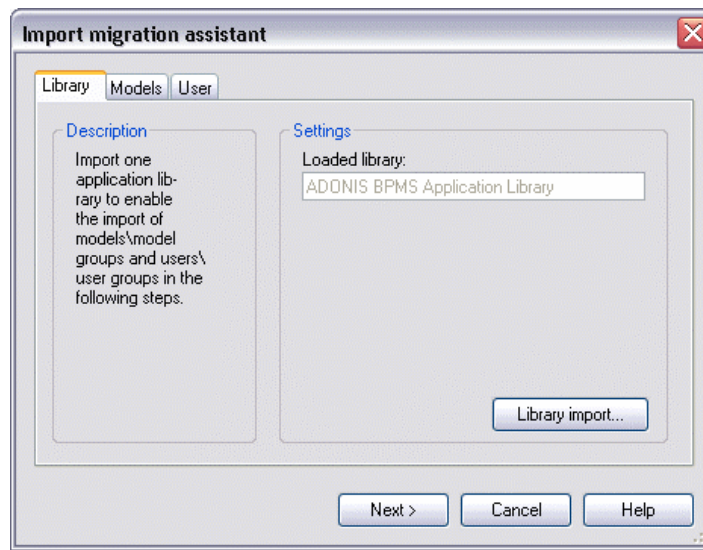
Figure 232: Import migration assistant - library

Click on the button "Library Import". The library import is documented in the chapter "Import Application Libraries" (see chap. 2.3.1, p. 282).

Once you have successfully imported the application library, you can import the ADONIS models related to it in the tab "Models" (see fig. 233).
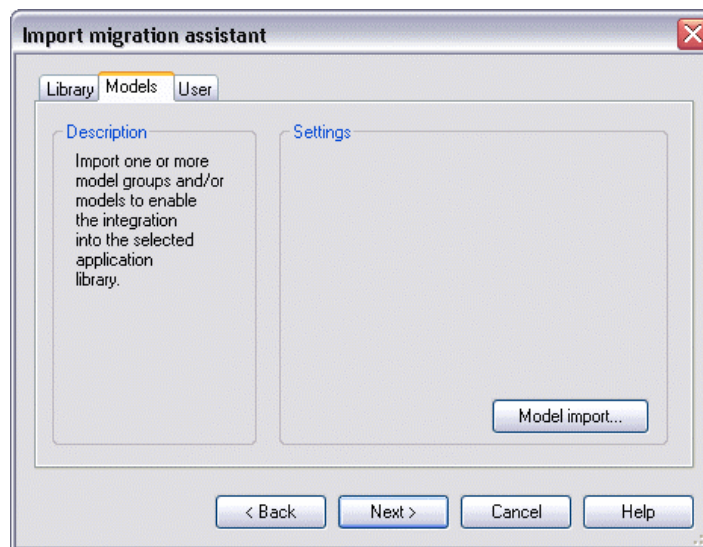
Figure 233: Import migration assistant - models

Click on the button "Model Import" to start the import of the models. Further information on the import functionality can be found in the chapter "Import Models" (see chap. 3.4.1, p. 302).

In addition to importing models, you can import ADONIS users in the tab "Users" (see fig. 234).
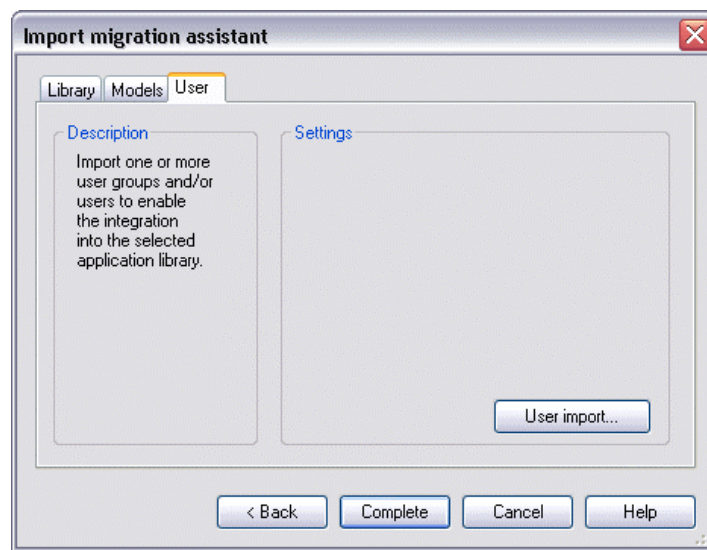
Figure 234: Import migration assistant - users

Click on the button "UDL Import". The import of users is documented in the chapter "Import Users" (see chap. 1.6, p. 116).

2.5 Export migration assistant

The export migration assistant supports you when transforming ADONIS Version 3.9 libraries and their corresponding models and users. In combination with the import migration assistant (see chap. 2.4, p. 287), you can transfer the necessary data from one ADONIS database into another ADONIS database.

ATTENTION: The migration of libraries, models and users from ADONIS Version 1.x or 2.x to ADONIS Version 3.9 is <u>not</u> possible.

Hint: The functionality of the library, user-and model import are documented in earlier chapters. We will thus refer to the appropriate chapters.

To start the export migration assistant, select the menu option "Export migration assistant" in the "Migration" menu or click on the respective smart-icon in the quick-access panel.

The window "Export migration assistant" which shows the tab "Library" is displayed.
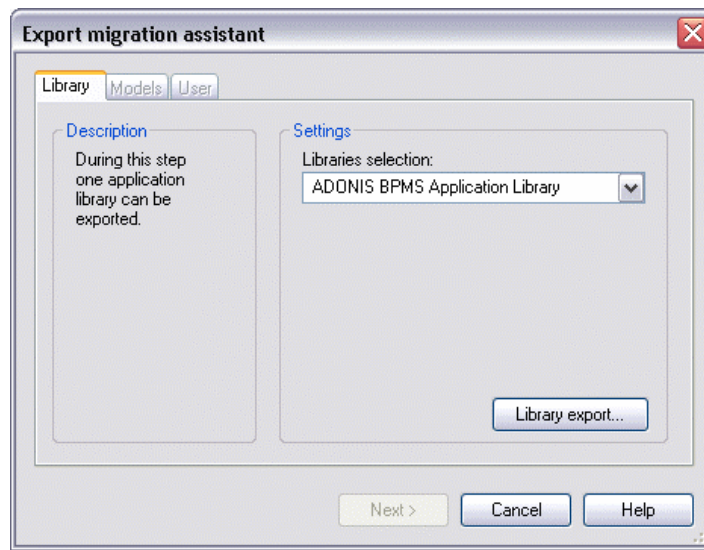
Figure 235: Export migration assistant - library

Select the application library to be migrated from the "Libraries" list and then click on the button "Library export". This functionality is documented in the chapter "Export application libraries" (see chap. 2.3.2, p. 284).

Once you have successfully exported the application library, you can export the ADONIS models based on it in the tab "Models" (see fig. 236).
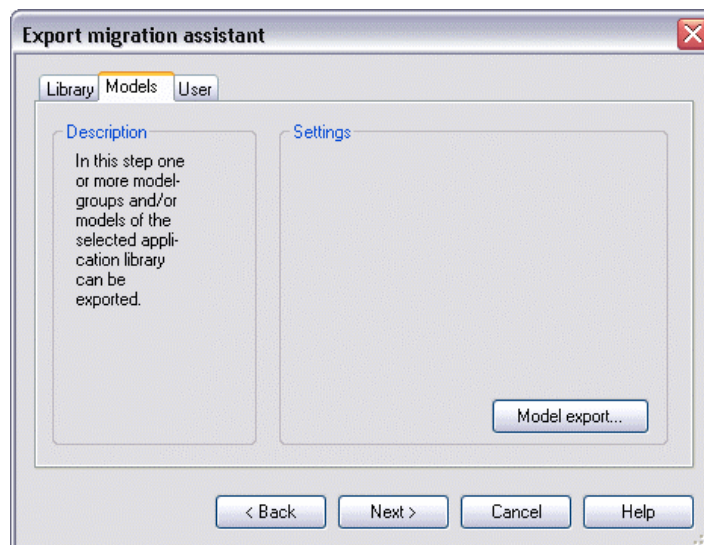
Figure 236: Export migration assistant - models

Start the export by clicking on the button "Model export". Further information on the export of models can be found in the chapter "Export models" (see chap. 3.5.1, p. 325).

In addition to exporting models, you can export ADONIS users in the tab "Users" (see fig. 237).
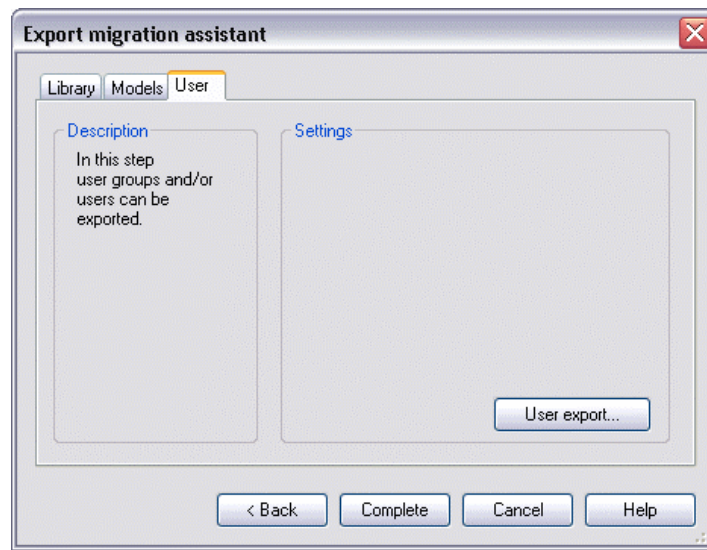
Figure 237: Export migration assistant - users

Click on the button "User export" to start the export. Further information on the export of users can be found in the chapter "Export users" (see chap. 1.7, p. 122).

3. Model Management

The grouping concepts in ADONIS permit the ADONIS models stored in the ADONIS database to be grouped in so-called ADONIS model groups. This can be compared to the hierarchy of directories in a file system.

In this chapter you will find the following descriptions concerning ADONIS model management:

● relations of Model Management in the Administration Toolkit (see chap. 3.1, p. 292)

● functionality available in the Model Management (see chap. 3.2, p. 294)

If you want to get access to services of model management, click on the Smart-Icon ![icon] from the quick-access bar.

Alternatively, you can activate the Model Management through opening the popup menu of the component bar (with the right mouse button clicking on the component bar, next to the Smart Icon) and choosing the menu option "Model Management". The popup menu can also be opened with the function key <F9>, and then model management can be activated with the function key <o>.

After activation of the model management, the quick-access bar with its Smart-Icons for model group management, ADL import, ADL export and delete models will be shown (see fig. 238).
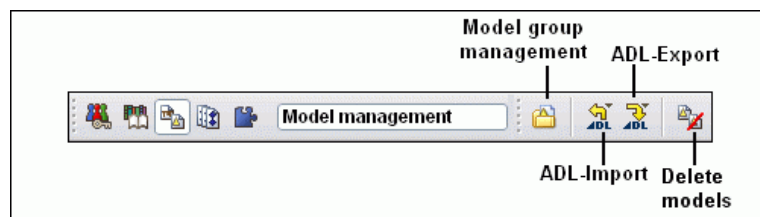
Figure 238: Model Management - components and quick-access bar

3.1 Relations of the Model Management

The below graphic (see fig. 239) should give you a general overvview of how model management is related with other parts of the ADONIS Administration Toolkit.
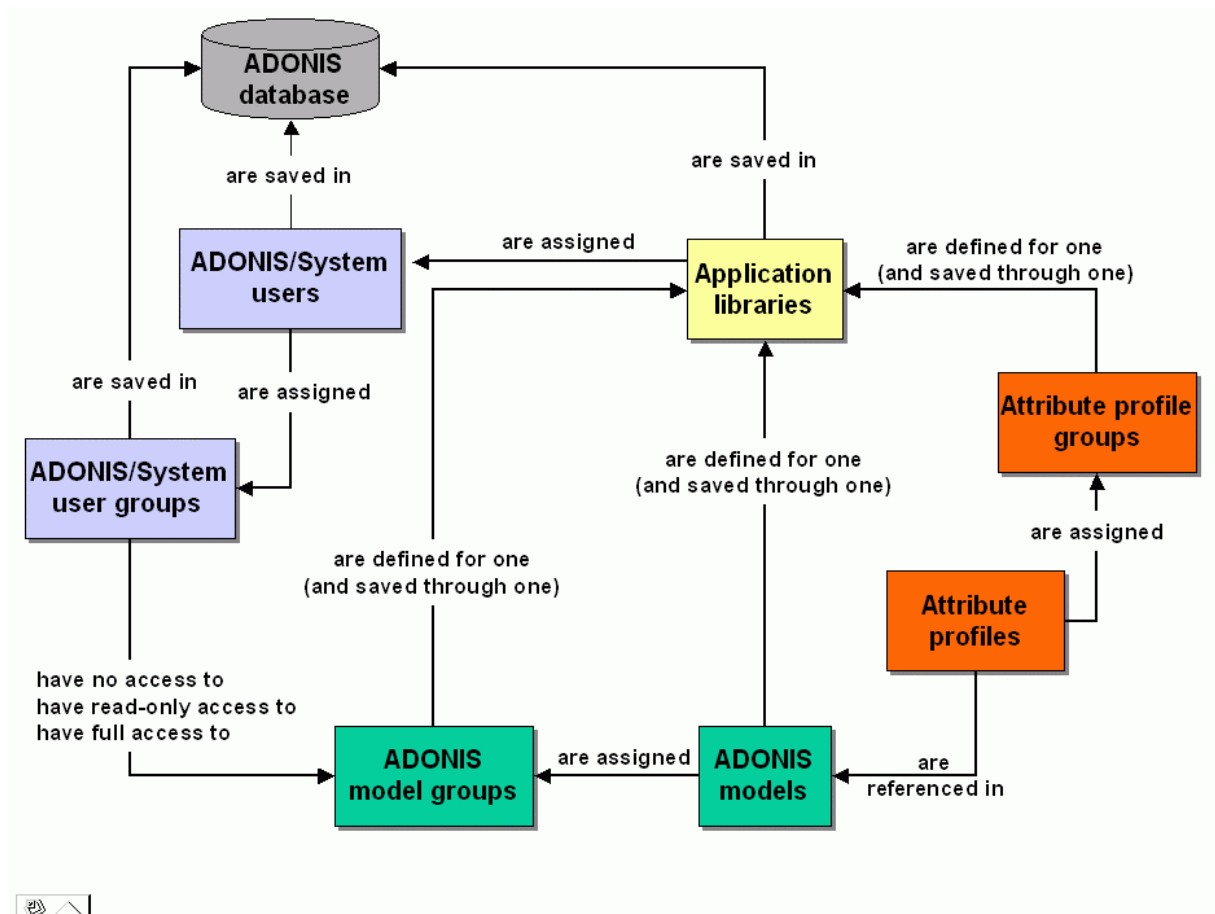
Figure 239: Overview of components in the Administration Toolkit

Each ADONIS model is based on an application library for which a model group hierarchy is also defined.

An ADONIS model stored in the ADONIS database can be edited by an ADONIS user, if

1. the application library assigned to this ADONIS user is the same library as that with which the model to be edited is based,

2. the ADONIS user is assigned to an ADONIS user group,

3. an ADONIS model group is defined for the application library on which the model to be edited is based,

4. the ADONIS model is assigned to this ADONIS model group and

5. this ADONIS user group has read and write access to this ADONIS model group.

An ADONIS user can create an ADONIS model, if

1. the ADONIS user is assigned to an ADONIS user group,

2. an ADONIS model group has been defined for the application library which has been assigned to the ADONIS user and

3. this ADONIS user group has read and write access to this ADONIS model group.

3.2 Functionality of the Model Management

The following functionality exists within the Model management component:

- Model group management (see chap. 3.3, p. 294) (add, rename, delete, assign access rights),

- Import (see chap. 3.4, p. 302) ADONIS models and ADONIS application models,

- Export (see chap. 3.5, p. 325) ADONIS models and ADONIS application models and

- Delete ADONIS models (see chap. 3.6, p. 330).

All the functions of the model management (see fig. 240) refer to a particular application library. Therefore, you must first select the desired application library from the list of all application libraries stored in the ADONIS database, before you can actually execute a function.
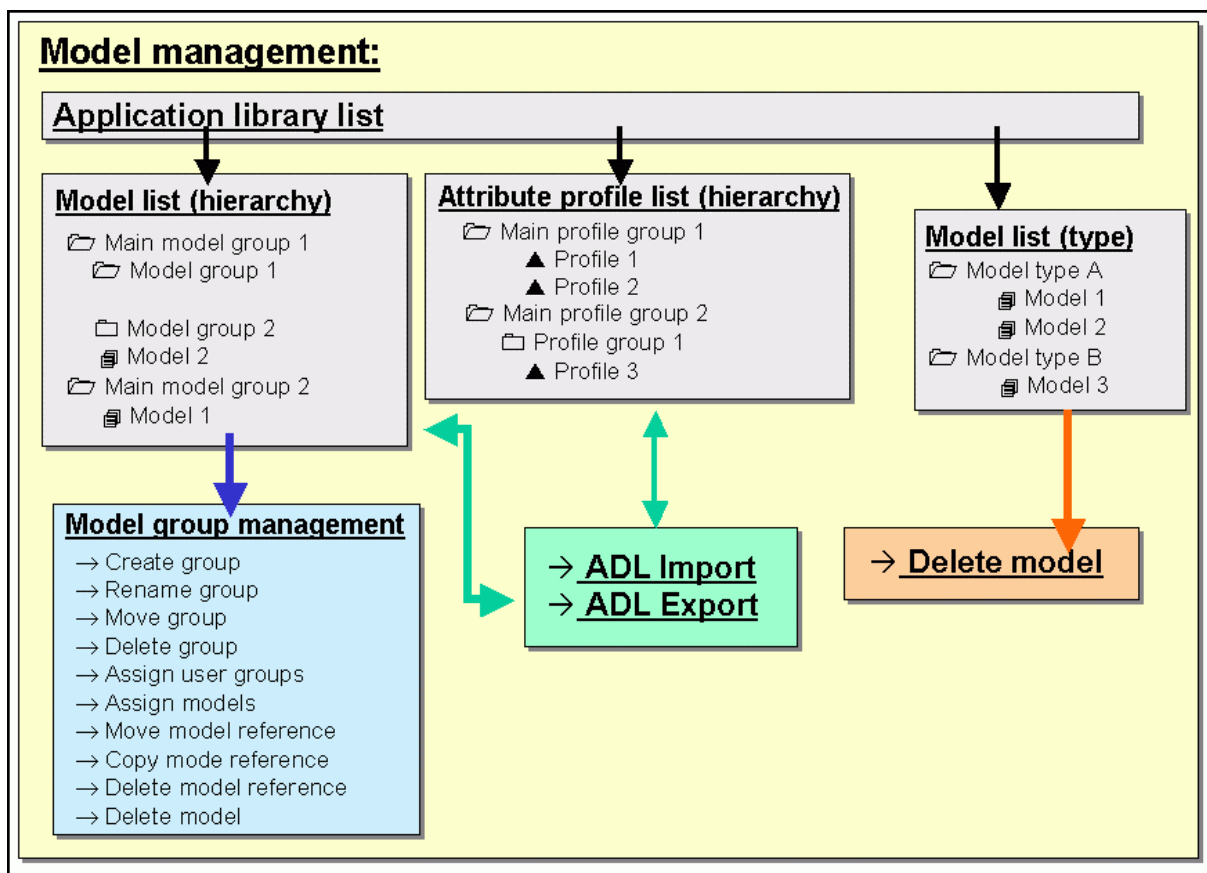
Figure 240: Functionality in Model Management

3.3 Model Group Management

The ADONIS models stored in the ADONIS database are referenced in an ADONIS model group. The ADONIS model groups establish the connection with the ADONIS user groups and are thus responsible for the ADONIS user's access rights to ADONIS models.

The list of model groups (see fig. 242) is the basis for the administration of the ADONIS model groups.

If you wish to carry out some administrative functions on model groups, select the menu option "Model group management" in the "Models" menu or click on the corresponding smart-icon 📂 in the quick-

access bar. The window "Model Management - Model group management" (see fig. 241) listing all the application libraries stored in the ADONIS database will be displayed.
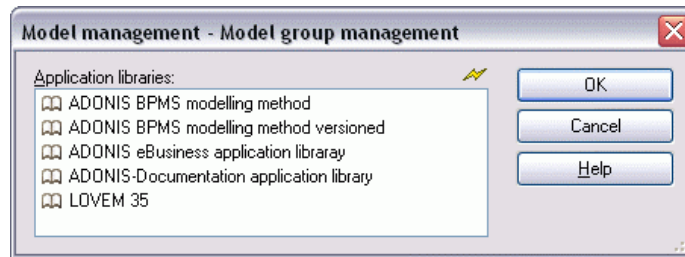
Figure 241: Model Management - Model Group Management

Select from the application libraries list the model groups you would like to edit and then click on the "OK" button.

The application library selected is now loaded and the window "<Application library name> - Model group list" (see fig. 242) appears.

3.3.1 Managing Model Groups

The model group hierarchy is displayed in the window "<*name of application library*> - model groups list" (see fig. 242).
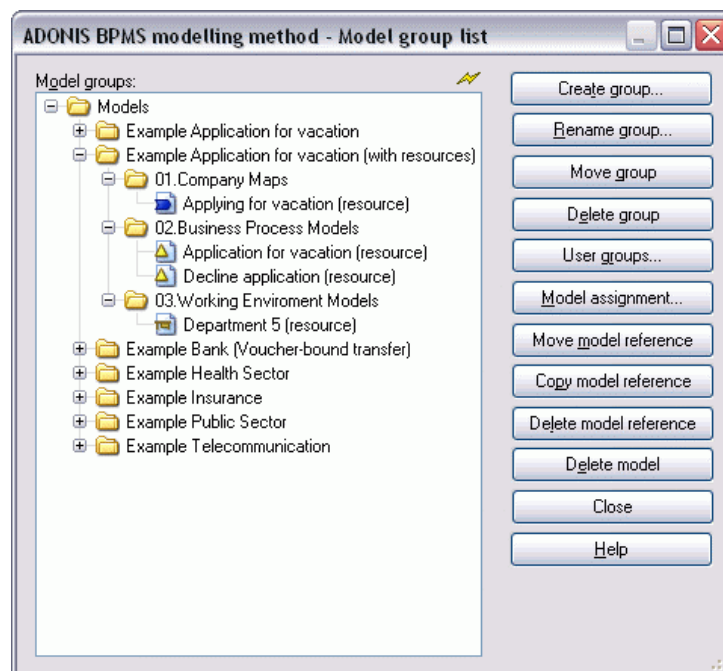
Figure 242: Model groups list

The list "Model groups" contains all existing model groups.

You can select the following functions by clicking on the appropriate button:

"Create group"   To create (see chap. 3.3.1.1, p. 296) a new model group.

"Rename group"   To rename (see chap. 3.3.1.2, p. 297) an existing model group.

"Move group"   To move (see chap. 3.3.1.3, p. 297) existing model groups.

"Delete group"	To delete (see chap. 3.3.1.5, p. 297) existing model groups.
"User assignment"	To assign (see chap. 3.3.1.6, p. 298) user groups to a model group (with read or write access).
"Model assignment"	To assign models (see chap. 3.3.1.7, p. 299) to a model group;
"Move model reference"	To move (see chap. 3.3.1.8, p. 300) model references from one model group to another.
"Copy model reference"	To copy (see chap. 3.3.1.9, p. 301) model references from one model group into anther.
"Delete model reference"	To delete (see chap. 3.3.1.10, p. 301) model references from a model group.
"Delete model"	To delete existing models.
"Refresh"	To update the model group list (database update).
"Close"	To close the window.

When you open the context menu (right mouse button), the general functions (see chap. 4.1, p. 34), the above listed functions of the model group list as well as the additional functions mentioned below are available:

"Model pool"	for displaying the models not assigned (see chap. 3.3.1.11, p. 301).
"Make group a main model group"	for moving the selected model groups to the top level (see chap. 3.3.1.4, p. 297).
"Model types"	for the model view of specific model types (see chap. 4.1.1, p. 34).

> **Hint:** The availability of the functions depends on whether or not you have previously se-
> lected either a model group or a model.

3.3.1.1   Create Model Group

Model groups can be created at every hierarchy level, i.e. you can create a new model group within any model group.

In order to create a new model group, select the model group in which the new model group should be contained and then click on the button "Create group".

> **Hint:** To create a model group as a main group, i.e. on the top hierarchy step, no model
> group shall be selected.

The window "Create model group" appears (see fig. 243).

Figure 243: Create model group

Enter the name of the new model group and click on the OK button. The window is closed and the up-dated model group hierarchy is displayed.

> **ATTENTION:**   The name of the new model group must be unique on the level of the main
> groups and within each model group.

3.3.1.2   Rename Model Group

If you wish to rename an existing model group, select the appropriate model group and click "Rename group". The window "Rename model group" (see fig. 244) appears, in which the current name (field "Old model group name") is shown. The field "New model group name" also holds the current name and you can adjust or over-write this with the new name you require.

Figure 244: Rename model group

Enter the new name of the model group and "OK". The window is closed and the updated model group hierarchy is displayed.

> **ATTENTION:**   The name of the new model group must be unique on the level of the main groups and within each model group.

3.3.1.3   Move Model Group

To move a model group, select the appropriate model group and then click on the button "Move group". As soon as you move the cursor over the list of the model hierarchy its shape will change to

.

Now click on the model group in which you would like to move the model group previously selected.

If you want to move a model group to the highest hierarchy level (to the main group), you must click on the area below model hierarchy, or alternatively on the left, next to the main models' groups.

> **Hint:** The name of the model group to be moved must be unique within the new model group and within the main group.

> **Hint:** When moving write-protected model groups to a model group with write access the write protection will be transferred. Should you wish to change the protection rights, must be carried out explicitly.

Before the group is moved you will be asked to confirm this action.

3.3.1.4   Move Model Group to Top Level

To move a model group to the top level (i.e. main model group), select the appropriate model group and then select "Make model group a main group" from the context menu.

Before the group is moved you will be asked to confirm this action.

3.3.1.5   Delete model group

> **Hint:** Model groups can be deleted only if they contain either other model groups or models.

Select one or more model groups which you want to delete and click "Delete group".

Before you finally delete the model group, an appropriate security message will be shown.

> **Hint:** When deleting a model group still containing other model groups and/or model refer-
> ences a corresponding message will be shown.

> **Hint:** When deleting the last model reference of a model while deleting a model group a cor-
> responding message will be shown, as this model will not be assigned to any model
> group of the application library and therefore cannot be accessed by any ADONIS user.
>
> Should you nevertheless choose to delete the last existing model reference it will only be
> possible to re-assign the model to a model group using the model pool, i.e. using the dia-
> logue "Show all models currently not assigned" (see chap. 3.3.1.11, p. 301).

3.3.1.6   Assign User Group

When assigning user groups to a model group, the access rights of the user groups to this model
group will be defined. The access rights will be defined as following:

No access   The user group has no access to the model group, i.e. the model group is
                           not visible for the users of this group.

Read access   The user group has read access to the model group, i.e. the users of this
                           group can open and read the models of this model group but can not edit
                           them.

                           It is **not** possible for the users to add a new model into this model group.

Read/write access   The user group has a write and read access to the model group, i.e. the
                           users of this group can open and edit the models of this group.

                           It is possible for the users to add a new model into this model group.

The access rights of user groups to a model group can be defined or changed by selecting the appro-
priate model group and then clicking "User group". The window "<Model group name> - User group
assignment" (see fig. 245) appears, which lists all the user groups saved in the ADONIS database.
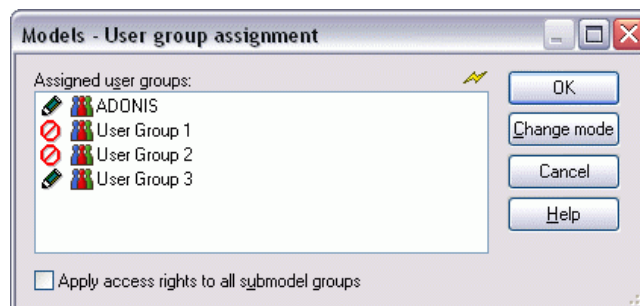
Figure 245: User group assignment

If your ADONIS database has been configured for the use of Single-Sign-on access rights, then it can
be defined for internal ADONIS user groups as well as for system user groups (see fig. 246). In this
case an additional icon will be shown in front of the name of a user group, indicating the type of the
user group. The icon 🏘 represents an internal ADONIS user group while the icon 🔧 indicates sys-
tem user groups.
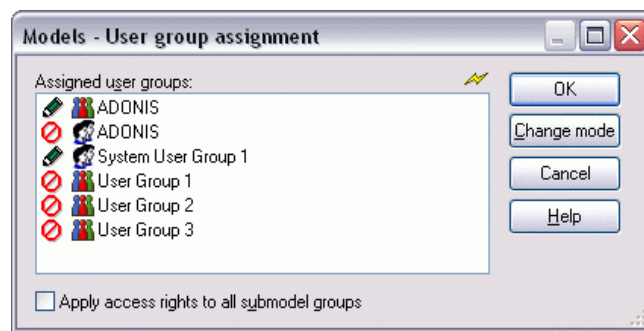
Figure 246: User groups assignment with system user groups

The icon before the name of the user group indicates the current right access. The icon

⊘   for "no access",

👓   for "read access" and

✏   for "write/read access"

✏   for changing access rights

is on the selected model groups.

Change the access rights by double-clicking on the user group. (The sequence for the change of access rights is "no access" -> "read access" -> "write/read access" -> "no access" etc.)

Alternatively you can change the access rights of several user groups in one step, by selecting the appropriate user group and clicking on the "Change" button to redefine the access rights in the above described sequence.

When activating the option **"Transfer rights to all submodel groups"**, the defined access rights will also be transfe^rred to the (sub)model groups contained in the previously selected model groups.

> **Hint:** The effects of the transfer of rights to submodel groups is described within the "Access rights to submodel groups" chapter (see p. 299) .

Once you have successfully changed the access rights, close the window by clicking on the OK button.

Access Rights to Submodel Groups

The access rights of user groups to submodel groups can like the higher model groups be expanded or shrunk.

This way it is possible to exactly define the appropriate access to models of a model group for the users from a user group.

> **Hint:** If you give a user group (write or read) access to a submodel group and this user group has no access to the high model groups, the user of this user group can exclusively have access to the submodel group. The models contained in the high model groups remain inaccessible to the users in the submodel group.

3.3.1.7   Assign Models

By assigning models to a model group, references to the models stored in the **ADONIS** database are created. The models assigned to a model group can be loaded with read-only or read-write access depending on the access authorisation of the user who attempts to open them.

If you wish to assign models stored in the **ADONIS** database to a model group, select the appropriate model group and click on the button "Model assignment". The window "<*model group name*> - Model assignment" (see fig. 247) is displayed, showing all models saved in the **ADONIS** database, organised according to model types.
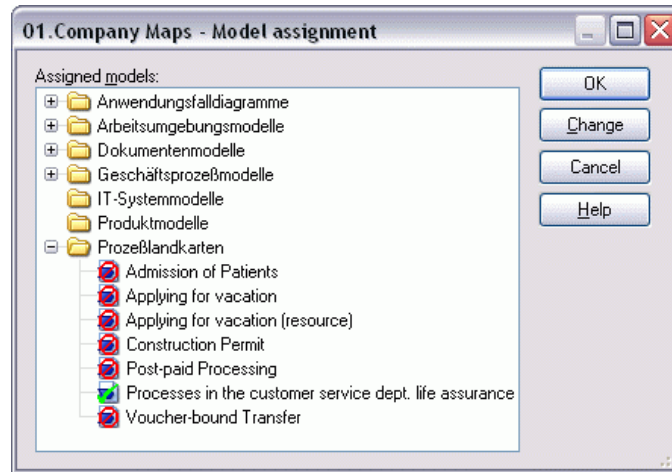
Figure 247: Model assignment

The icon before each model indicates the current status of the model with regard to the model group you have currently selected.

⊘   means "not assigned",

✔   means "assigned" and

✔   for "varying assignments".

Change the assignment by double-clicking on the respective model or by single clicking on the "access" or "no access" button.

> **Hint:** If you want to delete the last existing model reference from a model group, an appropriate message will be displayed, as this means that this model will no longer be assigned to any model group within the **ADONIS** database and thus cannot be accessed by any **ADONIS** users. However, unassigned models can be re-assigned by the **ADONIS** administrator.

After changing the model assignment as required (by double-clicking on the appropriate models to change the assignment mode or by selecting one or more models and then clicking the button "Change mode"), close the window by clicking on the OK button. The updated model hierarchy will be displayed.

3.3.1.8   Move Model Reference

If you wish to move a model reference to another model group, select the respective model reference and click on the button "Move model reference". As soon as you place the cursor over the list of the

model hierarchy its shape changes to: .

Now click on the model group to which you would like to move the previously selected model reference.

> **Hint:** A model group may only contain one model reference to a model stored in the **ADONIS** database. You cannot move a model reference into a model group in which such a reference already exists.

The system will ask you to confirm this action before the model reference is actually moved.

3.3.1.9 Copy Model Reference

If you wish to copy a model reference to another model group, select the appropriate model reference and click on the button "Copy model reference". As soon as you place the cursor over the model hier-

archy list its shape will change to .

Now click on the model group into which you want to copy the previously selected model reference.

> **Hint:** A model group may only contain one model reference to a model stored in the **ADONIS** database. You cannot copy a model reference into a model group in which such a reference already exists.

The system will ask you to confirm the action before the model reference is actually copied.

3.3.1.10 Delete Model Reference

To delete a model reference from a model group, select the model reference and then click on the button "Delete model reference".

The model reference will be deleted after a security message.

> **Hint:** When deleting the last model reference of a model while deleting a model group a corresponding message will be shown, as this model will not be assigned to any model group of the application library and therefore cannot be accessed by any ADONIS user.
>
> Should you choose to delete the last existing model reference it will only be possible to re-assign the model to a model group using the model pool, i.e. using the dialogue "Show models not assigned" (see chap. 3.3.1.11, p. 301).

3.3.1.11 Show Models not assigned

These models are not assigned to a model group and so are not visible for ADONIS users.

To find out which models are not assigned , activate the "model pool" display by selecting the menu item "Model pool" in the context menu (right mouse button). In the window "*<application library name>* - Model group list" (see fig. 248), the field "Model not assigned" is shrunk.
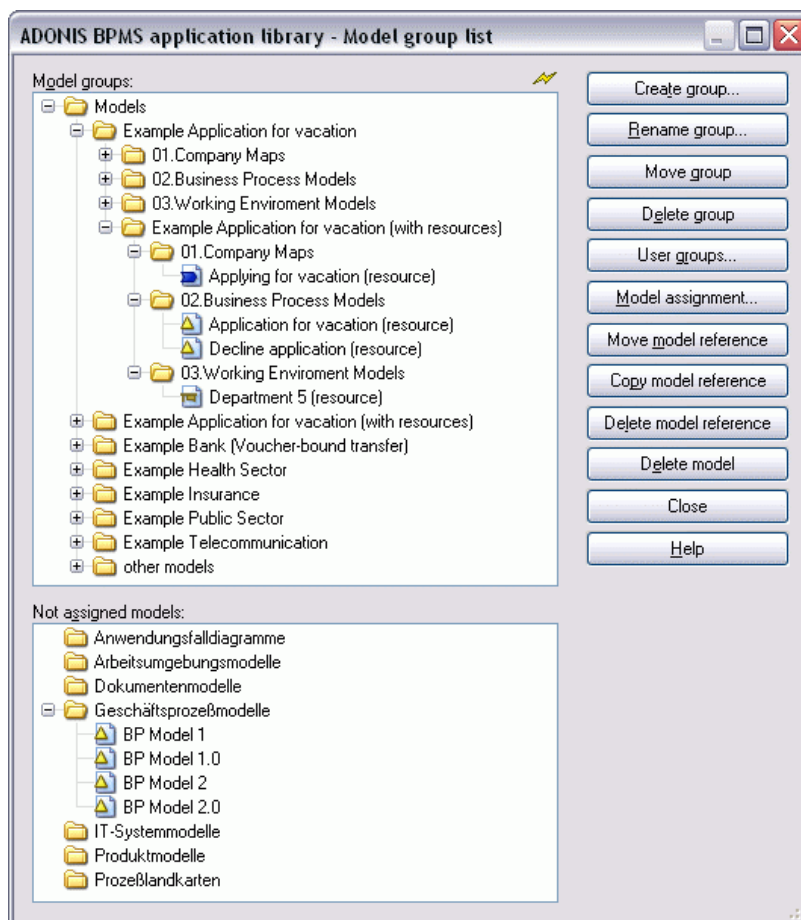
Figure 248: Show models not assigned

You can assign the models not assigned to a model group using the function "Move model references" (see chap. 3.3.1.8, p. 300) or completely delete them from the ADONIS database by clicking on the "Delete model" function.

3.4 ADL Import

The ADL import function allows the integration of data (models, attribute profiles, application models ...) in one ADONIS database. The data to be imported must be saved using the ADL Export (see chap. 3.5, p. 325) function prior to import.

ADL import is possible not only for **Models/Model groups and attribute profiles/attribute profile groups** (see chap. 3.4.1, p. 302) but also for **application models** (see chap. 3.4.5, p. 321).

3.4.1 Model Import

This function allows for the import of models, model groups, attribute profile and attribute profile groups. This process contains more steps:

● Start import

● Establish import settings and options

● Establish the aims of import and things to be imported

● Import

Start ADL Import of models/attribute profiles:

It is possible to start import in three ways:

● Click *shortly* on the smart-icon,

● Click and *hold on* the smart-icon and choose item from the drop-down menu **"models/attribute profiles"**,

● Choose **"models"** from a menu item **"ADL Import"** sub menu item **"models/Attribute profiles"**.

The Window "Model Management - ADL Import" showing all ADONIS application libraries saved in database will be displayed:
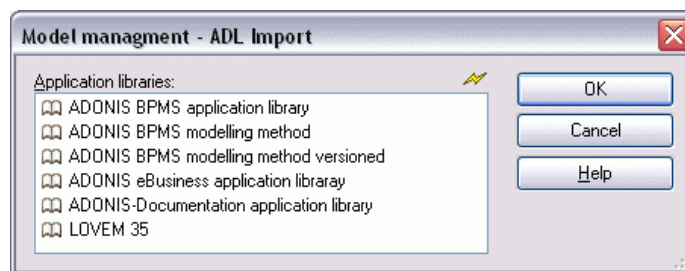
Figure 249: Model management - ADL Import

Choose the application library which contains the models or attribute profiles to be exported and then click on the **"OK"** button. The chosen library will be shown in the "ADL Import - Settings" window:
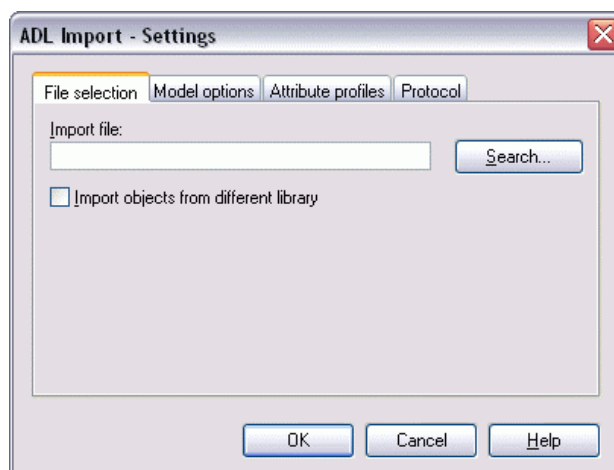
Figure 250: ADL Import

This dialogue consists of four parts, which can be identified with tabs.

ADL Import - Settings

In the window "ADL Import - Settings" you can define settings concerning import in the tabs "File selection", "Model options", "Attribute profiles" and "Protocol".

 Hint: The tab "Versioning" is only available after an appropriate definition in the application library (Customising) .

Tab "File selection":
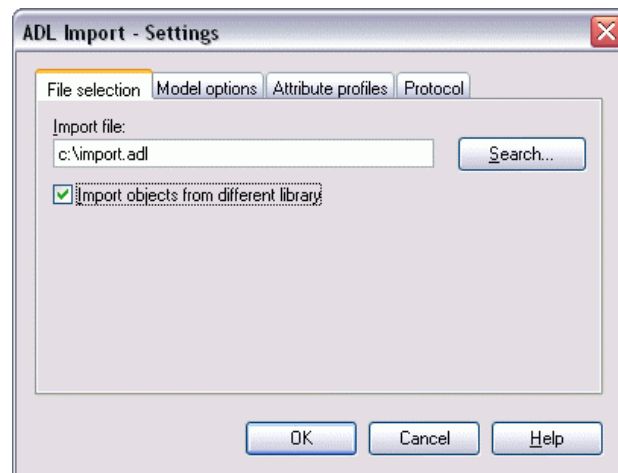
Figure 251: ADL Import - File selection

Enter the path and name of the ADL file with the model you wish to import into the **"File selection"** tab. The Button **"Search"** opens a window for the user to searching for the location of the file to be imported.

The **option "Import object from different library"** is available to import models which are based on a different application library than that previously selected.

> **ATTENTION:** The import of objects from different libraries is reasonable only if most of the classes and relations are the same in both libraries. This is because only the classes and relations which exist in the library assigned to the user (current library) are imported. Similarly, only attributes which appear in the user library (current library) will be imported. Attributes that are in the current library and are missing in the objects of models to be imported will be added and filled in with default values of the current library. Attributes which exist in objects of the models to be imported and are not available in the current library will not be imported.

> **Hint:** If you do not choose this option, it will not be possible to import external models. An information window will appear and import will be cancelled. If you are ussure in which application library your models are based, choose this option to successfully import the models.

Tab "Model options":

Here you can find settings for the import of models or model groups:
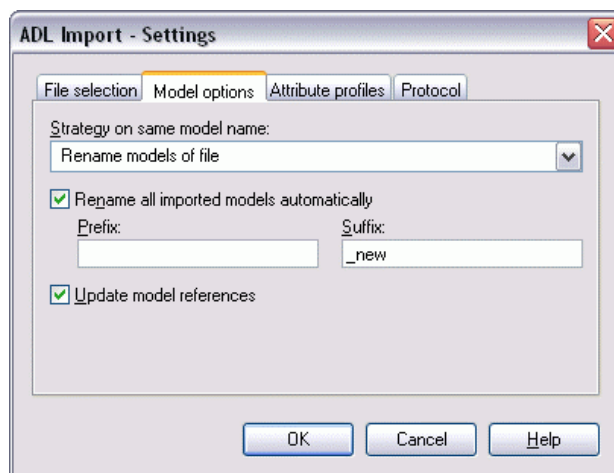
Figure 252: ADL Import - Model options

To import models, which have the *same name as the models stored in the ADONIS database*, you can select one of the following options from the list "Strategy on same model name":

- "Rename models of file" (see chap. 3.4.4.2, p. 312)

- "Paste into existing models" (see chap. 3.4.4.3, p. 313)

- "Overwrite existing models" (see chap. 3.4.4.4, p. 317)

- "Ignore models of file" (see chap. 3.4.4.5, p. 318)

- "Increase version number" (see chap. 3.4.4.6, p. 318)

The **option "Rename all imported models automatically"** assigns a prefix to the name of all imported models and/or a suffix. In the case that a name generated already exists in the database it will be processed according to the configuration of the option "Strategy on same model name".

> **Hint:** If you have chosen "Rename models of file" or the option "Rename all imported models automatically" and assigned a prefix/suffix, then it is recommended to also select the option "Update model references".

The **option "Update model references"** is available for all options and should always be selected, when the imported models are renamed. By updating the model references the references to imported models are automatically updated in case one of the imported models is being renamed.

> **Hint:** The strategy "Increase version number" as well as the additional action "Include version number in the model names" are available only for ADL Import to an application library with time-related versioning (see chap. 6.1.2, p. 69).

"Attribute profiles" tab:

In the "Attribute profiles" tab you can select one of the options for the import of attribute profiles:
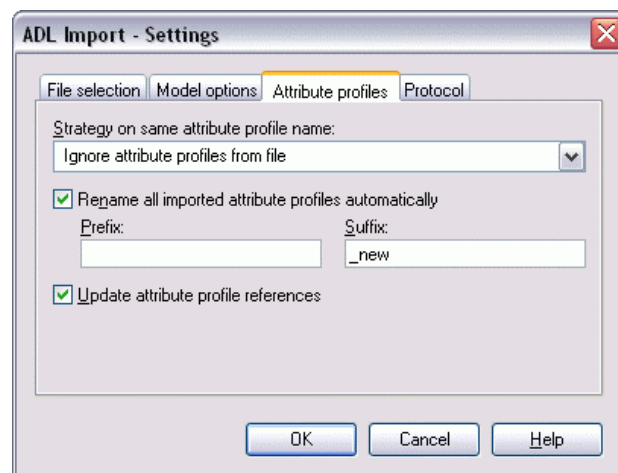
Figure 253: ADL Import - Attribute profiles

To import attribute profiles which have the *same name as the models stored in the ADONIS database*, you can select one of the following options from the list **"Strategy on same attribute profile"**:

- "Rename attribute profiles from ADL file" (see chap. 3.4.4.7, p. 318)

- "Paste into existing attribute profiles" (see chap. 3.4.4.8, p. 319)

- "Overwrite existing attribute profiles" (see chap. 3.4.4.9, p. 320)

- "Ignore attribute profiles from ADL file" (see chap. 3.4.4.10, p. 320)

- "Increase version number" (see chap. 3.4.4.6, p. 318)

The **option "Rename all imported attribute profiles automatically"** allows assigning a prefix to the name of all imported attribute profiles (it will be put in front of the attribute profile name) and/or a suffix (it will be put behind the attribute profile name). In the case that a name generated already exists in the database it will be processed according to the configuration of the option "Strategy on same attribute profile name".

> **Hint:** If you have chosen as a strategy "Rename attribute profile from the file" or the option "Rename all imported attribute profiles automatically" and assigned a prefix/suffix, then it is recommended to select the option "Update attribute profile references" as well.

The **option "Update attribute profile references"** is available for all attribute profile options and should always be selected when the imported attribute profiles are renamed. By updating the attribute profile references, the references to imported attribute profiles are automatically updated in case one of the imported attribute profile is being renamed.

Tab "Protocol":

It will enable you to generate a protocol for the ADL Import:
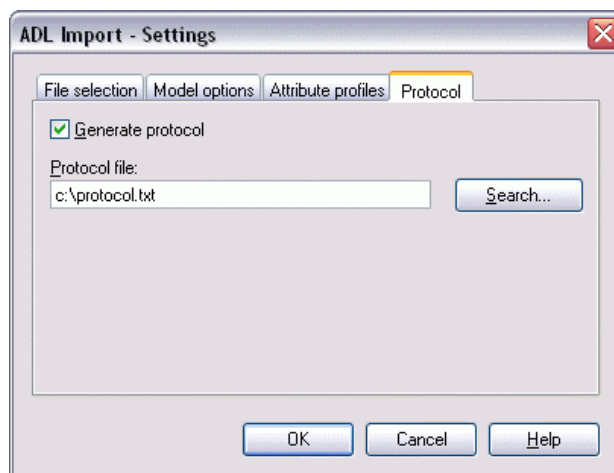
Figure 254: ADL Import - Protocol

Activate the option **"Generate protocol"** and enter into the **"Protocol file"** field the path and the name for the import protocol (TXT file). The button **"Search"** opens a window for the user to locate a file to be chosen.

After selecting all the options, click on **"OK"**, to start ADL Import of models or attribute profiles.

3.4.2   Selection

After selecting an ADL file and determining the required settings (see chap. 3.4.1, p. 302) the imported data is read; All the contained models (see fig. 255) and/or attribute profiles (see fig. 257) will be shown in the **"ADL Import - selection"** window.

> **Hint:** The display of tabs "Models" and "Attribute profiles" depends on the input of the ADL file, i.e. only models in tab "Models", or only attribute profiles in tab "Attribute profiles" can be shown, exclusively one of two tabs can be available.

Tab "Models":

 Models contained in an ADL file will be shown here:
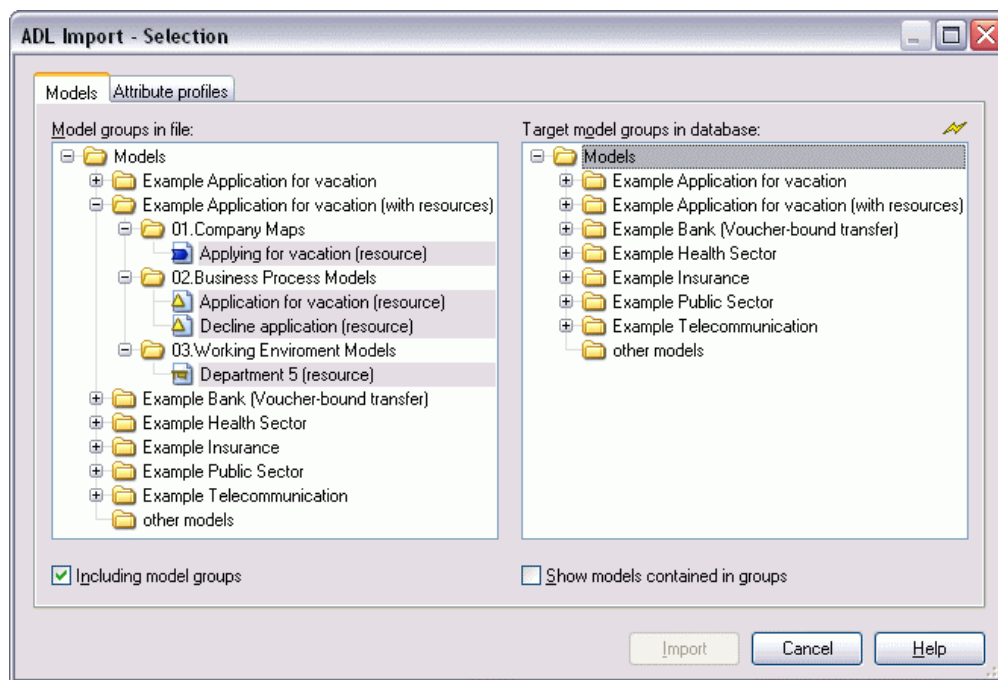
Figure 255: ADL Import - Model selection including model groups

If the option **"Including model groups"** is activated in the "Models" tab, the models contained in the ADL file will be displayed according to their model hierarchy (see fig. 255). When this option is selected the model group hierarchy will be imported as well as the selected models.

> **Hint:** If no models are selected to be imported, only the model group hierarchy will be imported.

If you deactivate the option **"Including model groups"**, the models contained in the ADL files are displayed according to their model type. Only the selected models will be imported while using ADL Import:
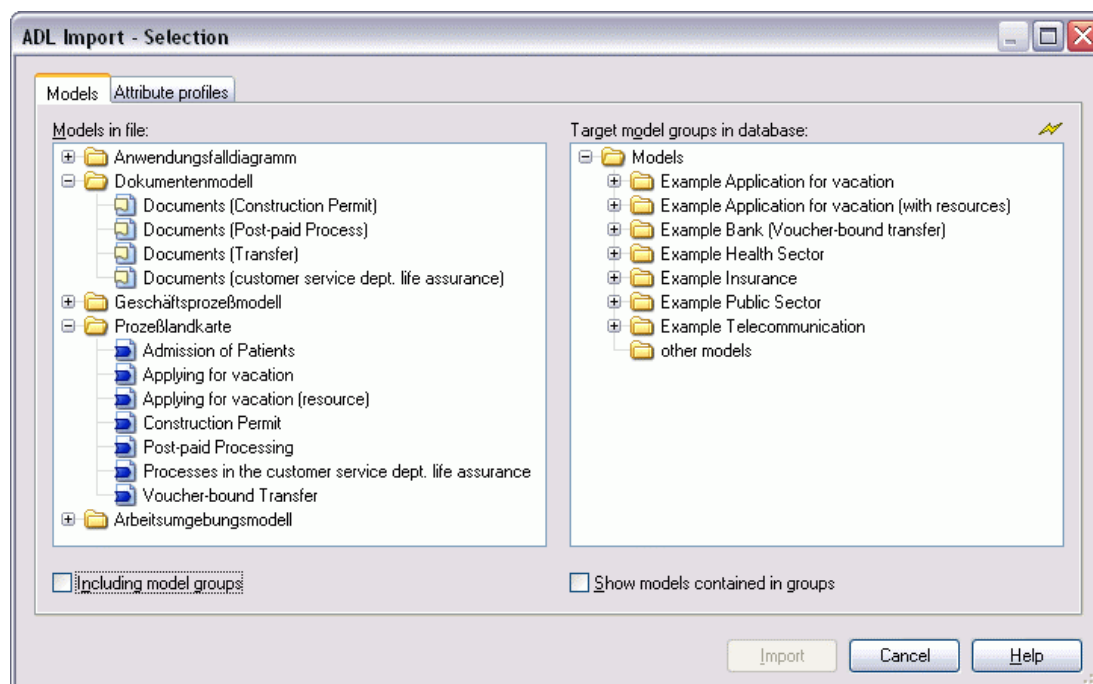
Figure 256: ADL Import - Model selection including model groups

Select the option **"Show models in groups"** to see lists of models contained in all model groups.

To choose imported models: Select all the models you want to import along with the model groups which you want to create references to the imported models.

<u>**Tab "Attribute profiles":**</u>

All the attribute profiles contained in the ADL file will be displayed here:
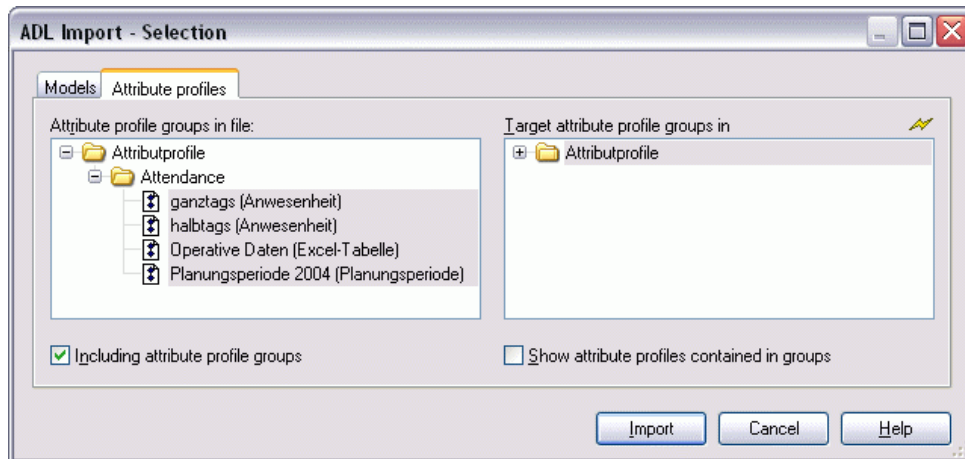
Figure 257: ADL Import - Attribute profile selection including attribute profile groups

If you *activate* the option **"Including attribute profile groups"**, the attribute profiles contained in the ADL file will be displayed according to their hierarchy (see fig. 257). When this option is selected the attribute profile group hierarchy will be imported as well as the selected attribute profiles.

> **Hint:** If no attribute profiles are selected to be imported, only the attribute profile group hierarchy will be imported.

If you *deactivate* the option **"Including attribute profile groups"** in the tab "Attribute profiles", the attribute profiles contained in the ADL file will be displayed with no structure. Only the attribute profiles selected will be imported during the ADL Import:
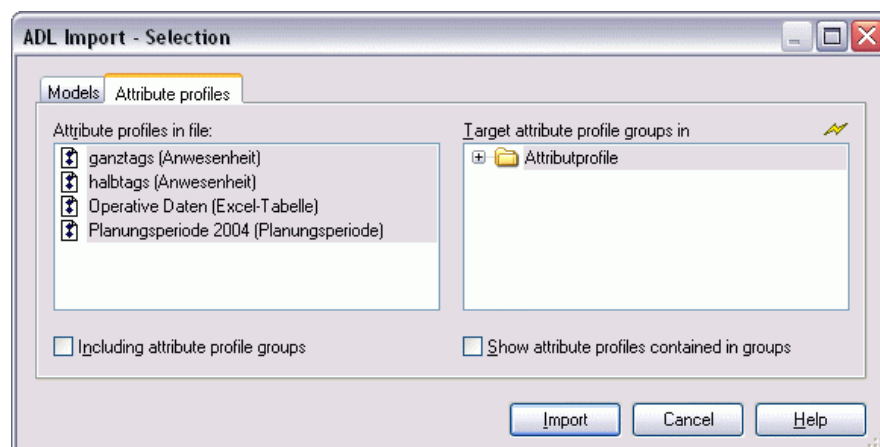
Figure 258: ADL Import - Attribute profile selection excluding attribute profile groups

When you select the option **"Show attribute profiles contained in groups"** the attribute profiles contained in the ADONIS Database will be displayed.

To choose import profiles: Select all the attribute profiles you want to import and the attribute profile groups in which you want to create a reference to the imported profile.

<u>**To import:**</u>

After choosing model and attribute profiles click on **"Import"**, to start the ADL Import.

> **Hint:** If you don't select any destination model group, the main model group in the ADL file (the uppermost hierarchy level) will be included in the main model group in the ADONIS database.
>
> During the import excluding model groups, the selected models will not be assigned to a model group and so no ADONIS user will be entitled to open them. Before the import, a security message will inform you about this.

> **Hint:** If you import an ADL file with its attribute profile groups and you don't select any destination attribute profile group, the main attribute profile group in the ADONIS database (the uppermost hierarchy level) will be used as a destination group.
>
> During the import excluding attribute profile groups, the import of selected attribute profiles is not possible, if previously no destination attribute profile group has been selected.

At the beginning of the import, the ADONIS database is first checked for models or attributes which have the same name as the models or attributes to be imported.

> **Hint:** Only the name of the models or of the attribute profiles to be imported is compared to the models or attribute profiles stored in the ADONIS database, **not** their contents. Therefore, an imported model or attribute profile may have the same content as one stored in the ADONIS database, but under a different name. This means it is very important to select the appropriate options as listed in the window "ADL Import - Settings" (see fig. 252) for importing models (see chap. 3.4.1, p. 302) with identical names or in the window "ADL Import - Settings (see fig. 253)for importing attribute profiles (see chap. 3.4.1, p. 302) with identical names.

A status window keeps you informed about the current status of the import.

> **Hint:** During the import, if the ADL file contains a model of a type which is not defined in the current application library and the option "Import models from different library" is activated in the window "ADL Import - Settings" (see fig. 250), it is possible to assign a model type (see chap. 3.4.2.1, p. 310) defined in the application library to this model .

3.4.2.1   Assign model type

If a model's type as specified in the ADL file being imported does not exist in the current application library, then the window "ADL Import - Assign model type" is displayed:
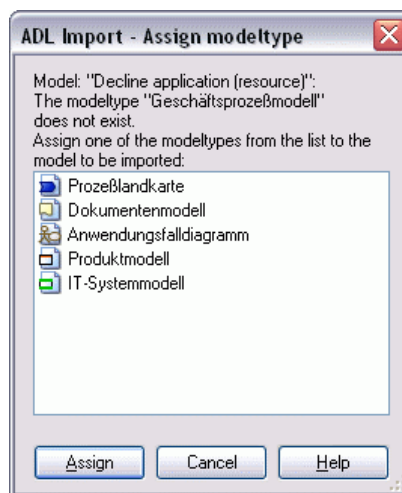
Figure 259: ADL Import - Assign model type

Select one of the suggested model types and assign it to the model being imported by clicking on the **"Assign"** button. The ADL Import will continue.

3.4.3   Result

After the ADL Import has been successfully completed, the window "ADL Import - Results" will inform you about the conclusion of the import. This window lists the names under which the imported models are stored in the ADONIS database:
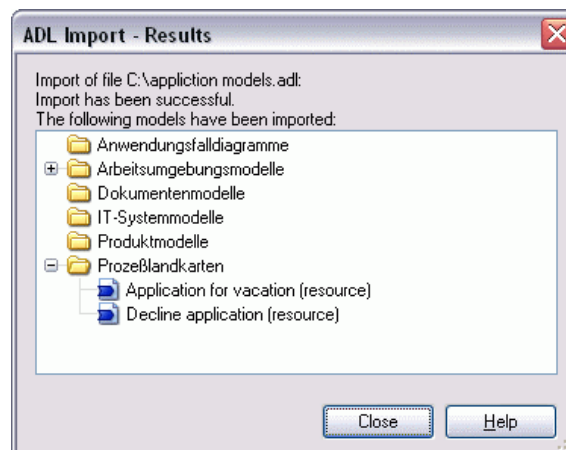
Figure 260: ADL Import - Result display (with protocol)

If in the window "ADL- Import - Settings" - Tab "Protocol" (see fig. 251) the option "Generate protocol" is active and a path and the name for the protocol file to be created is provided, then the button "Protocol" will be shown in the "ADL Import - Results" (see fig. 260) window. By clicking on this button the generated protocol will be shown in a text editor.

   Hint: If you have imported exclusively model groups, the successful ADL Import will be indicated through an appropriate message window.

3.4.4 Options and examples

> **Hint:** This chapter lists all ADL and XML import options as well as examples. Please note that not each option is available for both functions.

3.4.4.1 General options for ADL import

Activate in the tab "model options" the option **"Rename all imported models automatically"** and enter a prefix (field **"Prefix"**) and/or a suffix (field **"Suffix"**) that will be added to the model name during import.

Activate in the tab "model options" the option **"Update model references"** to ensure that all references within the model you wish to import will be updated according to the change of name.

<u>**Option when using an application library with time-related versioning (see chap. 6.1.2, p. 69):**</u>

Activate in the tab "model options" the option **"Take over version numbers of model names"** to inherit model-oriented version numbers contained in the model names (see chap. 6.1.1, p. 69) which now would have time-related versioning. As a result each imported models will gain its own version order.

3.4.4.2 Rename models from ADL file

The model to be imported is renamed, if a model of the same name and model type is already stored in the ADONIS database.

> **Hint:** Activate in the window "ADL Import - Settings" the option **"Update model references"** to ensure that all references within the model you wish to import will be updated according to the change of name.

> **Hint:** Activate in the window "ADL Import - Settings" the option **"Take over version number in the model name"** to take over the model-related version number (see chap. 6.1.1, p. 69) of a model to the name of a model in a time-related version. This way each imported model will have it's own version thread created.
>
> The option "Take over the version number in the model name" is available only for ADL Import to an application library with time-related versioning (see chap. 6.1.2, p. 69) .

When the name and model type are the same as of a model in the database, a window will open with the question of whether a new name should be assigned.

> **Hint:** If the model with the same name stored in the database is not shown, it means that it is not assigned to a model group or it is referenced to a model group to which the user has neither read nor write access.

If you click on **"Yes"**, a window **"Rename model"** will be displayed:
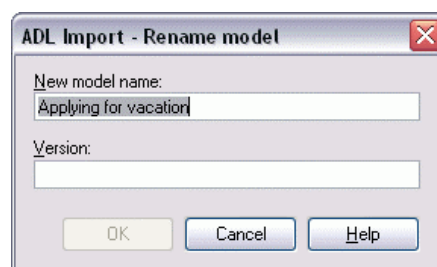
Figure 261: Import - Rename model

Enter a new name for the model to be imported and click on "OK" or press the Enter key to start importing.

> **Hint:** Using the **"Cancel"** option in the window "Rename model" causes the import to abort as in the ADONIS database, only models with unique names can be stored.

3.4.4.3   To paste into existing models

Models to be imported will be aggregated in the ADONIS database with all models which have already been stored there. In case of objects with the same names and connectors, attribute values of the imported objects or connectors will be taken over. Other objects and connectors of the imported models will be newly created.
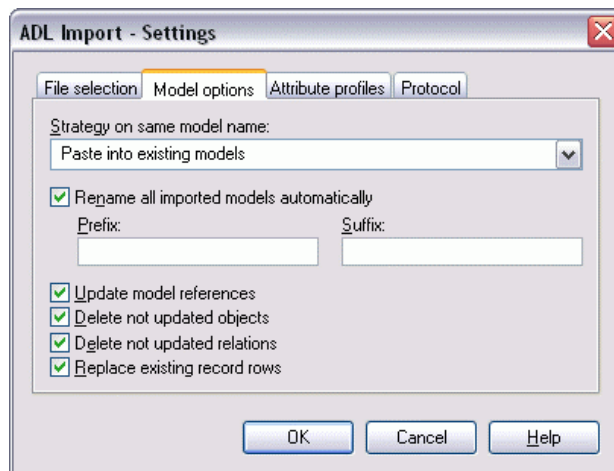
Figure 262: ADL import settings - extended model options

When the option **"replace existing record rows"** is *active*, the existing rows in record attributes will be cancelled and replaced with rows from the models to be imported. If the option is *deactivated*, the rows of record attributes in the attribute profiles to be imported will be appended to the lines in the existing attribute profiles. The effects are shown in an example (see p. 314) .

If you *activate* the options **"Delete not updated objects"** or the **"Delete not updated relations"**, all objects/connectors in the existing model will be deleted, if the objects/connectors with matching names are not contained in the model to be imported. The effects are illustrated in an example (see p. 315). Objects, which are stored in the ADONIS database and are not contained in the model to be imported, are left unchanged in the model, if the "Delete not updated objects" option is *deactivated*. A connector and its attribute values are transferred, if the objects and their links have matching names and types in the model stored as well as in the model to be imported.

If there are objects or connectors with matching names in both the model to be imported and the existing model, the window **"ADL Import - replace values"** will be displayed for each object (see fig. 263) or for each connector (see fig. 264).
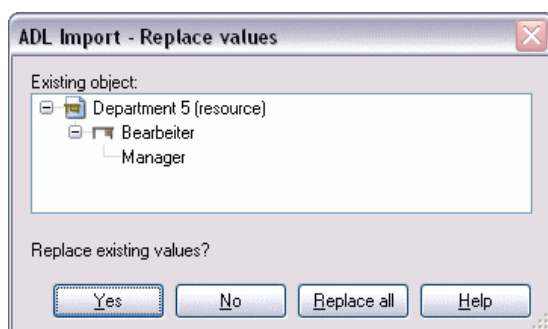
Figure 263: ADL Import - Replace value of an object
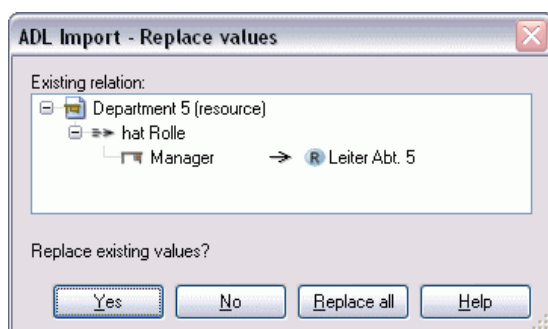
Figure 264: ADL Import - Replace value of a connector

Click **"Yes"** if you want to replace the existing attribute values with the values of the object or connector in the import model. By clicking **"Replace All"**, all attribute values of objects and connectors with matching names will be replaced by the values of the imported model. Clicking **"No"** means that the existing attributes of the object being queried will not be changed.

Example of the option "Replace existing record rows"

The effects of the option "Replace existing record rows" is shown in the picture below (see fig. 265).

Assumption:

A model "Example model 1" of the type "BP model" stored in the database contains the object "Activity 1" with an attribute of type record. The values of the attributes are shown in the picture below (see fig. 265).

Figure 265: Attribute value of the model saved in the database

A model "Example model 1" of the type "BP model" is stored in an ADL file. This model contains the object "Activity 1" with an attribute of type Record. The values of this attribute (see fig. 266) are different from those stored in the database.

Figure 266: Attribute value of the model contained in the file

<u>**Results:**</u>

When the option **"Replace existing record rows" is activated**, the entries in the table in the model stored in the database will be deleted and the values will be transferred from the model to be imported. The result of the ADL Import therefore equals the values of the attribute from the model in the ADL file (see fig. 266).

When the option **"Replace existing record rows" is deactivated**, the entries remain in the record in the model stored in the database and the values from the model to be imported will be appended. The result of the ADL Import is shown below:

Figure 267: Results of the deactivated option "Replace existing record rows"

ATTENTION: Record attributes can be limited to a certain number of rows (entries). Should the maximum number of entries be exceeded while having the option "Replace existing record rows" disabled, the import will be aborted.

Example of the option "Delete not updated objects/connectors"

The effects of the option "Delete not-updated objects/connectors" are illustrated in the following example.

<u>**Assumption:**</u>

A model "Example model 1" of type "WE model" stored in the database contains the objects displayed in the picture below (see fig. 268).
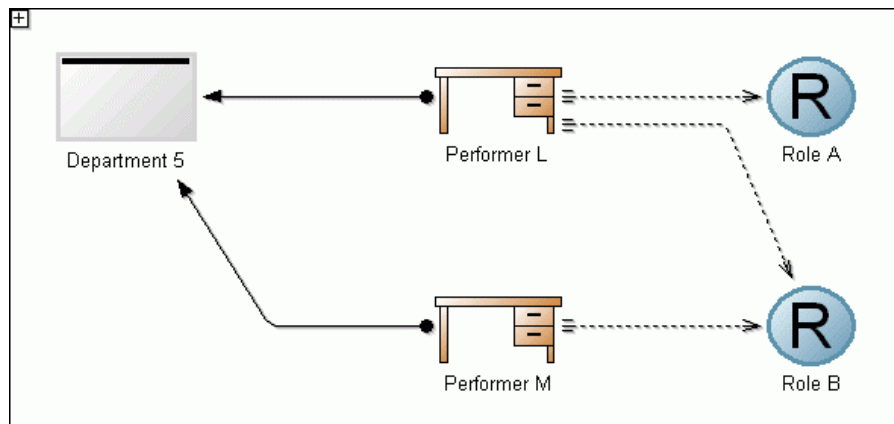
Figure 268: Example model in the ADONIS database

Variant 1: A model "Example model 1" of the type "WE model" is stored in the ADL file, which is different from the model stored in the database, since it does not contain the object "Role A" (see fig. 269).
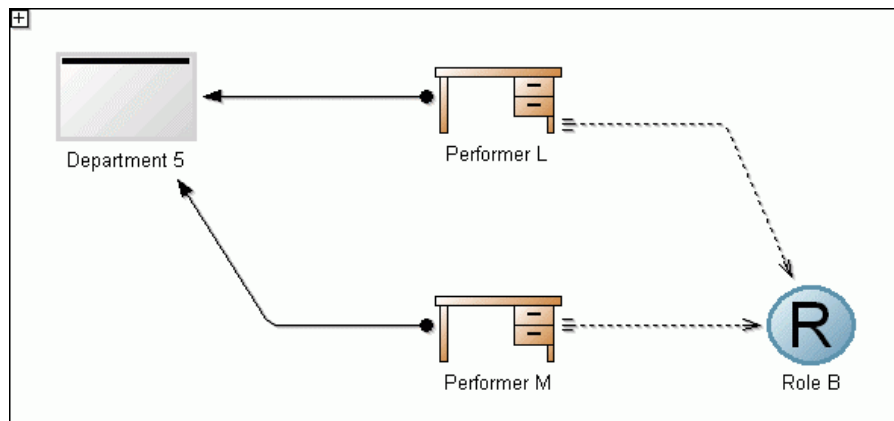
Figure 269: Example model in the ADL file (Variant 1)

Variant 2: A model "Example model 1" of the type "WE model" is stored in the ADL file, which is different from the model stored in the database, since it does not contain the connector from the object "Performer L" to the object "Role B" (see fig. 270).
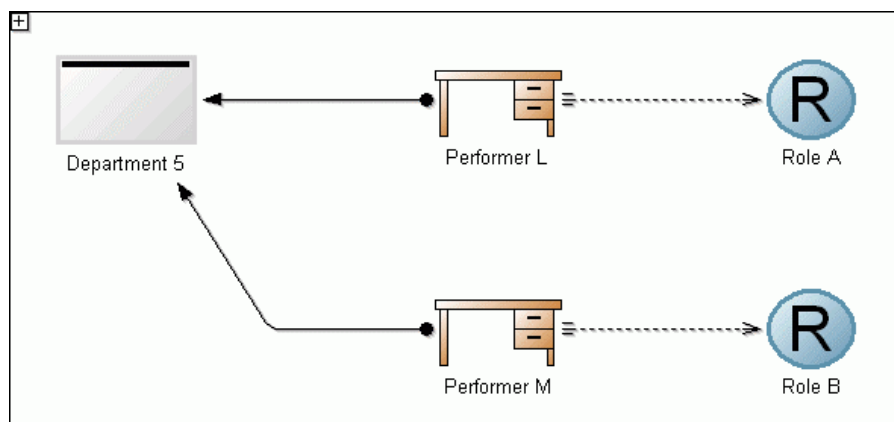
Figure 270: Example model in the ADL file (Variant 2)

<u>**Results:**</u>

When the option **"Delete not updated objects" is activated**, the result of the **Variant 1** equals the contents of the model stored in the file (see fig. 269) since the object "Role A" is not contained in the model to be imported.

When the option **"Delete not updated connectors" is activated**, the result of the **Variant 1** is as shown below (see fig. 271), since the connector from the object "Performer L" to the object "Role B" is not contained in the model to be imported.
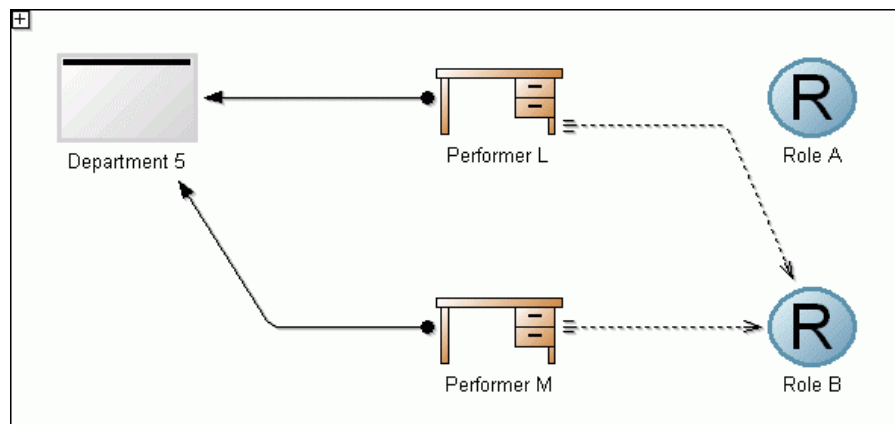
Figure 271: Import results by activated option "Delete non updated connectors"

When the **option "Delete not updated objects" is activated**, the result of the **Variant 2** equals the contents of the model stored in the database (see fig. 268).

When the option **"Delete not updated connectors" is activated**, the result of the **Variant 2** equals the contents of the model stored in the file (see fig. 270), since the connector from the object "Performer L" to the object "Role B" is not contained in the model to be imported.

3.4.4.4   Overwrite existing models

An existing model which is stored in the ADONIS database is overwritten by the model to be imported.

> **Hint:** In addition the general options for import (see chap. 3.4.4.1, p. 312) **"Update model references"** and - when using time-related versioning - **"Take over version numbers of model names"** are available.

> **ATTENTION:**   All application models that contains one of the imported models will be deleted during the ADL import in the overwriting mode. Inthis case a saftey query enables you to cancel the import.

> **Hint:** Activate in the window "ADL Import - options" the option **"Update model references"** to ensure that all references within the model you wish to import will be updated according to the change of name during the import.

> **Hint:** Activate in the window "ADL Import - options" the **"Take over version number in the model name"** to take over the model-related version number (see chap. 6.1.1, p. 69) of a model to the name of a model in a time-related version. This way each imported model will have it's own version thread created.
>
> The option "Take over the version number to the model name" is available only for ADL Import to an application library with time-related versioning (see chap. 6.1.2, p. 69) .

3.4.4.5　Ignore Models

If you select this option, any models to be imported from the ADL file which have matching names with existing models will in fact not be imported. Should an ADL file contain **only** models with matching names, an appropriate window will be shown:
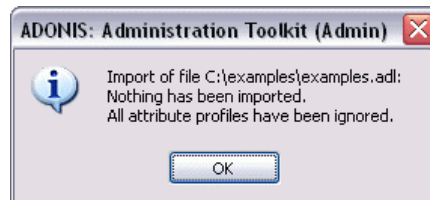
Figure 272: Import - all models were ignored

Click **"OK"** or hit the "Enter" key to close the window.

> **Hint:** In addition the general options for import (see chap. 3.4.4.1, p. 312) **"Update model references"** and - when using time-related versioning - **"Take over version numbers of model names"** are available.

3.4.4.6　To increase the version number

> **Hint:** The option "To increase version number" is only available in application libraries which have time-related versioning (see chap. 6.1.2, p. 69).

Models which should be imported and have the same names as the ones already stored in the database will be saved there under a new model version. The version number will then be assigned automatically i.e. such a model gets the next available validity date.

> **Hint:**　The version number of a model to be imported is not valid if it is based on an ADONIS-3.9 application library with model-related versioning (see chap. 6.1.1, p. 69) or when it comes from an ADONIS Version lower than 3.9 or when the version number does not correspond to the syntax definition in the current application library.

> **Hint:** When importing models coming from a model-related application library into the time-related one or when importing models with invalid version numbers from a time-related library (i.e. with syntax different from the used one) the user has to assign version number to every imported model (see chap. 3.4.4.11, p. 320).

> **Hint:** If a model with the same name and type as the imported one has already been stored in a database, the option **"Increase version number"** will be used and the model will be automatically assigned the next available version number.

> For models with matching names and model type and different invalid version numbers, the version numbers will be given increasingly (i.e. "January 2000", "February 2000"). The initial value will be assigned to the model with the oldest date of creation.

3.4.4.7　Rename attribute profiles from ADL file

The attribute profile to be imported will be renamed, if an attribute profile with the same name and from the same attribute profile class is already stored in the ADONIS database. In the case of matching names and classes, a window will be displayed asking if a new name should be assigned.

If you click **"Yes"**, the window **"ADL Import - Rename attribute profile"** appears:

Figure 273: Import - Rename Attribute profile

Hint: Clicking on the **"No"** button in the query dialogue will abort the import, since only attribute profiles with unique names can be stored in an ADONIS database.

Enter a new name for the attribute profile to be imported and click on the **"OK"** button or press the enter key to continue the ADL Import.

Hint: Clicking on the **"Cancel"** button in the window "ADL Import - Rename attribute profile" causes the abort of the import, since only attribute profiles with unique names can be stored in an ADONIS database.

3.4.4.8　Paste into existing attribute profiles

Attribute profiles to be imported will be aggregated in the ADONIS database with all attribute profiles from the same attribute profile class, which have already been stored there. In doing so, you can use the additional option "To replace existing record rows":
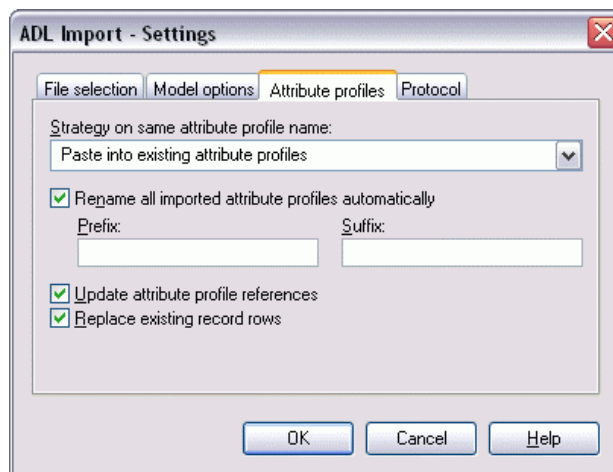
Figure 274: ADL Import - Extended Attribute profile options

Hint: If the option **"Replace existing record rows"** is *active*, the record rows of record attributes in the existing attribute profiles will be deleted and replaced by the rows in the attribute profiles to be imported. If the option is *deactivated*, the rows of record attributes in the attribute profiles to be imported will be appended to the lines in the existing attribute profiles . The effects are shown in an example (see p. 314).

In case of matching names and attribute profiles in the file and in the ADONIS database, the window "ADL Import - Replace values" appears for each of these attribute profiles.
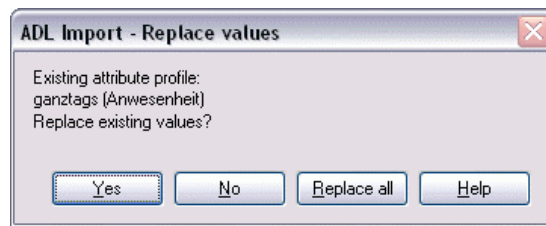
Figure 275: To Import attribute profiles - replace values

Click on the **"Yes"** button, if you want to replace the existing attribute value of the attribute profile by the value of the ADL profile in the ADL file.

If you click on the button **"Replace all"**, all attribute values will be replaced by the value of the imported attribute profile.

Clicking on the **"No"** button causes no changes to the attribute values of the existing attribute profile.

3.4.4.9 Overwrite existing attribute profiles

The attribute profile to import will be pasted into an attribute profile of the same class stored in the ADONIS database. While doing so, the attribute values will be taken over from the attribute profile to the file, if they differ from the values stored in the ADONIS database.

The attributes of these attribute profiles, which are stored in the ADONIS database and not contained in the model to be imported, will be set back to the standard value.

3.4.4.10 Ignore attribute profiles from ADL file

When selecting this option, attribute profiles with matching names will be ignored, i.e. not imported. Should there be **exclusively** attribute profiles with matching names in an ADL file, then an appropriate window will be displayed:
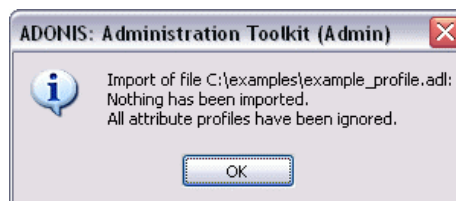
Figure 276: ADL Import - All attribute profiles have been ignored

Close the window by clicking on the **"OK"** button or pressing the Enter key.

3.4.4.11 Assign version number

> **Hint:** It is possible to assign a version number only to models of an application library, in which a time-related versioning (see chap. 6.1.2, p. 69) has been defined.

During the ADL Import of models, the window **"ADL Import - assign version number"** will be displayed with the defined initial value for each model with an invalid version number (i.e. not corresponding to the syntax or not available):
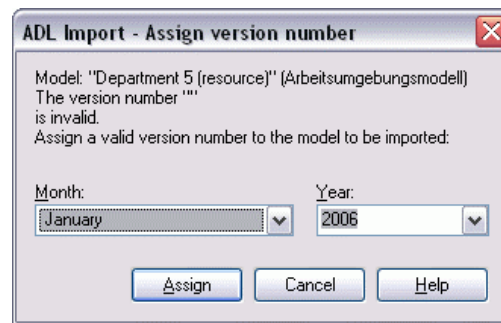
Figure 277: ADL Import - Assign version number

Select a version number and click on the button **"Assign"**.

> **Hint:** Clicking on the "Cancel" button in the window "ADL Import - Assign version number" will abort the import, since models, which are based on the current application library, are only authorised with a valid version number.

3.4.5 Import application models

This function allows the user to import models, this process contains the following steps:

- Start import
- Establish import settings and options
- Establish the aims of import and items to be imported
- Import

To start ADL Import of application models:

It is possible to import in two ways:

- Click and *hold on* the smart-icon and choose from the drop-down menu. You can see the menu item **"Application models"**,
- Choose from the menu **"Models"** menu item **"ADL Import"** and then its sub menu item **"Application models"**.

The window "File selection - ADL Import" where all ADONIS application libraries saved in the database are shown:
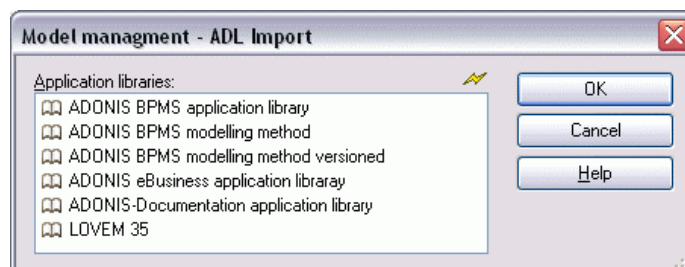
Figure 278: Model management - ADL Import

Choose the application library ,which contains the models or attribute profile to be exported and click on **"OK"** button. The chosen library will be shown in the "ADL Import - Settings" window :
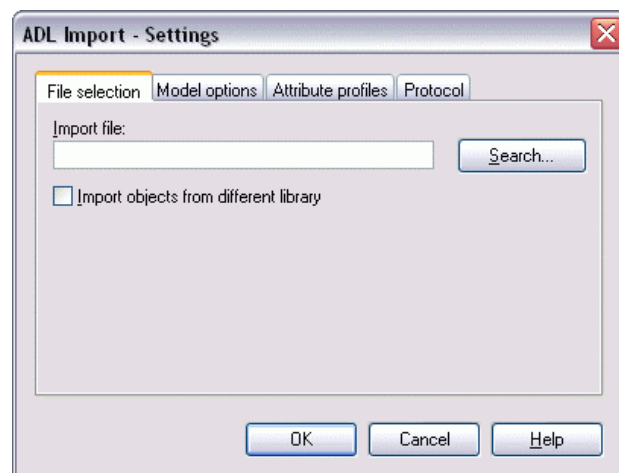
Figure 279: ADL Import

This dialogue consists of three parts, which can be identified with tabs.

ADL Import - Settings

In the window "ADL import - Settings" you can define the settings concerning the import in the tabs "File selection", "Model options", and "Protocol".

Tab "File selection":
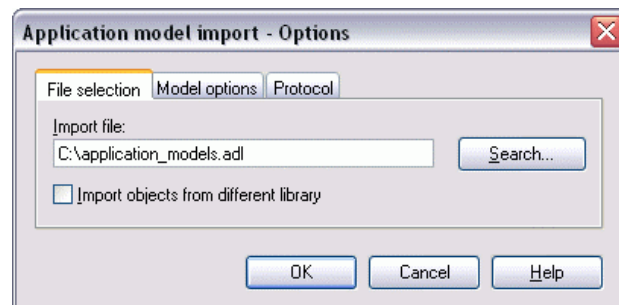
Figure 280: Application model import - File selection

Enter the path and the name of the ADL file with the model you wish to import into the **"File selection"** tab. The button **"Search"** opens a window allowing the user to search for the file to be chosen.

The **option "Import object from different library"** is available to import models which are based on a different application library than the previously selected.

Tab "Model options":

Here you define settings for import of application models:
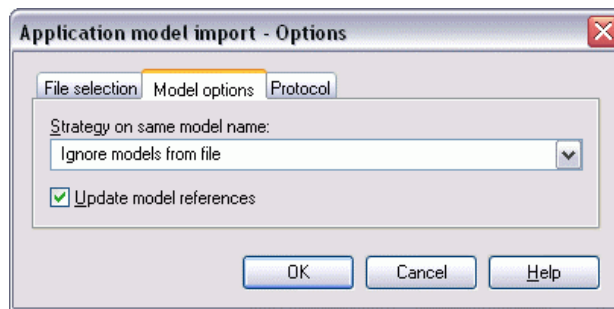
Figure 281: Import of application models - Model options

In the case of application models in the file to be imported, which have *the same names as the models stored in the ADONIS database*, you can always use the option **"Ignore models from file"** as the **"Strategy on same model name"**.

> **Hint:** When importing application models, which come from an application library with time-related versioning, you can choose the option "Increase version number" as reaction when the names of the models are the same. As a result, when you get an application model of the same model type and with the same model name, the next version number will be assigned to it during import automatically.

Activate the option **"Update model references"** to correspondingly update the references within the imported models during the change of the name of the imported models. Thanks to this option all the references to the imported models will be automatically updated.

> **Hint:** The option "Update model references" only has an affect when importing application models to an application library with a time-related version where you have selected the option "Increase version number" as a response.

Tab "Protocol":

In this tab you can have a protocol generated for the ADL Import.
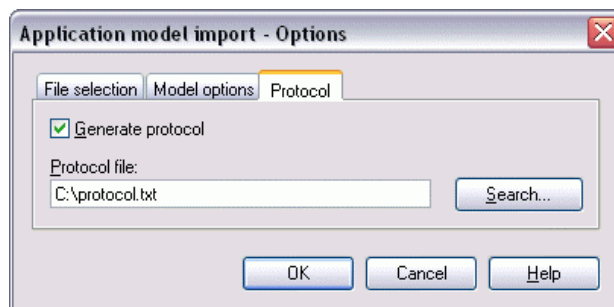
Figure 282: Application model import - Protocol

For this, activate the option **"Generate protocol"** and enter the path and the name for the import protocol (TXT file) into the **"Protocol path file"**. The Button **"Search"** allows the user to search for the ADL file.

Once you have successfully entered the settings, click on the **"OK"** button to assign them.

3.4.6   Model overview

The content of the ADL file to be imported is read and the application models contained in the file are listed in the window **"Application Model Import - Model overview"**:
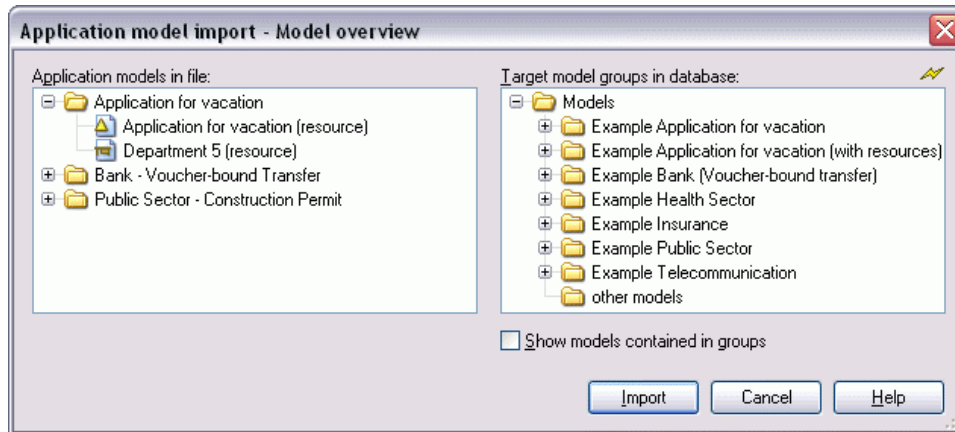
Figure 283: Application model import - Model overview

Select the model group into which a reference to the models defined in the application models imported should be generated.

> **Hint:** If you don't select a destination model group, the imported models will not be assigned to a model group and so can not be opened by an ADONIS user. Before the import, a security query will inform you about this.

Click on the button **"Import"** to import the models.

A status window will inform you on the progress of the ADL import.

> **Hint:** If the ADL file contains a model of a type which is not defined in the current application library, you can assign a model type (see chap. 3.4.2.1, p. 310) defined in the application library to this model during the import.

3.4.7   Result

After the ADL Import has been successfully completed, the window **"ADL Import - Results"** will inform you about the conclusion of the import. This window lists the names under which the import models are stored in the ADONIS database.

ADL Import - Results

Import of file C:\appliction models.adl:
Import has been successful.
The following application models have been imported:

⊞ 📁 Application for vacation
⊞ 📁 Bank - Voucher-bound Transfer
⊞ 📁 Public Sector - Construction Permit

Close   Help
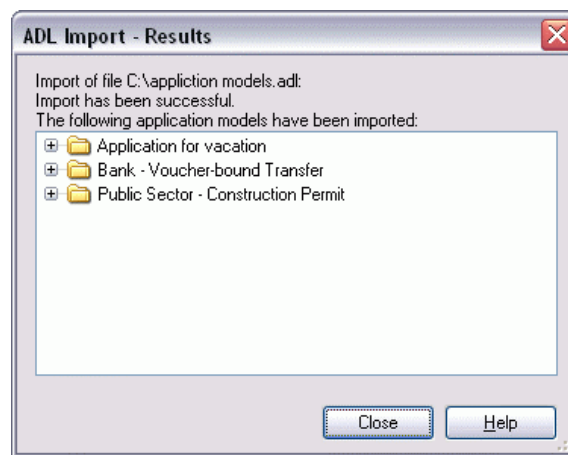
Figure 284: Application model import - result display (with protocol)

If you activate the option "Generate protocol" in the window "ADL Import - Options" - "Protocol" tab (see fig. 280) and give the path and the name for the protocol file to be created, then the button **"Protocol"** will be shown in the "ADL Import - Results". By clicking on this button the generated protocol will be shown in a text editor.

3.5 ADL Export

The ADL Export functionality allows you to save your models in text format (ASCII) and, if necessary, import them into a different ADONIS database.

You may export either **models/attribute profiles** (see chap. 3.5.1, p. 325) (and model groups/attribute profiles groups) or **application models** (see chap. 3.5.2, p. 329).

3.5.1 Model Export

With this function you can export models, model groups, attribute profiles and attribute profile groups.

Start an ADL export for models/attribute profiles:

There are three options how to begin exporting:

- Click on the smart-icon 📥,

- Click and hold the smart-Icon 📥 and select from drop down menu the option **"Models/attribute profiles"**,

- Select from menu **"Models"** the option "ADL Export" **from the sub menu select the option "Models/Attribute profiles"**.

The window "ADL export - Model selection" with all application libraries stored in the ADONIS database will be shown:
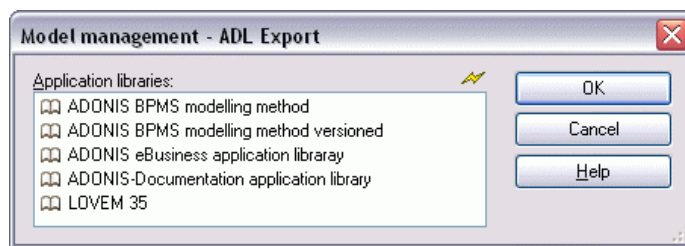
Figure 285: ADL export - Model selection

Select the application library, that contains both the models and the attribute profiles to be exported and click on **"OK"**. The selected application library will be loaded and the window "ADL Export - Model selection" will appear in which all the models stored in the ADONIS database are listed according to the model group hierarchy.:
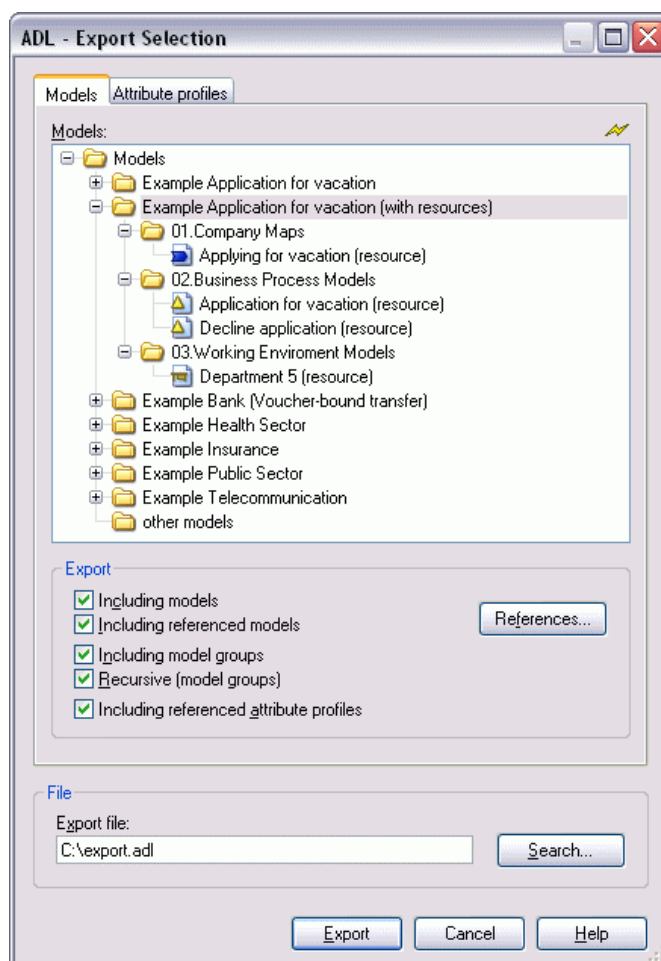
Figure 286: ADL Export - Model selection

Through **tabs** "Models" and "Attribute profiles" you can toggle between model and attribute profile selection. In the window **"Selection"** the complete hierarchy of models, model groups or attribute profiles is visable. The function **"Export"** contains export options.

Prepare model export:

Select the tab **"Models"**, the model hierarchy will be shown:
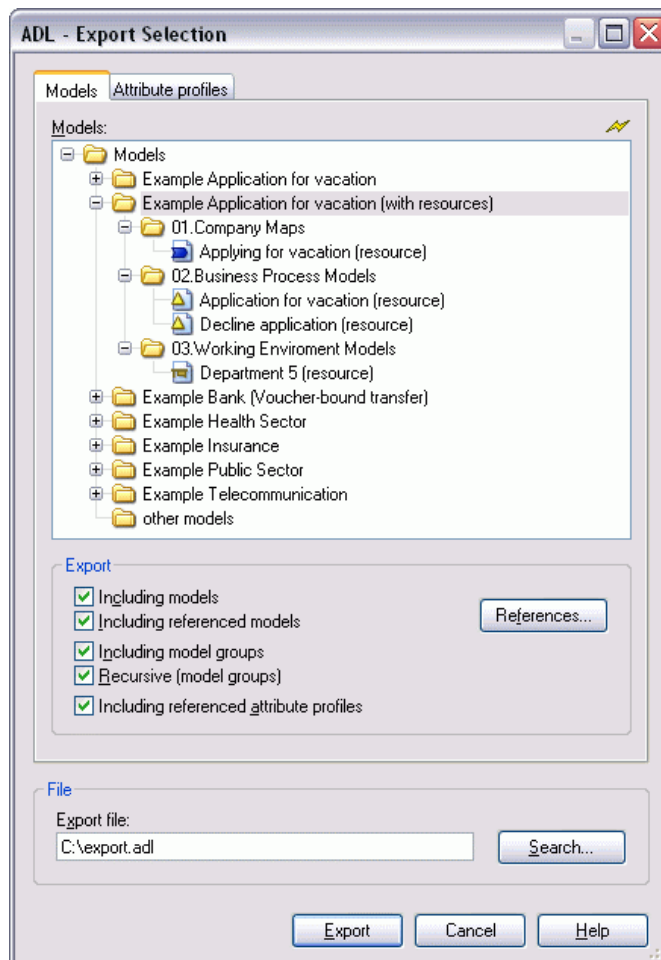
Figure 287: Model and model group selection

Select in the window **"ADL Export - Selection"** the models and/or model groups to be selected.

Define the required export options:

Including models The selected models will be imported.

Including referenced models

> Apart from the selected models to be exported, the models referenced to these selected models will be exported - regardless of their selection. Clicking on the option "References", will enable you to change the settings . ("inclusive referenced models" is only accessible, if option "Models" is active.)

Including model groups The selected model groups will be exported.

Recursive (model groups)

> Apart from the selected model groups to be exported, all the sub groups of all hierarchy levels will be exported. ("Recursive (model groups)" is only accessible, if option "Model groups" is active.)

Including referenced attribute profiles

All referenced attribute profiles of the models to be exported will also be exported. ("Including referenced attribute profiles" is only accessible, if the option "Models" is active.)

Prepare attribute profile export:

If it is not necessary that you also export automatically all referenced attribute profiles, you can always select the ones to be exported manually. In order to do this choose the tab **"Attribute profile"**. Attribute profile hierarchy will be shown:
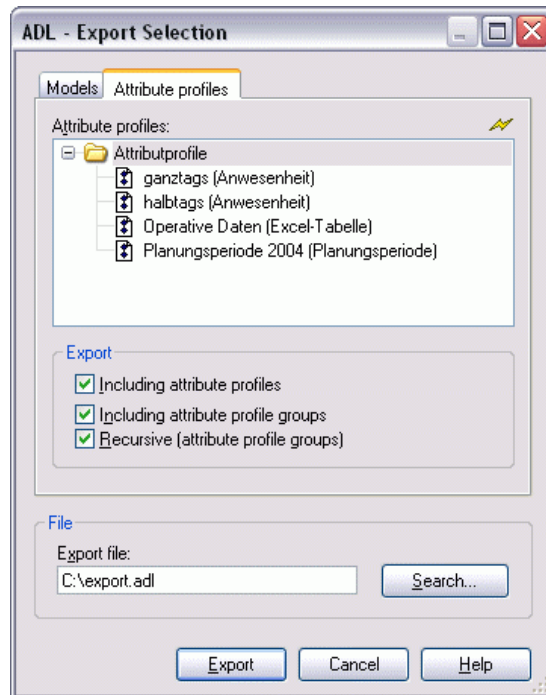
Figure 288: Select attribute profiles and attribute profile groups

ATTENTION: You can select attribute profiles to be exported, only if in tab "Models" the option "Referenced attribute profiles" is *deactivated*.

Mark in the window **"Selection"** the attribute profiles and/or groups to be exported.

Define required export options:

Including attribute profiles Selected attribute profiles will be exported.

Including attribute profile groups Selected attribute profile groups will be exported.

Recursive (attribute profile groups)

Apart from the selected attribute profile groups, all the sub groups of all hierarchy levels will be exported. ("Inclusive all sub groups" is only accessible, if option "Attribute profile groups" is active.)

Export:

Define in **"File name"** the path and the name for the ADL file to be created. The button **"Search"** opens the supporting dialogue window.

Click on button **"Export"** to start exporting. A status window will keep you informed about the pro-
gress. Another window will inform you when the ADL export has been completed successfully.

Special case: ADL Export for ADONIS Version 3.0x

If you want to export models and/or model groups as an ADL file for ADONIS Version 3.0x open the
window **"Browse"** to enter the path and the name for the file. In the drop down field **"Data type"**
choose **"ADL 3.0 files"**.

> **Hint:** When exporting ADL to ADONIS Version 3.0x you will lose all model information,
> which was implemented from Version 3.5 (e.g. attributes in a new attribute type).

Special case: ADL export for ADONIS version 3.81

During ADL export in ADONIS 3.9, redundant class attribute values of connectors are no longer ex-
ported. If you want to include these attribute values into an ADL export, open the window **"Browse"** to
enter the path and the name for the file. From the drop-down field **"Data type"** choose **"ADL 3.81
files"**.

> **Hint:** The class attribute values of connectors are defined in the application library and are
> not modified by the ADL file (up to version 3.81). Not exporting the class attribute values
> (default in ADONIS 3.9) does not have any negative impact on data backup or data trans-
> fer between ADONIS databases.

3.5.2   Export Application Models

This function allows for the export of application models stored in a database.

Start ADL export of application models:

There are two possible ways to do it:

● Click and hold on the smart-Icon 📥 and select from the drop-down menu the option **"Application
   models"**,

● Select from menu **"Models"** the menu item **"ADL Export"** and from its sub menu **"Application
   models"**.

In the window "Model management - ADL export", all applications libraries stored in a database will be
displayed:
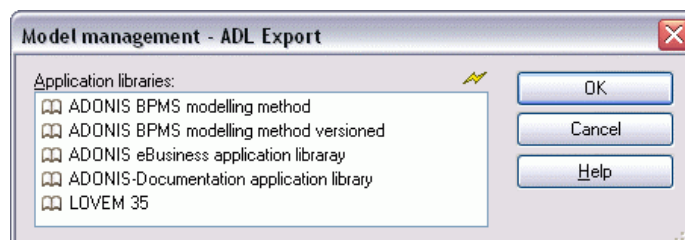
Figure 289: Model management - ADL export

Select the application library, on which the models to be exported are based on and click on **"OK"**.
The chosen application library will be loaded and the window "ADL Export - selection of application
models" with all defined application models will be shown:
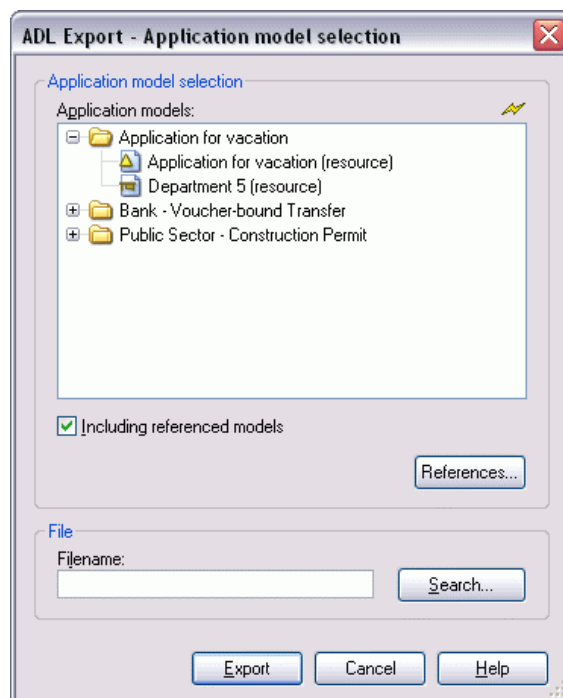
Figure 290: To export application models

All application models are displayed in the window **"Application models"**. Mark all models which shall be exported.

Activate the option **"Including referenced models"**, to export selected application models as well as all their referenced models. With the button "References" these settings can be changed.

Export:

Define in **"File name"** the path and the name for the ADL file. With button **"Search"** you can open the supporting dialogue window.

Click on the button **"Export"** to start exporting. A status window will keep you informed about the progress. Another window will inform you when the ADL export has been completed successfully.

Special case: The use of a sequence version and a version-specific application library

When using a time-based application library, you can switch between the options *Export of version specific application models* and *Export of sequence version application models.*

When exporting *Version specific application models*, only models with the same model versions as the models contained in the application model will be exported. When exporting *Sequence version application models*, all versions of models contained in application model will be exported.

3.6   Delete ADONIS Models

In order to delete models, you must first either select the option "Delete models" from the "Models" menu or click on the appropriate smart-icon 🖳 in the quick-access bar.

Once you have selected the menu item or clicked on the smart-icon, the window "Model management - delete models" (see fig. 291) will be displayed with all the application libraries stored in the ADONIS database.
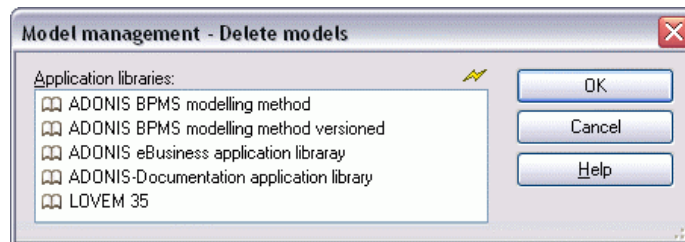
Figure 291: Model management - delete models

Select the application library that contains the models to be deleted and then click on the" OK" button. The application library selected is now loaded and the window "Delete Models - model selection" (see fig. 292), which lists all the models stored in the ADONIS database ordered according to model type, is displayed.

3.6.1 Deleting models

All models saved in the ADONIS database are listed in the "Delete models - selection" windows (see fig. 292)- arranged according to their modeltype .
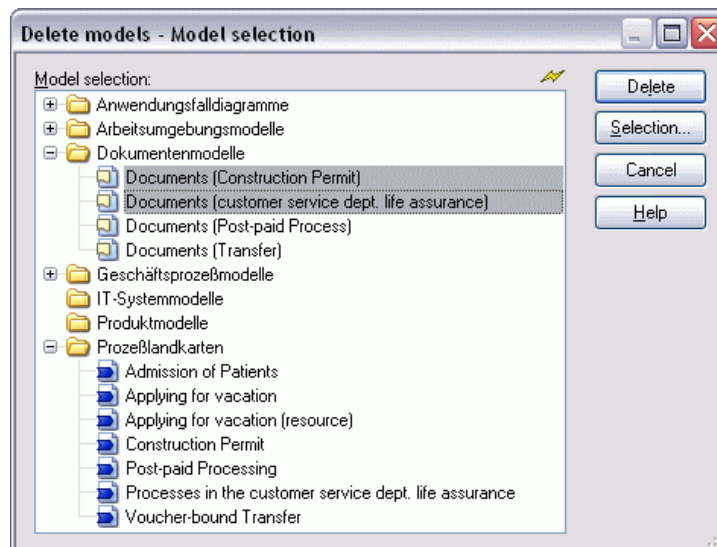
Figure 292: Delete models - selection

Select the models you want to be deleted and click **"Delete"**.

> **Hint:** The models to be deleted, which are referenced by other models, will be displayed in the "Delete models - Information" window before the deletion (see fig. 293).

If you click **"Selection"** the "Find model" window (see fig. 294) will be shown, in which you can limit the model selection via the search (see chap. 3.6.1.2, p. 332) to specific model attributes.

Before the models are finally deleted from the ADONIS database, an appropriate security message is displayed.

On confirming the security message, it will be checked whether the models to be deleted are referenced by other models. If this is the case for at least one model, the window "Delete models - Informa-

tion" will be displayed. This window lists the models, which are referenced by other models in the ADONIS database. Click **"Delete"** again to continue the deletion. If you hit the **"Cancel"** button the deletion will be cancelled.

> **ATTENTION:** A model can only be deleted if no ADONIS user has opened this model in the ADONIS Business Process Management Toolkit at that time.

> **ATTENTION:** The deletion of models from the **ADONIS** database cannot be undone!

3.6.1.1   Show Referenced Models

In the window "Delete models - Information" (see fig. 293), all models, which are referenced by other models will be displayed.
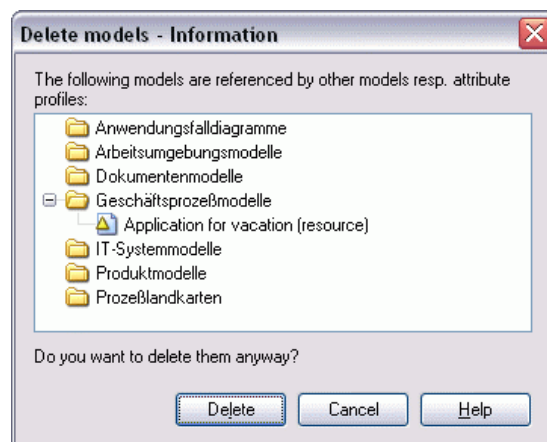
Figure 293: Delete models - referenced models

Click **"Delete"** to continue the deletion of the selected models.

> **Hint:** When deleting referenced models, the references outgoing from the start models can no longer be followed and will be displayed as broken references.

3.6.1.2   Find Model

The model search based on model attributes can either be carried out according to standardised or user-defined queries (similarly to the queries of the Analysis Component).

If you wish to search for a model by means of a standardised query, select one of the standardised queries from the list **"Query"** in the window "Model search" (see fig. 294).
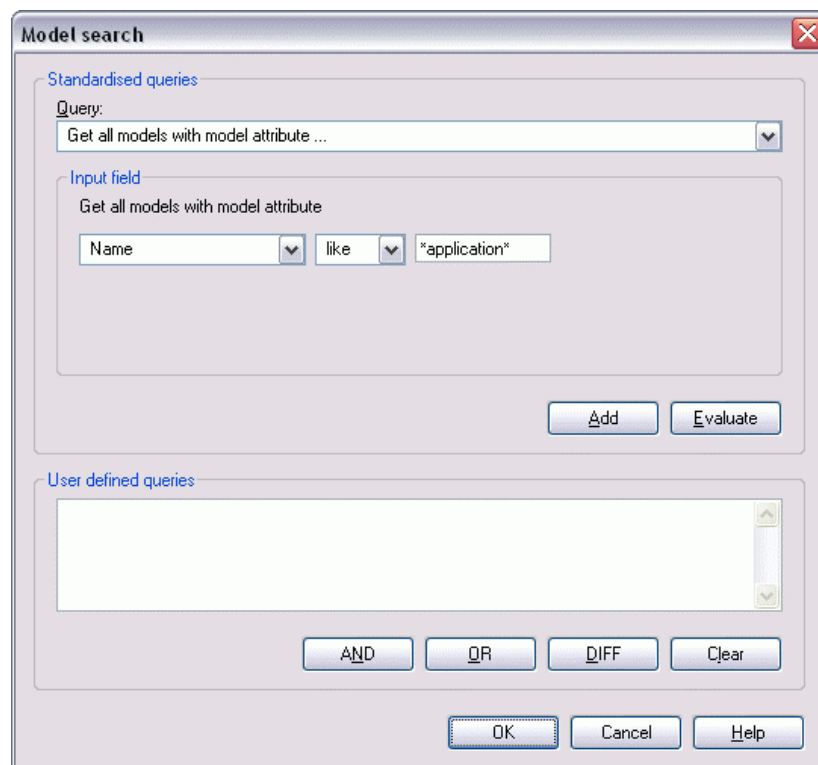
Figure 294: Model search

Enter the necessary information or select the values required from the list of modeltypes, attributes or operators of comparison.

The following pre-defined queries are available:

- `Get all models.`

  This query will display all models saved to the database.

- `Get all models of type` [*selection*]`.`

  This query will display all models of the selected type.

- `Get all models with`

  `model attribute` [*selection*][*operator of comparison*][*input*]`.`

  This query will display all models whose model attributes meet the criteria defined.

- `Get all models of type` [*selection*] `with`

  `model attribute` [*selection*][*operator of comparison*][*input*]`.`

  When using this query, you will find all models of the specified type which have a specific model attribute.

  Note: First enter the modeltype, then the model attribute, the operator of comparison and then enter the attribute's value to be compared with.

  Example:

  Selecting the model type business process model with the model attribute "Date last changed", the comparison operator "<" and inputting the value "01.01.2000" lists all models that were last changed before January 1st 2000. (Note, the dates stored in the **ADONIS** database include the time as well as the date).

You can carry out a standardised query by clicking **"Evaluate"**. Clicking **"Add"** changes the query into an AQL expression which is then displayed in the field **"User defined queries"**. You can edit this query by entering additional AQL expressions (see chap. 12., p. 593). You can combine or connect the queries by using the buttons **"AND"**, **"OR"** and **"DIFF"**. Delete the contents of the field **"User defined queries"** by clicking on the **"Clear"** button.

You can create user defined queries by yourself in the field **"User defined queries"**. By using AQL you can define any query you like. You may of course use standardised queries to support you when creating user defined queries.

4. Attribute Profile Management

Attribute profiles represent the repository in ADONIS and are reusable attribute structures.

The following graphic (see fig. 295) gives an overview of the relations of attribute profile management in the ADONIS Administration Toolkit.
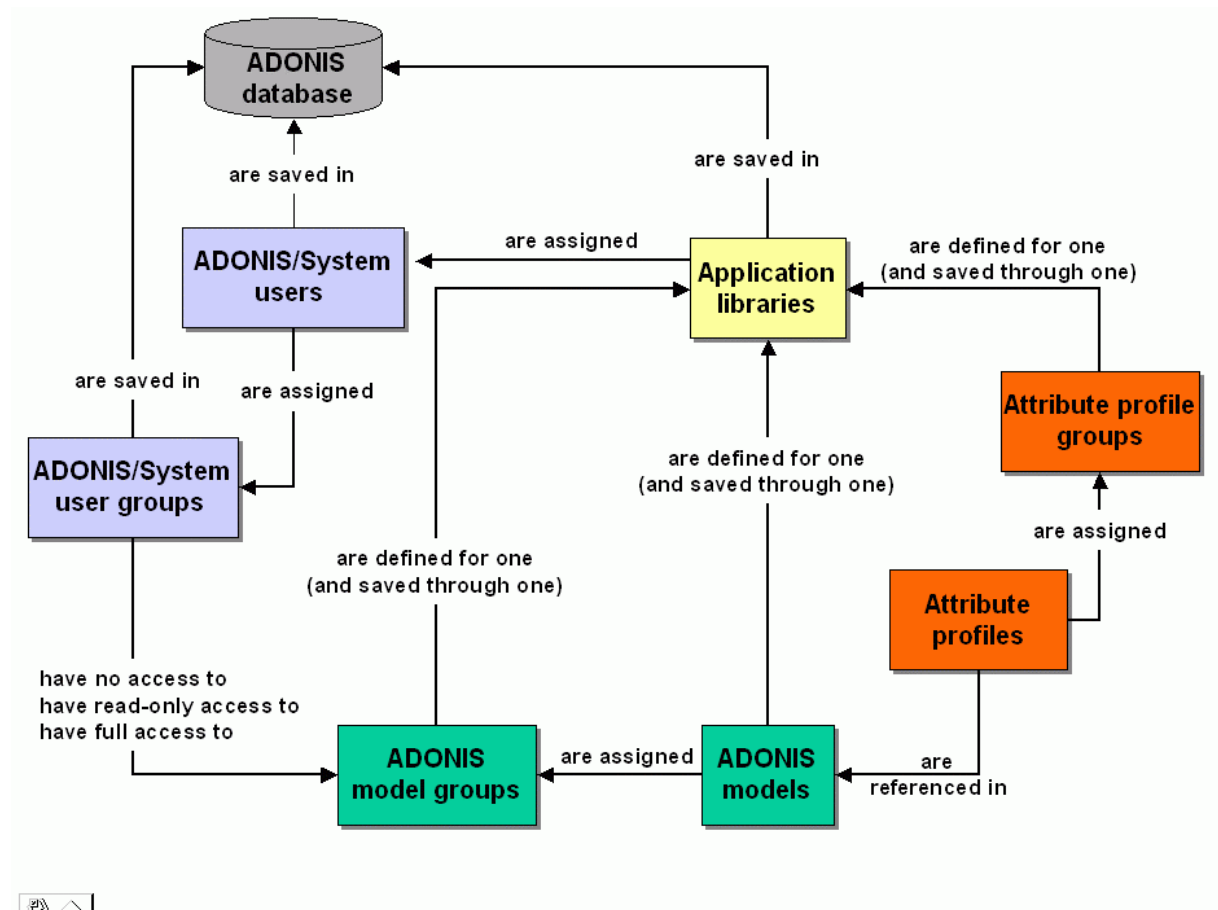
Figure 295: Overview of the components in the Administration Toolkit

Attribute profiles (see chap. 8., p. 77) are defined for all application libraries and are available both in models of the BP library and in models of the WE library. Attribute profiles are created in the attribute profile groups in the attribute profile management. The reference to an attribute profile class is defined in the classes. The concrete assignment of the attribute profile (of an attribute profile class) is carried out on the objects during the modelling.
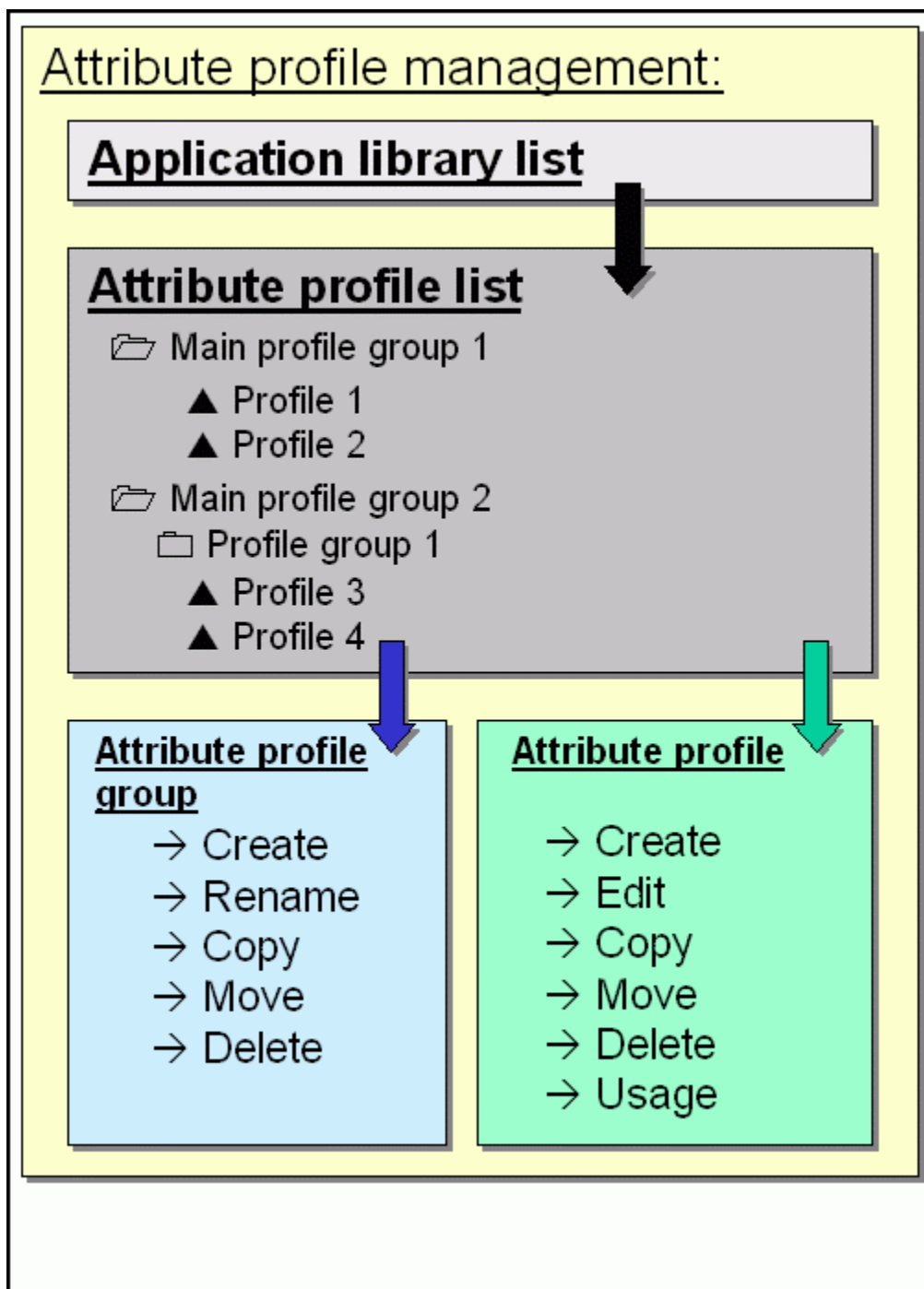
Figure 296: Functionality in the Attribute Profile Management

Using the Attribute Profile Management (see fig. 296), you can define new attribute profiles, edit, copy, move and delete existing attribute profiles as well as show their utilisation and additionally create new attribute profile groups, rename, copy, move or delete existing attribute profile groups.

To use the services provided by the attribute profile management component of ADONIS, please proceed according through the following steps:

1. Start the ADONIS Administration Toolkit. The window ADONIS Administration Toolkit (see chap. 3., p. 16) will appear.

2. Activate the "Attribute profile management" by clicking on the corresponding smart-icon 🔢 in the quick-access bar.

   Alternatively, you can activate this component by opening the popup menu of the component bar. Click with the right mouse button on the component bar beneath the smart-icons for the components. Then select the menu option "Attribute profile management". Alternatively you can use the function key <F11> to open the popup menu and the key <a> to select th Attribute Profile Management.

   Once the Attribute Profile Management is active, the quick-access bar is displayed with the smart icon for the editing of this attribute profiles (see fig. 297).
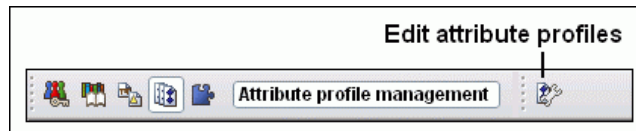
Figure 297: Icon bar of the Attribute Profile Management

4.1 Editing Attribute Profiles

By selecting the menu item "Edit attribute profiles" in the menu "Attribute profiles" and then loading the application library selected in the window "Edit attribute profiles", it is possible starting from the list of attribute profiles, to edit attribute profiles and attribute profile folders. All existing attribute profiles and attribute profile folders will be shown in the "Attribute profile" list:
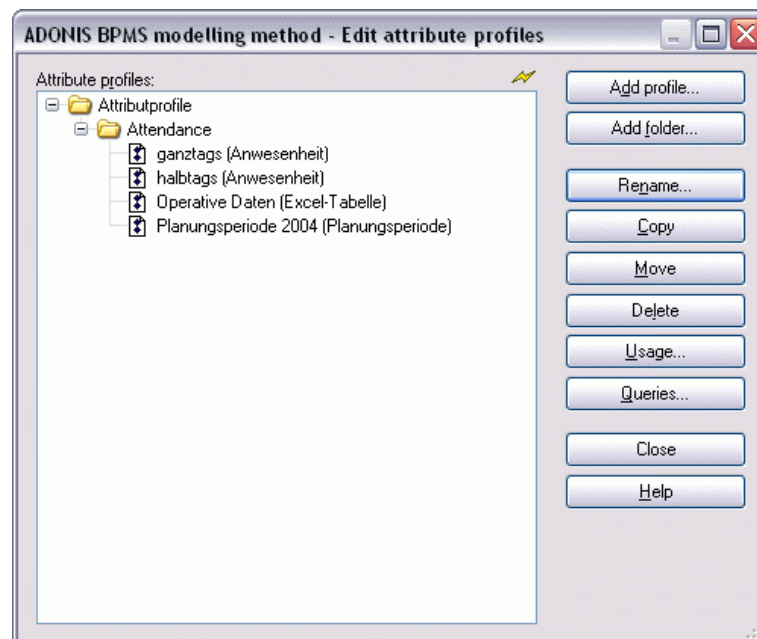
Figure 298: Attribute Profile (group) list

Clicking on the following buttons will enable you to:

"New attribute profile"   Create a new attribute profile (see chap. 4.1.6, p. 340) in a previously selected attribute profile folder.

"New version"   Save a previously selected attribute profile folder as a new version (see chap. 4.1.7, p. 341).

	Note: This button is only available if the attribute profiles are defined in an application library with time-related versioning (see chap. 6.1.2, p. 69).
"New group"	Create a new (main) attribute profile folder (see chap. 4.1.1, p. 338) in a previously selected attribute profile folder or in a not previously selected attribute profile folder.
"Rename"	Assign a new name (see chap. 4.1.2, p. 339) to a previously selected attribute profile folder.
"Edit"	Edit (see chap. 4.1.8, p. 341) the values of a previously selected attribute profile or the values of several previously selected attribute profiles at the same time.
"Copy"	Copy a previously selected attribute profile folder (see chap. 4.1.3, p. 339) or a previously selected attribute profile (see chap. 4.1.12, p. 372).
"Move"	Move a previously selected attribute profile folder (see chap. 4.1.4, p. 340) or a previously selected attribute profile (see chap. 4.1.13, p. 372) into another attribute profile folder.
"Delete"	Delete a previously selected attribute profile folder (see chap. 4.1.5, p. 340) or a previously selected attribute profile (see chap. 4.1.14, p. 373).
"Usage"	Show a list of the objects and models, in which the previously selected attribute profiles are referenced (see chap. 4.1.15, p. 373).
"Query"	Evaluate (see chap. 4.1.16, p. 373) the contents of the attribute profiles using the criteria you have defined.

The general functionality (see chap. 4.1, p. 34) as well as the above-listed functionality of the attribute profile list are available if you open the popup-menu (right mouse button) .

> **Hint:** The availability of functions depends on the previous selection of an attribute profile or of an attribute profile folder.

4.1.1 Add Attribute Profile Folder

It is possible to create attribute profile folders in each hierarchy level, i.e. you can create a new attribute profile folder in each attribute profile group.

To create a new attribute profile folder, select the attribute profile folder, which should contain the new attribute profile folder and then click on the button "add folder".

> **Hint:** To create an attribute profile folder as a main group, i.e. on the upper hierarchy level, no attribute profile folder should be selected.

The window "Add attribute profile folder" (see fig. 299) appears.
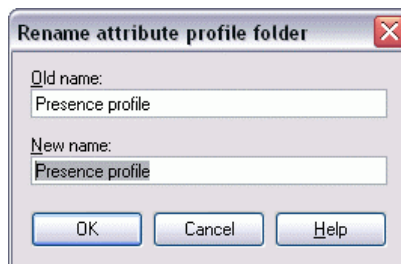

Figure 299: Add attribute profile folder

Enter the name of the new attribute profile folder and then click on the OK button.The window will be closed and the updated hierarchy of the attribute profile folders will be shown.

> **ATTENTION:** The name of the new attribute profile folder must be unique on the level of the main folders and within each attribute profile folder.

4.1.2 Rename Attribute Profile Folder

If you want to give a new name to an already existing attribute profile folder, select this attribute profile folder and then click on the button "Rename". The window "Rename attribute profile folder" will appear (see fig. 300), in which the current name (Field "Old name") is displayed. In the field "New name" the current name is also entered and selected so that you can directly enter the new name.

Figure 300: Rename attribute profile folder

Enter the name of the attribute profile folder and then click on the OK button. The window will be closed and the updated hierarchy of attribute profile folders will be displayed.

> **ATTENTION:** The name of the new attribute profile folder must be unique on the level of the main folders and within each attribute profile folder.

4.1.3 Copy Attribute Profile Folder

Select the attribute profile folder you want to copy into another attribute profile folder and then click on the button "Copy". As soon as you move the mouse pointer onto the list of the model hierarchy, it will

be changed into .

Then click on the attribute profile folder to which you want to copy the previously selected attribute profile folder.

If you want to move the attribute profile folder to the upper hierarchy level (main group), click on the free space below the model hierarchy or on the left near the main attribute profile folders.

> **Hint:** When you copy a new attribute profile folder, the whole structure of this folder, i.e. all contained sub-profile folders as well as all attribute profiles contained in the folders, will be doubled.

> **Hint:** When you copy an attribute profile folder at the same place, a number will be automatically appended to the name of the profile folder, in order to make it as clear as possible.

4.1.4   Move Attribute Profile Folder

If you want to move an attribute profile folder, select this attribute profile folder and then click on the button "Move". As soon as you move the mouse pointer onto the list of the model hierarchy, it will be

changed into   .

Click on the attribute profile folder to which the previously selected attribute profile folder should be moved to.

If you want to move the attribute profile folder to the upper hierarchy level (main folder), click on the free space below the model hierarchy or on the left near the main attribute profile folders.

> **Hint:** The name of the attribute profile folder to be moved shall not be given to another attribute profile folder in the new attribute profile folder or within the main folder.

4.1.5   Delete Attribute Profile Folder

Select one or more attribute profiles that you want to delete and then click on the button "Delete".

Before the final deleting of the attribute profile folder, an appropriate security query will appear.

Should some references to attributes profiles (contained in the attribute profile folder to be deleted) remain, then the window "Delete attribute profile - used attribute profile" (see fig. 301) will be displayed, indicating from which objects in which models these attribute profiles are referenced.
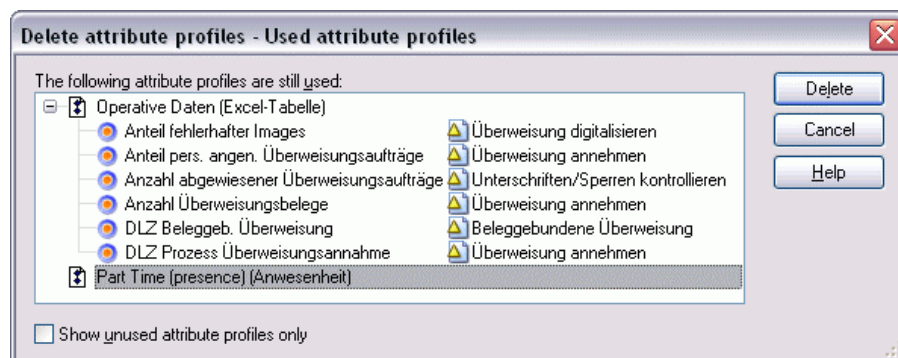
Figure 301: Delete attribute profile folders - used attribute profiles

Select the attribute profiles you want to delete despite its usage, activate the option "Delete selected attribute profiles" and click on the OK button.

4.1.6   Add Attribute profile

When you add an attribute profile, an instance of the attribute profile class defined in the application library will be created.

> **Hint:** Attribute profiles must be created in a previously selected attribute profile folder.

To create a new attribute profile, select the attribute profile folder, in which you would like to create it in and then click on the button "New Attribute profile".

If there are several attribute profile classes defined in your application library, the window "New attribute profile" will appear (see fig. 302).
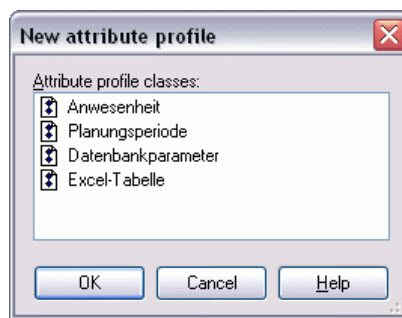
Figure 302: Add attribute profile

Select the attribute profile class from which the instance of the new attribute profile must be created and then click on the OK button.

> **Hint:** When using an application library with **time-related versioning** (see chap. 6.1.2, p. 69), you can give a version number for the new attribute profile.

In the window displayed as a result (see fig. 304), you can define the attribute value of the new attribute profile (see chap. 4.1.8, p. 341).

4.1.7 Save Attribute Profile as New Version

> **Hint:** The functionality "Save attribute profile as a new version" is only available if the attribute profiles are defined in an application library with time-related versioning (see chap. 6.1.2, p. 69).

In the window "Save as new version", you can define a new version number to save an existing attribute profile.
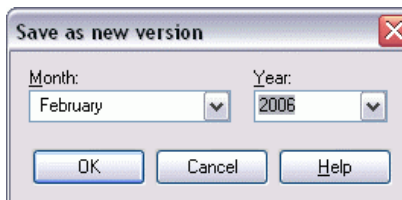
Figure 303: Save as new version

4.1.8 Edit Attribute Profile Values

When editing an attribute profile, you will define the attribute values of this profile. These attribute values will be integrated into the modelling object through the referencing of the attribute profile during modelling in the Business Process Management Toolkit.

The editing of an attribute profile is carried out in an ADONIS Notebook (see fig. 304)- the same way as for attributes in objects.

> **Hint:** The representation of attributes and the structure of ADONIS Notebooks depends on the cutsomised definition in the application library.

> **Hint:** Select several attribute profiles to edit their values simultaneously in the ADONIS browser (see chap. 4.1.9, p. 370).
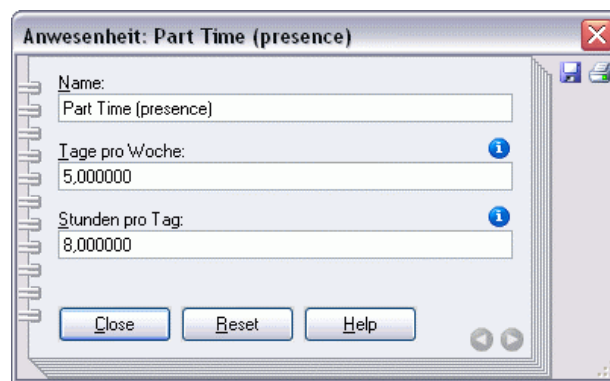
Figure 304: Edit attribute profile

Define the value of an attribute profile by entering the appropriate values in the attribute fields.

> **ATTENTION:** The name of an attribute profile must be unique within an attribute profile class.

Then click on the button "Close" to save the attribute values and close the ADONIS' Notebook. The updated list of attribute profiles will be displayed.

Support for entering data

ADONIS provides dialogues for entering complex attribute values (or formulas) into the ADONIS Notebooks. Whether such a dialogue is available for a given attribute is indicated by the "Dialog" icon ▢ above the attribute input field.

> **Hint:** The availabilty of the support for entering data depends on the definition in your application library.

The following dialogue supports for entering data can be available:

Edit record attributes (see chap. 4.1.8.1, p. 343)
to edit attributes of type "Record".

Define colour (see chap. 4.1.8.2, p. 344)
to select a colour (for the graphical display of an object).

Edit date attributes (see chap. 4.1.8.6, p. 346)
to edit attributes of type "Date".

Date /Date and time attributes (see chap. 4.1.8.6, p. 346)
to edit attributes of type "Date and Time".

Edit time attributes (see chap. 4.1.8.8, p. 347)
to edit attributes of type "Time".

Add references (see chap. 4.1.8.9, p. 348)
to create model references in attributes of type "Interref".

Define the performer's calendar (see chap. 4.1.8.14, p. 359)
to define the time and presence of performers.

Define the process calendar (see chap. 4.1.8.15, p. 365)
to define the appearence profiles of processes.

> **Hint:** The attributes "Calendar" and "Process calendar" are edited through a button (called "Calendar" or "Process calendar") in the ADONIS Notebook instead of the "Dialog" icon and a separate input field.

Hint: Open the support for entering data for references attributes using the "Add" icon ✚.

A special support for entering data (see chap. 4.2.3.3, p. 42) is available for the definition of attribute values for attributes of type "Expression". This support for entering data will be started with the "expression dialogue" icon *fx* .

4.1.8.1 Edit Record Attributes

ADONIS provides you with a support dialogue for editing record attributes, which is to be opened in the ADONIS Notebook by clicking on the "dialogue" button ▢, on the right above the record display.

Once you have clicked on the button, the record will appear in the window "*<Object name> - <Name of the record attribute>*" (see fig. 305), in which you can edit attribute values, as well as add and delete rows.
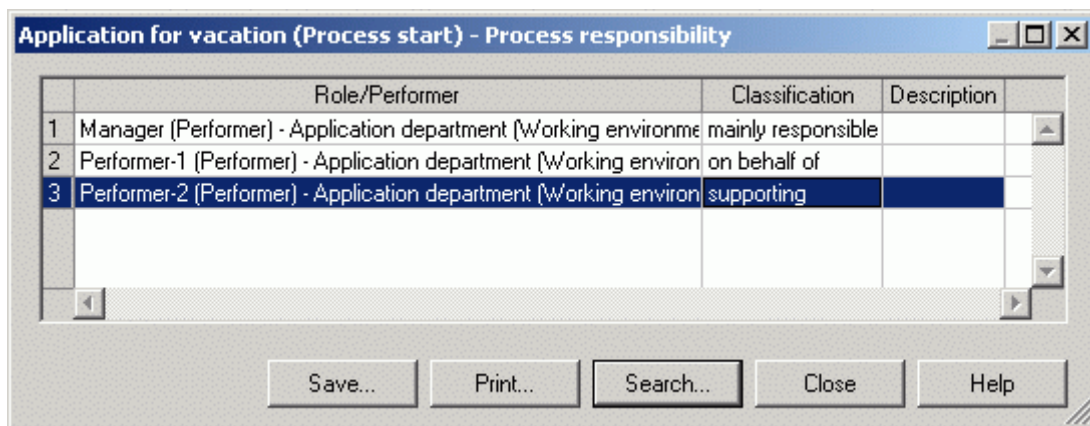
Figure 305: Edit record attributes

To **edit an attribute value**, first select the appropriate cell and then the menu item "Edit attribute" in the context menu (right mouse button). The input is done directly into the cell or an input window depending on the type of the selected attribute.

To **copy the value of an attribute**:

1. Select the cell with the attribute value to be copied

2. Select in the context menu (right mouse button) the menu item "Copy"

3. Select the cell into which you want to paste the attribute value and

4. Select in the context menu (right mouse button) the menu item "Paste".

   Hint: Copying values within a record is only possible for attributes of the same type or in attributes of type "String".

To **copy the value of a row**:

1. Mark the row to copy by clicking on the row number (grey column)

2. Select in the context menu (right mouse button) the menu item "Copy rows"

3. Select the row into which the copied one has to be pasted and

4. Select in the context menu (right mouse button) the menu item "Paste".

   Hint: When copying rows, the values of the copied rows will be inserted into an already existing row and so will overwrite the values contained in this row.

To **add a row to a record**, select in the context menu the menu item "Insert row". This new row will be inserted at the bottom of the record.

To **move a row in the record**, mark the row to be moved by clicking on the row number (grey column), then select in the context menu the menu item "Move row" - the mouse pointer will change to

- and click on the row above where the new row has to be inserted.

To **delete a row from the record**, mark the row to delete by clicking on the row number (grey column) and then select in the context menu the menu item "Delete rows".

4.1.8.2 Define Colours

ADONIS provides you with an input window for the selection of a colour (e.g. for the graphical display of the object), which you can open in the ADONIS Notebook by clicking on the "dialogue" button , on the right above the corresponding attribute field.

Once you have clicked on the button, the window "colour" (see fig. 306) will appear.
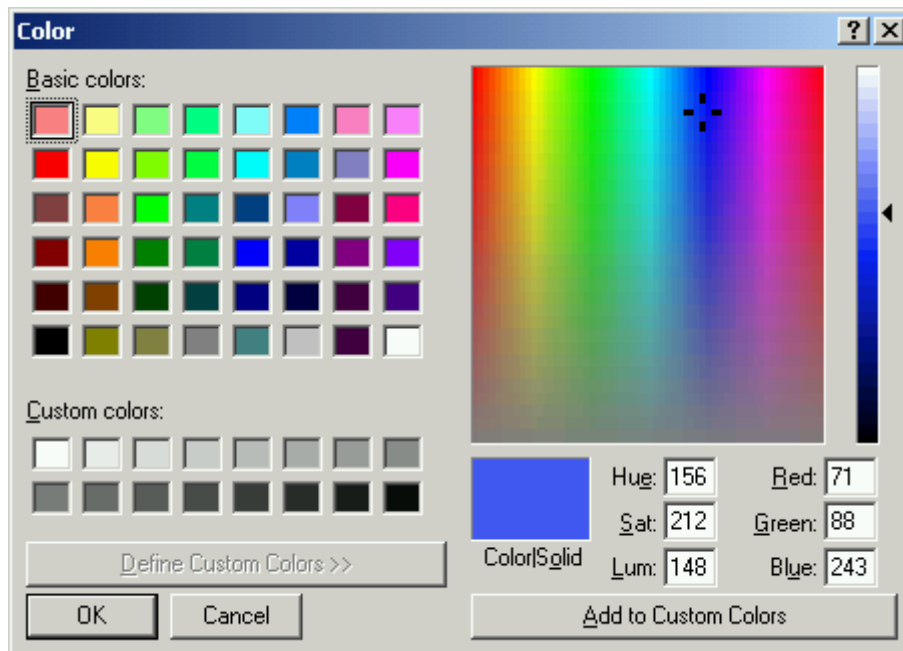
Figure 306: Edit colours

Select one of the **basic colours** or one of the **user-defined colours**. Alternatively you can edit a colour by entering the colour value (colour/saturation/brightness or red/green/blue) or by directly clicking on the colour spectrum. If you click on the **"Add colour"** button, the edited colour will be inserted into the "user-defined colours" and is available for the current ADONIS session.

Click on the **OK button**, to copy the selected colour.

4.1.8.3 Select Enumeration Value

When double-clicking on an attribute of the type "Enumeration" (see chap. 5.2, p. 536) the window "< *object name - attribute name*>" (see fig. 307) containing the available attribute values will be displayed.
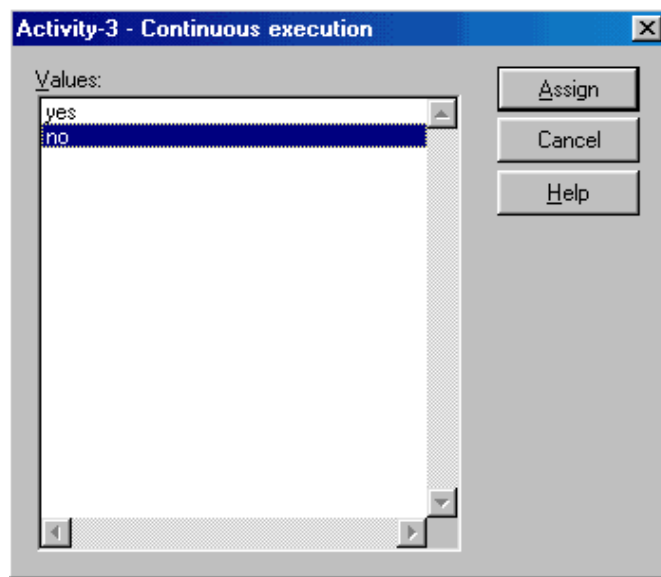
Figure 307: Enumeration

Choose an attribute value from the list **"Value"** and then click on the **"Assign"** button to take over the change.

> **Hint:** The window for the input of an enumeration value (see fig. 307) is only shown in the tabular display.

4.1.8.4 Select Values from an Enumeration List

When double-clicking on an attribute of the type "Enumeration list" (see chap. 5.3, p. 537) the window "*<Attribute profile class name: Attribute profile name - Attribute name>*" (see fig. 308) containing the available attribute values will be displayed.
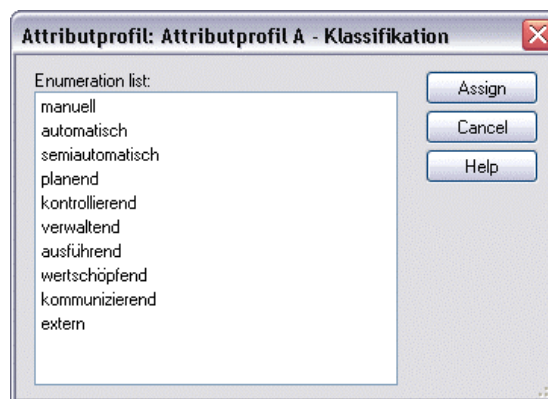
Figure 308: Enumeration list

Select the attribute value required from the field **"Enumeration list"** or mark it and click on the **"Assign"** button to assign the changes to the tabular model display.

> **Hint:** To select several attribute values hold the <Ctrl> key additionally.

4.1.8.5　Enter Program Call

When double-clicking on an attribute of the type "Program call" (see chap. 5.11, p. 539) the window ""<*Attribute profile name - Attribute name*>"" (see fig. 309) showing the current settings will appear.
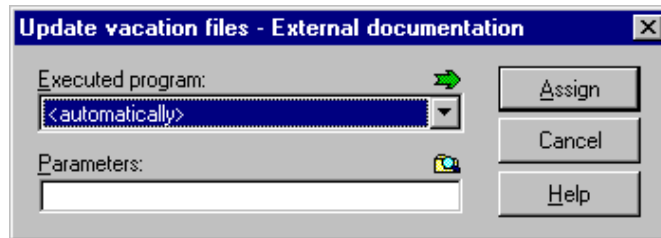
Figure 309: Program call

Select a program in the field **"Executed program"** and optionally enter the path and name of the file that should be opened on calling the program into the field **"Parameters"**. Then click on the **"Assign"** button to confirm the changes.

> **Hint:** The representation of the program call attribute (as in the figure above) may depend on the definition of your application library.
>
> **Instead of the selection list "Executable program"** a **button** with the name of the program or the text "<automatically>" can be shown.
>
> It is possible to fade out the **field "Parameter"**.

4.1.8.6　Edit Date Value

ADONIS provides you with a support dialogue for the assignment of date attributes, which you can open from the ADONIS Notebook by clicking on the "dialogue" button ☐, on the right above the attribute field.

Once you have clicked on the button, the window"<*Object name> - <Name of the date attribute*>" (see fig. 310)opens, where attribute value can be edited.
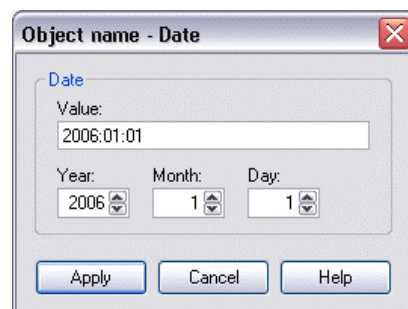
Figure 310: Enter date

The numbers given in the fields "year", "month" and "day" will automatically be converted into the date format (YYYY:MM:DD) and displayed in the field "value". By clicking on the button "Assign", the current value will be transferred to the ADONIS Notebook.

> **Hint:** The attribute type "Time" (see chap. 4.1.8.8, p. 347) is available for the coverage of a time period.

4.1.8.7   Edit Date and Time Value

ADONIS provides you with a support dialogue for the assignment of date and time attributes, which you can open from the ADONIS Notebook by clicking on the "dialogue" button ☐ at the right above the attribute field.

Once you have clicked on the button, the window "*<object name> - <name of the date and time attribute>*" (see fig. 311) will appear, in which you can edit the attribute value.
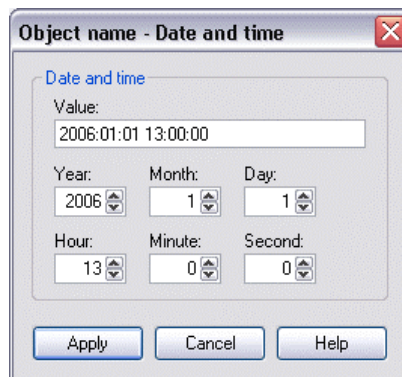
Figure 311: Enter date and time

The numbers given in the fields "year", "month" and "day" will automatically be taken over in the date format (YYYY:MM:DD) and displayed in the field "value". By clicking on the button "Assign", the current value will be transferred to the ADONIS Notebook.

> **Hint:** The attribute type "Time" (see chap. 4.1.8.8, p. 347) is available for the coverage of a time period.

4.1.8.8   Editing Times

ADONIS provides you with a support dialogue for entering time attributes (time period), which you can open from the ADONIS Notebook by clicking on the "dialogue" icon ☐ at the right above the attribute field.

The window "*<object name> - <name of the time attribute>*" (see fig. 312), in which you can edit the attribute value, appears.
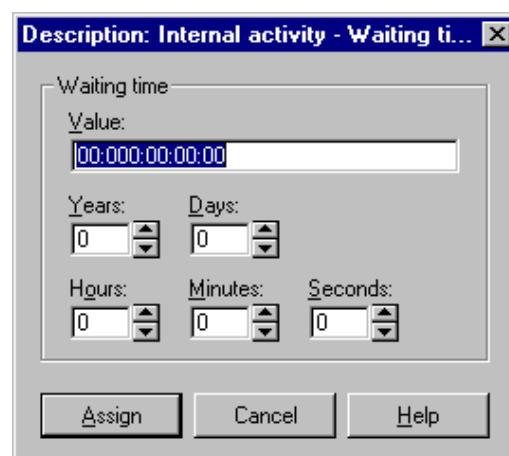
Figure 312: Enter a time period (example "Execution time")

The values entered into the fields "Years", "Days", "Hours", "Minutes" and "Seconds" are automatically transformed into the ADONIS time format (YY:DDD:HH:MM:SS) and displayed in the "Value" field. Confirm the current value by clicking on the "Assign" button.

> **Hint:** The provision of **times** in the models is **not a precondition** for the execution of the evaluation.
>
> However, if this data is not provided the results may be incomplete (e.g. cycle time may be zero).

> **Hint:** The attribute types "Date" (see chap. 4.1.8.6, p. 346) and "Date and time" (see chap. 4.1.8.7, p. 347) are available for the coverage of a time period.

4.1.8.9   Add References

You can add references:

● In the **graphical modelling**, by clicking on the "Add" icon ✚ of the corresponding reference attribute,

● In the **tabular modelling**, by double-clicking on the corresponding cell (in the row of the starting object and in the column of the reference attribute).

All models which can be referenced and are stored in the ADONIS database are available when adding a reference to a model (see p. 348).

It is possible to add a reference to an object (see p. 349) only if the model containing this object is opened.

Add Model References

In the window "<*Object name (class name) - Attribute name*> - Add reference" (see fig. 313), the model group hierarchy is displayed with all models you can reference.
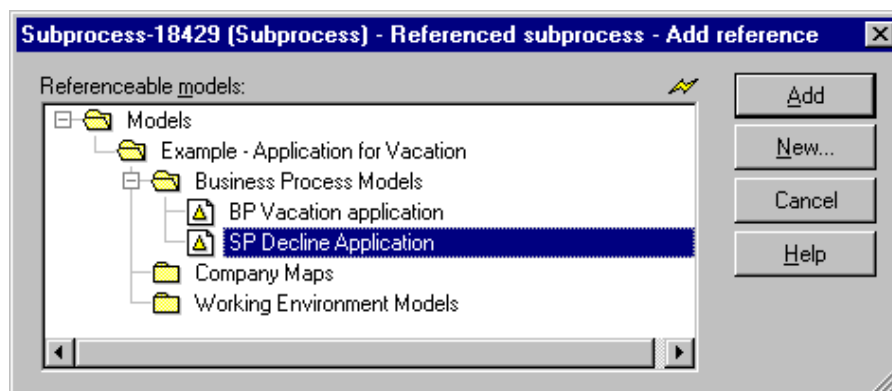
Figure 313: Add model reference

Select the model to refer to reference and click on the **"Add"** button. The window will be closed and the name of the referenced model will be registered in the text field of the ADONIS Notebook.

By clicking on the icon ⬈, you can update the "models that can be referenced" list.

Add Object References

The reference targets are shown in the window "*<object name (class name) - attribute name>* - Add reference" (see fig. 314).
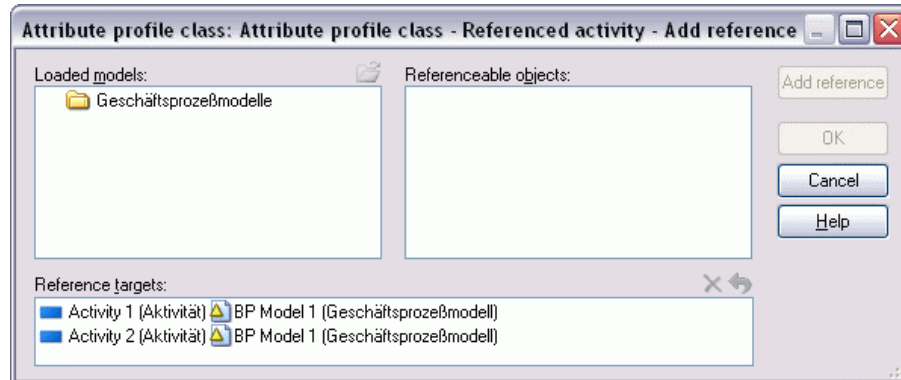

Figure 314: Add object reference

ATTENTION: In the Attribute Profile Management in the Administration Toolkit, you can neither add object references nor open models . The button "Add" and the "open" icon are not available.

You can delete the existing object references shown in the "Reference targets" list, by selecting the reference to delete and clicking on the "Delete" icon ✖.

If you want to undo the changes carried out, click on the cancel icon ↩ above the list of the reference targets.

4.1.8.10 Random Generator

A Random generator sets a value to a variable to which it is connected. The value depends on an expression.

Hint: ADONIS provides you with a support dialogue (see p. 351) for the definition of these expressions

The syntax of the assignment expressions (*ValAssign*) is based on the following grammar:

ValAssign ::= `MathExpr { ; MathExpr }`

MathExpr ::= `Term | ( MathExpr ) | Term Operator MathExpr`

Term ::= `String | Numeric | Variable | Distrib`

Operator ::= `+ | - | * | /`

● **Terms:**

In the simplest case an assignment expression consists of a single term. There are three possibilities when defining a term:

A term can be a **constant**. Depending on the type of the variable this can either be a constant of the type **"enumeration"** - e.g. 'standard case' - or of the type **"float"** - e.g. 365.

A term can also be **another variable** - e.g. X - of the same type. So the value of the variable can be assigned to another variable.

A term can also be a **distribution** (see p. 350). For variables of type enumeration a distribution type - the discrete distribution - is defined in ADONIS. Three different distribution types exist for variables of type float: exponential, uniform and normal distribution.

● **Arithmetical expressions:**

Using the terms listed above, complex **arithmetical expressions** (*MathExpr*) can be formed. Terms of type float can be combined with the arithmetical operators +,-,* and brackets. The operators * and / take priority over the operators + and -. Terms of type enumeration can be combined using the operator + and brackets if necessary.

● **Separating a list of assignment expressions:**

It is also possible to use more than one expression, by separating them with a semicolon. During the first simulation run, the first expression will be used, during the second run the second, and so on. When there are more runs than expressions, the last expression will be used repeatedly.

Examples for valid assignment expressions:

Some examples of valid expressions are now listed. X and Y are variables of type float, while S and T are variables of type enumeration.

Uniform(4; 10) : the variable is assigned with a random value, based on the distribution used in the example.

X+1 : the variable is assigned the value X plus 1.

X+Uniform(4; 10) : the variable is set to the value of the variable X increased by a random value, based on a uniform distribution.

X*Exponential(4; 10)+Y : the value of the variable X is multiplied by a random value (based on an exponential distribution), to this product the variable Y is added.

S+'a' : the character 'a' is added to the string S and the result is assigned to the variable.

'a';S+'b';S+'c' : during the first run the variable is assigned the character 'a'. During the second run the character 'b' is added to the variable S and the result assigned to the variable. During the third and following runs the character 'c' is added to the variable S and the result assigned to the variable.

Variables

Variables will be assigned (see chap. 4.1.8.10, p. 349) with values in ADONIS Business Process Models through assignment objects and will be queried during the process. This way, after branching in the process, paths can be passed according to the assignment of variables - defined in the transition conditions.

The display of variables depends on the definition in your application library. In the ADONIS BPMS Application Library, variables will be shown over objects of the class "variables".

> **ATTENTION:** The <u>variable names</u> shall <u>not</u> start with **numerals from 0 to 9**, the **figures + - : = < > * / ( ) . , '** as well as a **blank** or a **return**. Moreover the **figures + - : = < > * / ( ) . , '** as well as **blanks** or **returns** are <u>not</u> authorised in variable names. The key words **AND**, **OR**, **NOT** and **TRUE** are <u>not</u> possible for variable names.

Distribution

Variables are assigned (see chap. 4.1.8.10, p. 349) with values in ADONIS using objects of type Random generator. This can be done using distributions. For variables of type float three **continuous dis-**

tribution functions are offered: exponential, uniform and normal distribution. For variables of type enumeration a **discrete distribution** is available.

The syntax for **continuous distribution** is:

● for **normal distribution**: <u>normal (<number1>;<number2>)</u>

Enter the expected value and the standard deviation for the normal distribution, where <number1> represents the expected value and <number2> stands for the standard deviation.

Example: `Normal(1200;100)` The variable has a normal distribution with an expected value of 1200 and a standard deviation of 100.

● For **exponential distribution**: <u>exponential (<number>)</u>

Enter the expected value of the exponential distribution, letting <number> be 1 divided by the expected value.

Example: `Exponential(0,002)` The variable has an exponential distribution with an expected value of 500.

● For **uniform distribution**: <u>uniform (<number1>;<number2>)</u>

Enter the boundaries for the uniform distribution, where <number1> indicates the lower boundary and <number2> the upper boundary.

Example: `Uniform(0;100)` The variable is uniformly distributed between the boundaries 0 and 100.

The syntax for a **discrete distribution** is:

<u>Discrete (<Symbol1> <number1>;<Symbol2> <number2>; ...)</u>

You can define two or more symbols with their corresponding probabilities (number1, number2,...). The sum of the probabilities must always equal one!

> **ATTENTION:** The entry of symbol names is case-sensitive. The symbol names shall <u>not</u> **start** with **numerals from 0 to 9**, **blanks** or **returns** and the **figures : ( ) . , ; '**. Also **blanks** and **returns**, as well as the **figures: ( ) . , ; '** are <u>not</u> authorised in symbol names.

Examples:

Discrete (YES 0.6;NO 0.4):
> The variable is assigned with a probability of 0.6 of being 'YES' and a probability of 0.4 of being 'NO'. Therefore two transition conditions <variable name>='YES' and <variable name>='NO' are valid and possible.

Discrete (a 0.5;b 0.3;c 0.1;d 0.1):
> The variable is assigned with a probability of 0.5 of taking the value 'a', a probability of 0.3 of being 'b', a probability of 0.1 of taking the value 'c' and a probability of 0.,1 of taking the value 'd'. The four possible transition conditions therefore are: <variable name>='a', <variable name>='b', <variable name>='c' and <variable name>='d'.

Support Dialogue for Distributions

ADONIS provides you with help to enter a distribution in the ADONIS Notebook of objects of class "Random generator". This is found by clicking on the dialogue button next to the field "Value" ☐ . The window "<*class name*>: <*object name*> - Value" (see fig. 315) will appear.
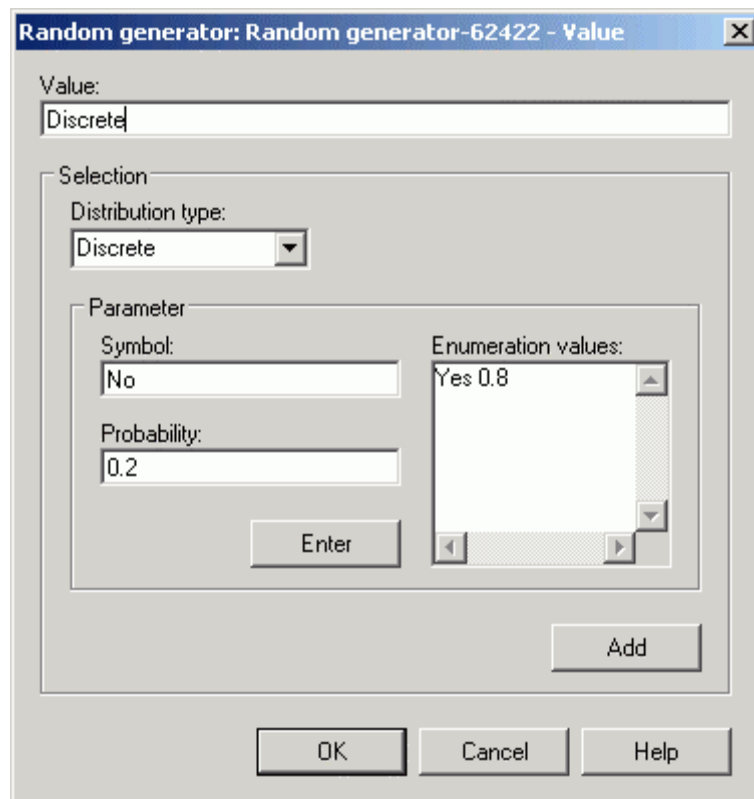
Figure 315: Variable assignment

The current distribution (see chap. 4.1.8.10, p. 349) can be found in the field **"value"**.

You can add distributions to any position within the **"Value"** field. Positioning the cursor within this field at the position at which you would like to place a new distribution does this. If you do not explicitly set the position of the cursor, then a new distribution will be added at the end of the field (separated by a';').(Please note that the first time you add a new distribution to a Random generator it will over-write the default distribution).

Choose the distribution (see p. 350) from the list box **"Distribution type"**. You can choose from the following distribution types:

● **"Discrete"**: discrete distribution function.

● **"Exponential"**: continuous distribution function, exponential distribution.

● **"Uniform"**: continuous distribution function, uniform distribution.

● **"Normal"**: continuous distribution function, normal distribution.

Depending on the distribution chosen you will be asked to enter different parameters.

By clicking the OK button the expression in the field **"Value"** will be syntactically checked and copied into the ADONIS Notebook.

ATTENTION: Arithmetical expressions (e.g. "X + 1") must be directly entered into the ADONIS-notebook

4.1.8.11 Assigning and Defining Sub Processes

In the simulation of business processes, all referenced sub processes in the Business Process Model must refer to existing sub processes . (When using hierarchical Working Environment models, the organisational units in the Working Environment model must refer to existing sub models.)

Sub processes enable you to keep your business process models clearly structured or to reuse models.

ADONIS supports the following proceedings to model process models through the assignment and definition of sub processes in referenced sub process models:

- **Bottom-up modelling (see p. 353)** and

- **Top-down modelling (see p. 353)**.

The assignment of sub processes is done using a reference attribute (e.g. "called process"), into which you will add a reference to a model (see chap. 4.1.8.9, p. 348). This reference attribute will be appropriately defined in the application library.

> **Hint:** Using sub processes will help you make your models more readable. Whenever you model sub processes, take care to design them in such a way that they are self-contained parts of the whole model. You should also use sub processes when describing routine or common processes, so that a common process need only to be modelled once ("reusability").

> **Hint:** ADONIS also allows recursive calls when processes are assigned to an object of the class "Subprocess", i.e. a process may call itself. Before you use such a construction, check it thoroughly. It is essential to have a suitable condition for exiting a recursive loop. When a recursive process call occurs, the system will therefore ask you, if you really want the process to call itself.

Bottom-up Modelling

Bottom-up modelling means that you start the model by modelling the sub processes (the processes of the lowest level), while modelling the higher processes later on. This means that you model your business process models from bottom to top.

Top-Down Modelling

Top-down modelling means that you begin modelling at the highest level (the main process). The main process is modelled completely. Possible sub process calls are integrated but the corresponding sub processes are modelled at a later stage. Your business process model is therefore modelled from top to bottom.

> **Hint:** In the ADONIS BPMS Application Library (see chap. 20., p. 693), the top-down Modelling will be additionally supported through the function "Transform", using which an object of the class "Activity" in Business Process Models can be transformed to an object of the class "Subprocess" .

4.1.8.12 Assigning Performers

Business process models and working environment models are linked on the one hand by defining application models and, on the other hand by assigning performers to activities. Such links enable the capacity and Workload simulations.

Assigning performers means that one or a group of performers are entered into the attribute "Performer" of each object of the class "Activity". These performers can execute the particular activity

within the simulation. The performers are assigned to activities by AQL expressions (AQL = ADONIS Query Language (see chap. 12., p. 593)).

ADONIS provides a support dialogue for assigning performers in objects of the class "Activity". Open this dialogue window by clicking on the "Dialog" button ⬜ above the attribute field in the notebook.

The window "<*class name*>:<*object name*> - Performer" (see fig. 316), in which you can define the activity's performer, is opened.
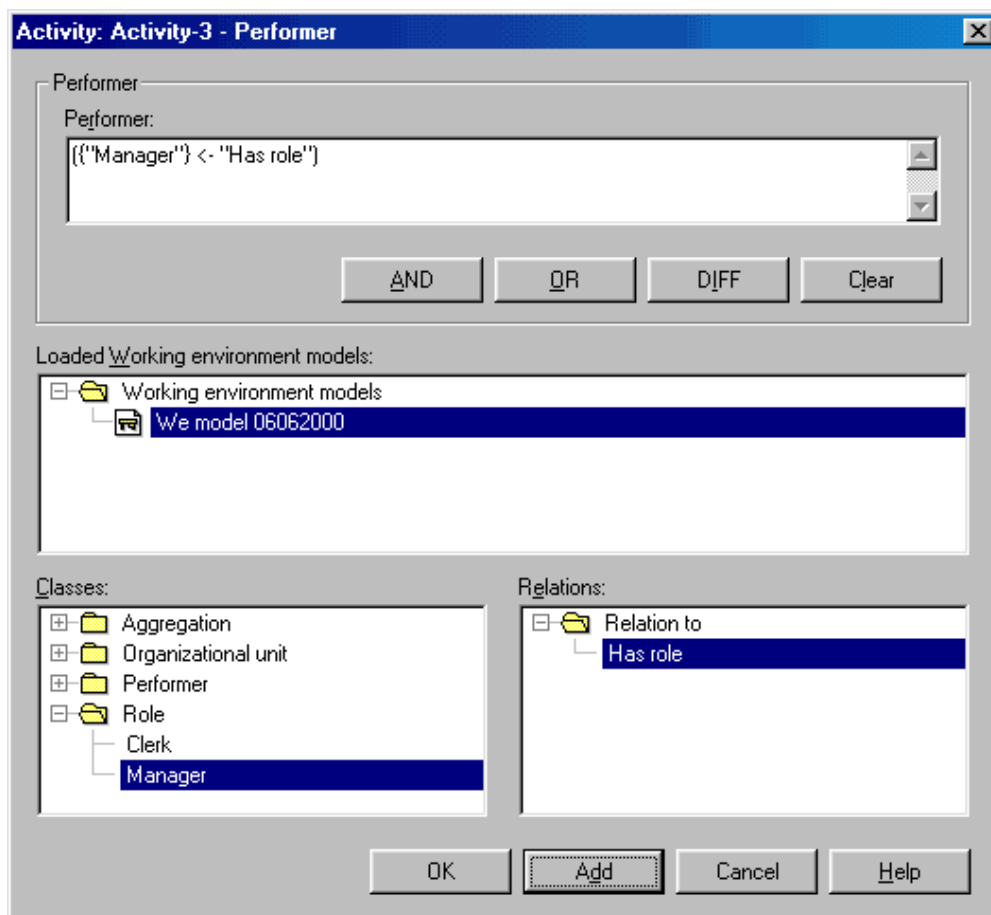

Figure 316: Assigning performers

ATTENTION: In the Attribute Profile Management in the Administration Toolkit, you can define the perfomer assignment expression exclusively in the field "Performer" by entering an AQL expression (see chap. 12., p. 593). The definition in the Working Environment models is not possible as the button "Add" is not available.

The buttons "**AND**", "**OR**", "**DIFF**" enable you to combine several AQL (see chap. 12., p. 593) expressions (self-defined or added). Delete the current AQL expression in the field **"Performer"** by clicking on the button **"Clear"**. It is possible to type the required AQL expression directly into the Performer field.

Clicking on the OK button causes the system to check the expression's syntax and then enter it into the object's notebook.

Clicking on the "Cancel" button closes the window "Performer assignment" while maintaining the previous value of the "Performer" attribute.

Assigning performers from hierarchical working environment models

If your current application library supports hierarchical Working Environment models (see chap. 4.1.8.11, p. 353), the position of the performer must be specified more precisely (see fig. 317) once you have clicked on the button **"Add"** (see fig. 316).
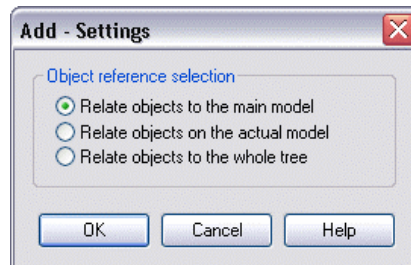
Figure 317: Add a performer from a Working Environment hierarchy

Select the option **"Refer objects to the main model"**, if the performer has to be searched for in the main model of the model hierarchy. The main model is the Working Environment model, which was given during the definition of the application model (see fig. 318).

Select the option **"Refer objects to the current model"**, if the performer has to be searched for in the currently selected model of the field "Loaded Working Environment models" (see fig. 318).

Select the option **"Refer objects to the whole tree"**, if the performers have to be searched for in the whole model hierarchy (see fig. 318).
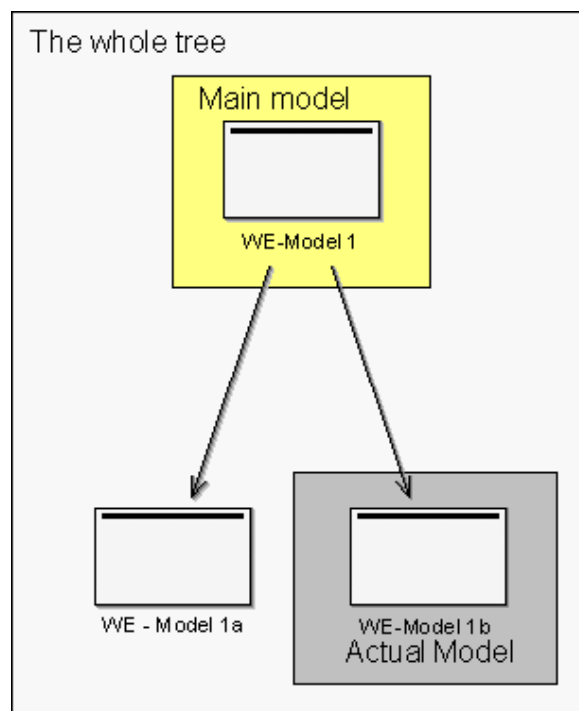
Figure 318: Terms in the hierarchical Working Environment models

Assigning performers to sub processes

The performer assignment is - according to an appropriate definition in your application library - also possible in sub process objects.

This way you can define a "standard performer assignment expression", which will be analysed during the simulation, if no performer has been assigned to the activities of the referenced process.

<u>**Example:**</u>

In the following example   (see fig. 319), three Business Process Models (main model, referenced model and referenced model of the second level) are represented. Additionally the performers of the activities are shown.
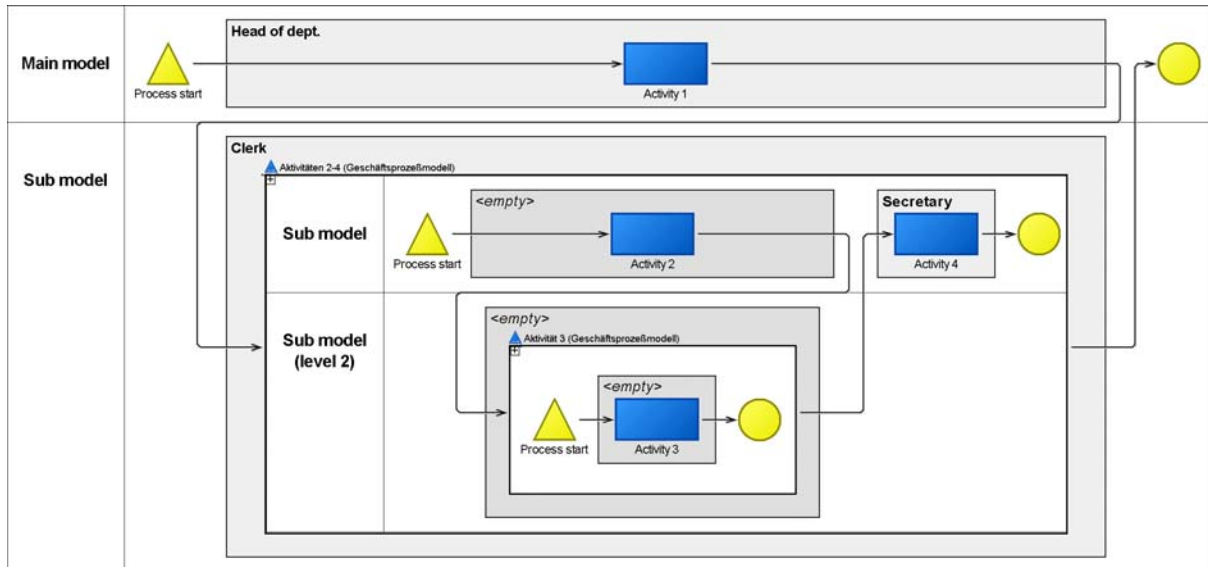
Figure 319: Assigning performers to subprocesses

In the main model the performer expression "Responsible person" will be defined in the sub process object of a submodel. In the sub model, no performer is defined in the activity "Activity 2", i.e. the standard value from the sub process object in the main model will be used during the simulation and so assigned to the "Activity 2" of the performer "Responsible person".

No performer is assigned in the activity of the sub model of the second level. During the simulation, the standard value from the sub process object will be used in the submodel. Since this standard value is also not assigned in the submodel, the standard value from the sub process object in the main model will be used and so assigned to the "Activity 3" of the performer "Responsible person".

Defining Probabilities

In the simulation algorithms Capacity Analysis and Workload Analysis, the performer assignment expression will be evaluated and the activity will be randomly (uniformly distributed) assigned to one of the resulting performers. The standard uniformly distributed random selection of the performer can however be influenced through the entry of probabilities in the performer assignment expression.

For this, adhere to the following syntax:

```
{ <AQL expression> <Probability>; }
<AQL expression> <Probability>
```

Example:

An activity should be assigned with a probability of 20% to a performer of the role "Clerk" and with a probability of 30% to a performer of the role "Secretary" and with a probability of 50% to a performer of the role "Temporary help".

Give the following performer assignment expression:

```
({"Clerk": "Role"} <- "has role") 0.2;

({"Secretary": "Role"} <- "has role") 0.3;

({"Temporary help": "Role"} <- "has role") 0.5
```

> **ATTENTION:**   The single AQL expressions (see chap. 12., p. 593) will be separated by the figure **';'** . **There shall be no ';' at the end of the whole expression!**

> **Hint:** The sum of the probabilities must equal 1.

4.1.8.13   Allocating Resources

Allocating resources means that one or more actual resources are assigned to each object of the class "Resource" via the "Selection" attribute. These resources are required when executing an activity. Allocated resources are evaluated in the Simulation Component during Capacity Analysis and Workload Analysis.

The allocation of resources is described by AQL expressions (AQL = ADONIS Query Language (see chap. 12., p. 593)).

ADONIS provides a support dialogue for the allocation of resources. Call this support dialogue by opening the ADONIS Notebook of an object of class "Resource" in a business process model and then clicking on the "Dialog" button ☐ next to the input field "Selection". The window "*<class name>*:*<object name>* - Selection" (see fig. 320) appears.
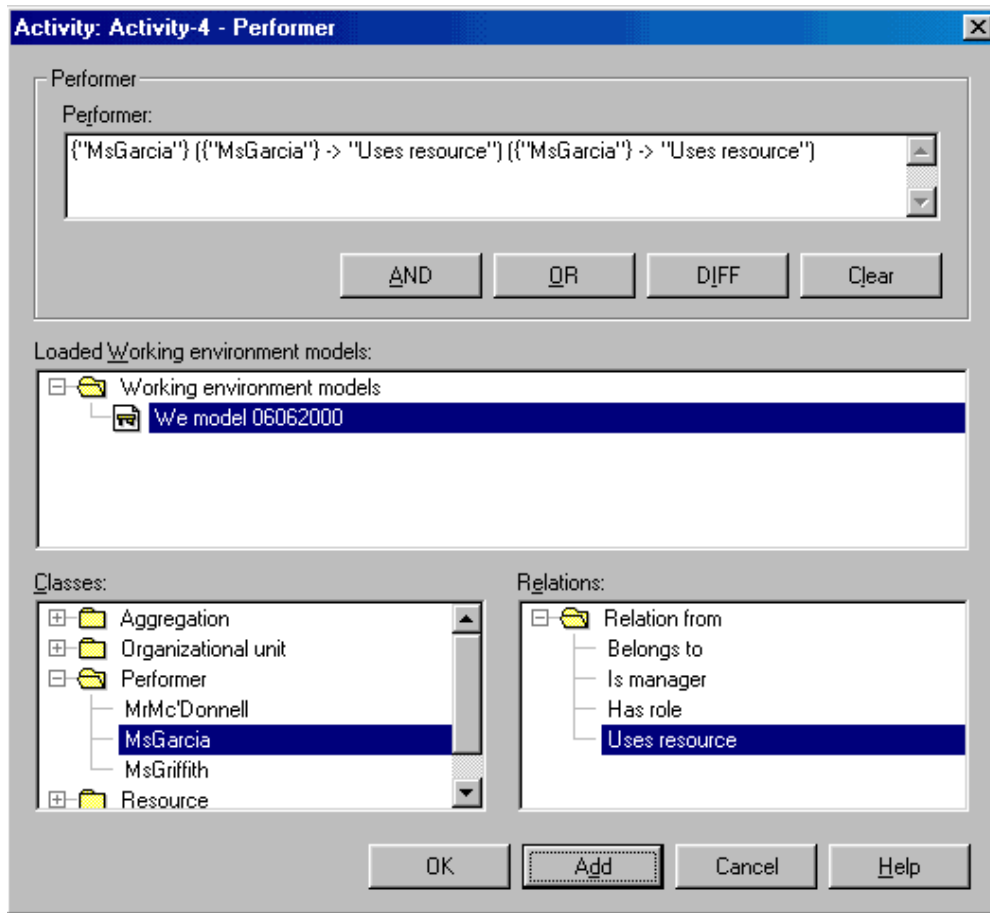
Figure 320: Allocating resources

ATTENTION: In the Attribute Profile Management in the Administration Toolkit, you can only define the resources assignment expression in the field "Resources" by entering a valid AQL expression (see chap. 12., p. 593). This definition is not possible in Working Environment models and the button "Add" is not available.

ATTENTION: Resources can only be allocated in business process models. The resources in the working environment models represent resources, which actually exist in the company. The resources in the business process models are references to the resources in the working environment model and represent the resources used while a business process is performed.

The field **"Resource"** holds the current resource expression. Before a simulation runs, this expression is the attribute value entered into the attribute "Selection" of the ADONIS Notebook from which the dialogue has been called.

The buttons "**AND**", "**OR**", "**DIFF**" enable you to combine several AQL expressions (see chap. 12., p. 593). You can delete the current AQL expression in the **"Resource"** field by clicking on the button **"Clear"**.

When defining a resource the expressions entered are displayed in the **"Resource"** field. Clicking on the OK button causes the system to check the expression's syntax and to copy it to the object's notebook.

Clicking on the "Cancel" button closes the window "Resource Selection" while maintaining the previous value of the attribute "Selection" in the notebook.

4.1.8.14   Performer Calendar

The performer calendar allows you to define working time profiles for objects of class "Performer". This calendar is evaluated by the "Workload Analysis" and specifies, for a one year period, the presence of a performer.

> **Hint:** The performer calendar is a standardised calendar. This means that it does not refer to a specific year and that it comprises **365 days**. Leap years are not considered and the first day is always a Monday, the 1st of January.

Open the calendar by clicking on the button **"Calendar"** in the "simulation data" chapter of the ADONIS Notebook of an object of class "Performer".

The window "<performer's name> - Calendar" (see fig. 321, p. 359) appears.

Display Performer's Calendar

The window "*<performer's name>* - Calendar" (see fig. 321) shows the current working time profile of the performer selected. The default setting for the class "Performer" is that Monday to Friday are working days, Saturday and Sundays are days off (bank holidays, vacation etc. are not taken into account).
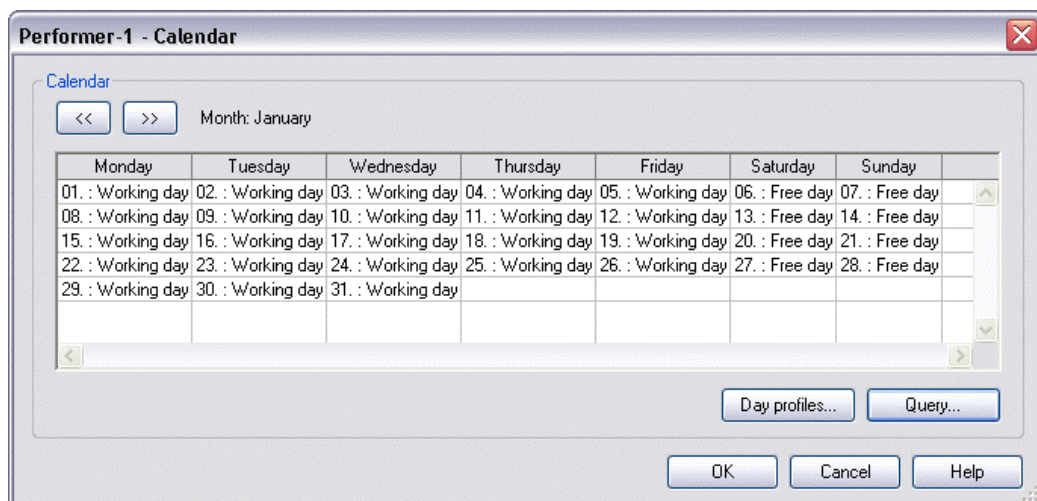
Figure 321: Performer's calendar

The working time profiles are displayed for one month. You can show the previous and following months by clicking on the buttons **"<<"** or **">>"** respectively. The name of the current month is shown to the right of the two buttons.

If the names of the day profiles are too long to fit in the standard column width of the calendar, you can adjust the column width accordingly. Double-click on the column's header (e.g. "Monday"). The width of the corresponding column will be set in such a way that all day profile names fit into the column.

Click on the OK button to close the calendar and to confirm the changes made.

In addition to the existing day profiles "Working day" and "Free day" you can define new day profiles (see p. 360) and change (see p. 361) or delete (see p. 362) existing profiles. By clicking on the **"Day profiles"** button, the window "Day profiles" (see fig. 322) listing all available day profiles will appear.

Click on the button **"Query"** if you wish to calculate the actual availability of the performer (see p. 364).
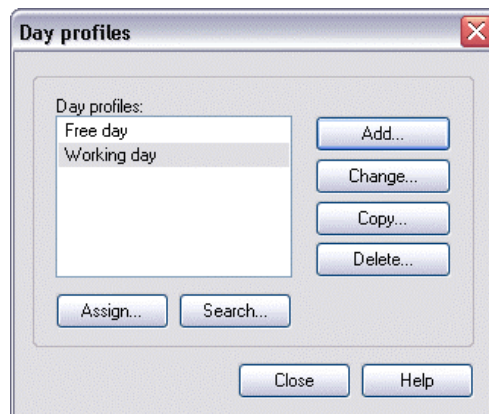
Displaying Day Profiles

Figure 322: Displaying day profiles

The window "Day profiles" (see fig. 322) enables you to:

- Add new day profiles (see p. 360).

- Change existing day profiles (see p. 361).

- Copy day profiles from other objects of the class "Performer" (see p. 363) - also from other models.

- Delete existing day profiles (see p. 362).

- Assign existing day profiles to days or periods of time (see p. 362).

- Search for days to which day profiles have already been assigned (see p. 364).

Adding Day Profiles

You can add new day profiles by clicking on the button **"Add"** in the window "Day profiles" (see fig. 322, p. 360) . The window "New day profile" (see fig. 323) opens up, into which you must enter a unique name for the new day profile (a name which has not yet been used for another day profile).

Figure 323: New day profile

Click on the OK button or press the enter key to continue the process. Clicking on the "Cancel" button closes the window and returns to the window "Day Profile".

Once you have entered a unique name for the day profile to be added and clicked on the OK button, the window 'Day profile "<name of the day profile>" - time intervals' (see fig. 324) appears.

Figure 324: Time intervals

Click on the **"Add"** button and enter the start and end time of the interval in the window "New interval" (see fig. 325).
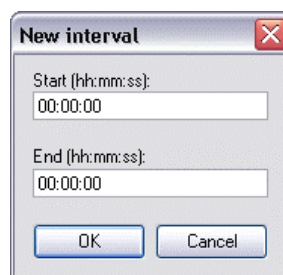
Figure 325: Add interval

Please make note of the following points when entering the interval:

- Time intervals must be entered in the time format "**hh:mm:ss**".

- The starting time must be **earlier** than the ending time.

- If more than one time interval is defined for a day profile, the intervals **may not** overlap.

After entering the interval, click on the OK button to copy the new interval to the list of working times in the window 'Day profile "<name of the day profile>" - to copy time intervals' (see fig. 324, p. 361).

Changing Day Profiles

If you wish to change an existing day profile, select the appropriate profile in the "Day profiles" (see p. 360) window and then click on the button **"Change"**. The window 'Day profile "*<name of the day profile>*" - time intervals' (see fig. 326) appears, showing the intervals already defined in the field **"Working intervals"**.
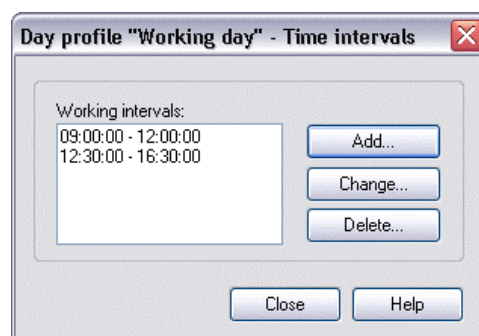
Figure 326: Changing day profiles

If you wish to **change an interval**, select the interval to be changed and then click on the **"Change"** button.

Enter the new starting time and/or the new ending time in the window "Change interval".

Please make note of the following points when entering the interval:

● Time intervals have to be entered in the time format "**hh:mm:ss**".

● The starting time must be **earlier** than the ending time.

● If more than one time interval is defined for a day profile, the intervals **may not** overleap.

After entering the interval, click on the OK button to copy the new interval to the list of working times in the window 'Day profile "<name of the day profile>" - to copy time intervals'.

To **delete an interval**, select the appropriate interval in the window 'Day profile "*<name of the day profile>*" - time intervals' and click on the **"Delete"** button.

A message asking you if you really wish to delete the time interval appears.

Deleting Day Profiles

If you wish to delete an existing day profile, select it by a mouse click in the window "Day profiles" (see fig. 322, p. 360) and then click on the button **"Delete"**.

A window appears, in which you have to confirm that you really wish to delete the day profile selected.

Click on the "Yes" button to continue deleting the profile. Click on the "No" button to stop the process.

Assigning Day Profiles

You can assign the day profiles defined to certain days and/or periods of time. There are two ways to do this:

● For **a certain day**.

● For **certain days of the week and/or periods of time during a year**.

In order to assign a day profile to **a specific day**, double-click on the particular day in the window "*<performer's name>* - Calendar" (see fig. 321, p. 359). The window "<Date>" (see fig. 327), showing the current day profile and the available (assignable) day profiles, are displayed.
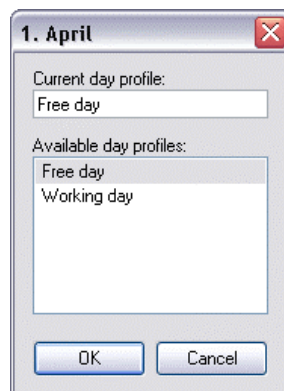
Figure 327: Assign profile for a certain day

Select the day profile to be assigned from the list **"Day profiles"** and click on the OK button. The selected day profile will then be assigned. Clicking on the "Cancel" button cancels the process.

If you wish to assign a day profile to **certain days of the week and/or periods of time during a year**, select the day profile to be assigned in the window "Day profiles" (see fig. 322, p. 360) and then click on the "Assign" button. The window "<name of day profile>" - assign to days' (see fig. 328) opens.

Figure 328: Assign day profiles

Activate the option **"Day"** and choose the day that you want to assign the day profile to. If you click on the OK button, the current day profile is assigned to all days (within the year) of the type selected.

You can define a period of time by selecting the option **"Time interval"**. After you have entered the period, click on the OK button to assign the current day profile to all days within the period selected.

In addition, you can form a set of days the current day profile will be assigned to by activating the options "Day" **and** "Time interval". The following options are at your disposal:

- **Intersection**: AND operator (standard setting); All days identical to the day of the week selected within the period defined will be determined (e.g. all Fridays from March, 1st to April, 30th).

- **Union**: OR operator; All days identical to the day of the week selected **or** within the period of time specified are determined (e.g. all Fridays of a year and all days from March, 1st to April, 30th).

After you have selected the day of the week, entered the period of time and the chosen operator, click on the "Assign" button to update the days in the calendar.

A preview of the days selected (see p. 363) can be seen before you assign the current day profile to those days. Click on the **"Test"** button to call this preview.

Show Specified Days

A preview of the days specified can be seen in the window "Specified days".

It is possible to save the content of the window "Specified days" to a file or print it out.

Copying Day Profiles

If you wish to copy an existing day profile, go to the "Day profiles" (see fig. 322, p. 360) window, select the day profile and click on the **"Copy"** button.

The window "Copy day profiles" (see fig. 329) appears. There you can select a reference object (specific performer, who has the same working time or the same days off) from another opened working environment model as well as the day profiles to be copied.
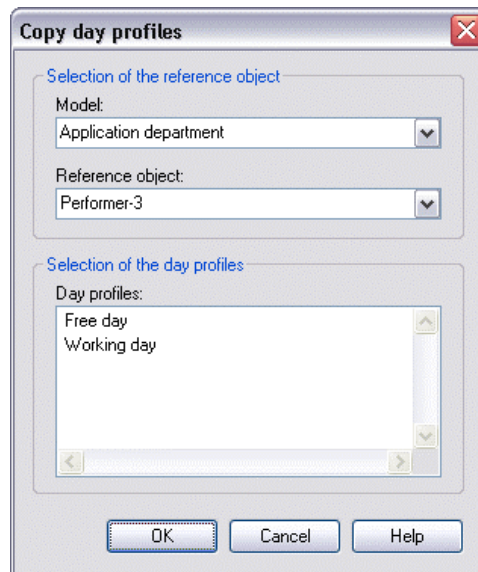

Figure 329: Copy day profiles

Confirm the day profile(s) to be copied by clicking on the OK button or close the window by clicking on the cancel button.

The window "Day profiles" will appear again, containing the copied day profile.

Searching for day profiles

If you wish to see all the days to which an existing day profile has been assigned, click on the respective day profile in the window "Day profiles" (see fig. 322, p. 360) and then click on the "Search" button.

The window 'Assignment of the day profile "<name of the day profile>"' lists all days in chronological order to which the previously selected day profile has been assigned.

You can save the contents of the window 'Assignment of the day profile "<name of the day profile>"' to a file or print it out.

Calculating the Time of Presence

The evaluation of the performer calendar enables you to calculate the time of presence (quantity of time during which a particular performer is actually available for work) of the respective performer during a certain period of time, based on the calendar defined.

Clicking on the button "Query" in the window "*<performer's name>* - Calendar" (see fig. 321, p. 359) opens the window "Calculate time of presence" (see fig. 330).

Figure 330: Calculate the time of presence

Specify the time interval and the result format and click on the "Query" button. The time of presence is shown in the format selected in the field **"Working time"**.

4.1.8.15   Process Calendar

The process calendar offers you the possibility to define occurrence profiles for every business process. The calendar is evaluated by the simulation algorithm "Workload Analysis" and states the period of occurrence and the probability of occurrence of a given business process referring to one year.

> **Hint:** The process calendar is a standardised calendar. This means that it does not refer to a specific calendar year and consists of **365 days**. Leap years are not considered and the first day is always Monday, 1st of January.

Click on the button **"Process calendar"** in the chapter "simulation data" in the ADONIS Notebook of an object of the class "Process start" to open the process calendar.

When clicking on the button the window "<process start name> - Process calendar" will appear.

Display Process Calendar

The window "*<process start name>* - Process calendar" (see fig. 321) presents the current occurrence profile of the process selected. The default setting for the class "Process start" is that the days from Monday to Friday are defined as working days, Saturdays and Sundays are days off (bank holidays etc. are not taken into account).

The working time profiles are displayed for one month. You can view the previous and following months by clicking on the buttons **"<<"** or **">>"** respectively. The name of the current month is shown to the right of the two buttons.

If the names of the day profiles are too long to fit in the standard column width of the calendar, you can adjust the column width accordingly. Double-click on the column's header (e.g. "Monday"). The width of the corresponding column will be set in such a way that all day profile names fit into the column.

Click on the OK button to close the calendar and to confirm the settings.

Click on the "Cancel" button to close the calendar and to discard any changes.

If you click on the cancel button, after making some changes to the calendar, a second message will appear asking you to confirm that you definitely want to exit without saving your changes. Click on the "Yes" button to close the calendar without executing any changes. You can return to the window "Process start: <process start name> -- calendar" by clicking on the "No" button.

In addition to the existing day profiles "Working day" and "Free day" you can define new day profiles, change or delete them. By clicking on the **"Day profiles"** button, the window "Day profiles" (see fig. 322) listing all available day profiles (see p. 366) will appear.

Click on the **"Query"** button to calculate the number of processes (see p. 369).

Day Profiles

The following functions can be carried out in the "Day profiles" (see fig. 322) window:

- Add (see p. 366) new day profiles.

- Change (see p. 366) existing day profiles.

- Copy (see p. 369) day profiles from other objects of the class "Process start", also from other models.

- Delete (see p. 368) existing day profiles.

- Assign (see p. 368) existing day profiles to days and/or time periods.

- Search (see p. 369) for days already assigned to an existing day profile.

Adding Day Profiles

You can define new day profiles by clicking on the **"Add"** button in the window "Day profiles". The window "New day profile" (see fig. 323) appears, in which you must enter a unique name for the new day profile (a name that has not yet been used).

Click on the OK button or press the Enter button to continue generating the new day profile. Clicking on the Cancel button interrupts the process and returns you to the window "Day profiles".

Once you have entered a unique name for the day profile to be added and have clicked on the OK button, the window Day profile appears "<name of the day profile>" - time intervals".

Click on the **"Add"** button. The window "New interval" will be displayed.

Changing Day Profile

If you wish to change an existing day profile, select the appropriate profile in the window "Day profiles" and then click on the **"Change"** button. The window 'Day profile "*<name of the day profile>*" - time intervals' (see fig. 331) appears, showing the intervals already defined in the field **"Intervals"**.
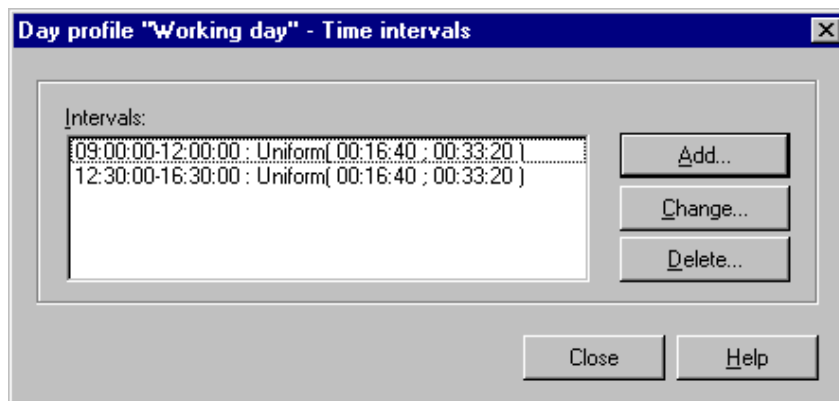
Figure 331: Change day profiles

In this window you can:

- Add (see p. 367) a new occurrence (or frequency) interval.

- Change (see p. 368) an existing interval of occurrence.

- Delete (see p. 368) an existing interval of occurrence.
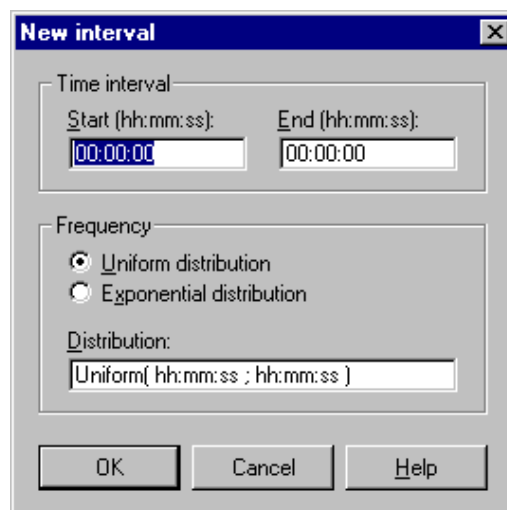

Adding Intervals

Figure 332: Adding intervals

Enter the start and end time of the interval into the field **"Time interval"** (see fig. 332). When doing this, consider the following points:

- Time intervals must be entered in the time format "**hh:mm:ss**".

- The starting time must be **earlier** than the ending time.

- If more than one time interval is defined for a day profile, the intervals **may not** overlap.

Define the probability of occurrence within the time interval in the field **"Frequency"** (see fig. 332). You can choose between the following:

- **Uniform distribution**: A business process' probability of occurrence within the given interval is distributed uniformly between a lower and upper boundary (period of time).

Example:

If you enter the frequency "Uniform (00:30:00; 02:00:00)", the business processes will be triggered from every 30 minutes (minimum) to every two hours (maximum).

● **Exponential distribution**: A business process' probability of occurrence within the interval given is distributed exponentially with the expectancy 1/E (E = period of time in seconds).

Example:

If you enter the frequency "Exponential (0,00027778)", the business processes will be triggered with an expectancy of 1/3600 (corresponds to one hour) for the intervals of time.

After you have entered the interval and the frequency, click on the OK button to copy the new interval into the list of working times in the window 'Day profile "<name of the day profile>" - time intervals'.

Changing Intervals

If you wish to change an interval, select the respective interval and click on the **"Change"** button. The window "Edit interval" (see fig. 333) appears.
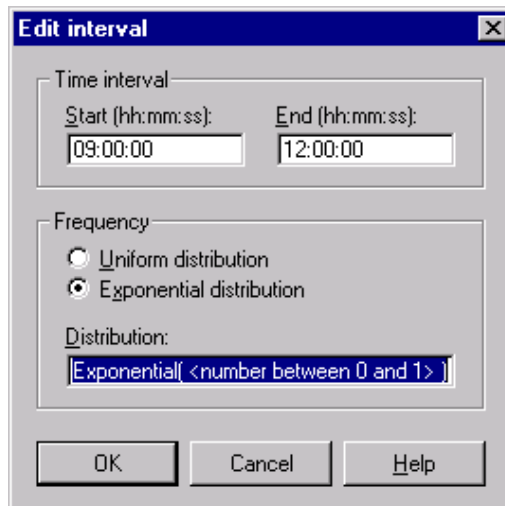
Figure 333: Edit interval

Deleting Intervals

In order to delete an interval, select the interval to be deleted and then click on the **"Delete"** button. The system will then ask you if you really want to delete the respective interval.

Deleting Day Profile

You will be asked to confirm the deletion of the day profile.

Assigning Day Profiles

Displaying Specified Days

Please refer to the information provided for the performer calendar (see p. 363).

Copying Day Profiles

As well as defining new day profiles for each object of the class "Process start" you can also copy the existing day profiles into other "Process start" objects. Click on the **"Copy"** button in the window "Day profiles" (see fig. 322). The window "Copy day profiles" (see fig. 329) will be displayed.

A day profile is copied according to the three steps described below:

1. Select from the list **"Model"** the business process model that contains the object of the class "Process start" whose day profile you wish to copy. (**Note**: The list only contains models, which are currently open)

2. Select the reference object of the class "Process start" from the list **"Reference object"**, i.e. that object of the class "Process start" whose profile you wish to copy.

3. Select the day profiles to be copied from the list **"Day profiles"**. After you have carried out the first two steps, the day profiles defined for the reference object will be shown in the list "Day profiles".

    Hint: You can only copy those day profiles which have a name different to the day profiles already defined in the Process start to which you are copying (as the day profiles are identified by their names).

Searching for Day Profiles

Please refer to the information provided for the performer calendar (see p. 364).

Calculating Average Number of Processes

The evaluation of the process calendar enables you to calculate the average number of processes during a specified period of time, based on the calendar defined.

When you click the "Query" button in the window "*<process start name>* - Process calendar" (see fig. 321) the window "Calculate average number of processes" (see fig. 334) is displayed.

Figure 334: Calculate average number of processes

Define the period for the calculation and then click on the "Query" button. The average number of processes will be displayed in the window **"Average number of processes"**.

4.1.9   Edit several attribute profile values simultaneously

When editing several attribute profiles, you will define the attribute values of these profiles in the ADONIS browser (see chap. 5., p. 44) in tabular form (see fig. 335). Attribute profiles will be displayed in rows and attributes in columns.

> **Hint:** The representation of attributes and the structure of ADONIS Notebooks depends on the customised definition in the application library.
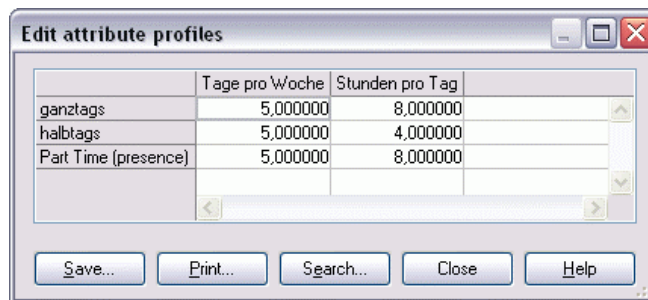

Figure 335: Edit multiple attribute profiles at a time

Define the values of attribute profiles by entering values in the appropriate cells.

> **ATTENTION:**   The name for an attribute profile must be unique inside of an attribute profile class.

Then click on the button "Close" to save the attribute values and close the window with the ADONIS browser. The updated list of attribute profiles will be shown.

Support for entering data

ADONIS provides dialogues for entering complex attribute values (or formulas) into the ADONIS Notebooks. The support for entering data in the ADONIS browser will be shown automatically if you double-click on the cell with the attribute profile value to be edited .

> **Hint:** The availability of the support for entering data depends on the definition in your application library.

The following supports for entering data can be available:

Edit record attributes (see chap. 4.1.8.1, p. 343)
     to editing attributes of the type "record".

Define colour (see chap. 4.1.8.2, p. 344)
     to select a colour (for the graphical display of an object).

Edit date attributes (see chap. 4.1.8.6, p. 346)
     to edit attributes of the type "Date" .

Edit Date /Date and time attributes (see chap. 4.1.8.6, p. 346)
     to edit attributes of the type "Date and time" .

Edit time attributes (see chap. 4.1.8.8, p. 347)
     to edit attributes of the type "Time".

Add references (see chap. 4.1.8.9, p. 348)
     to create model references in attributes of the type "Interref" .

Define the performer's calendar (see chap. 4.1.8.14, p. 359)
     to define the time of presence of performers.

Define the process calendar (see chap. 4.1.8.15, p. 365)
     to define the appearence profiles of processes.

4.1.10   Show Attribute Profile Values

When viewing an attribute profile, you can display the defined attribute profile values in an ADONIS Notebook (see fig. 336).

> **Hint:** The representation of attributes and the structure of ADONIS Notebooks depend on the customised definition in the application library.

Figure 336: Show attribute profile

> **Hint:** If it is not possible to edit attribute profile values. If necessary, contact your ADONIS consultant.

Click on the button "Close" to close ADONIS Notebook.

4.1.11   Show Several Attribute Profile Values simultaneously

When viewing several attribute profiles, you can display the defined attribute profile values in the ADONIS browser (see chap. 5., p. 44) in tabular form (see fig. 337). The attribute profiles will be displayed in rows and the attributes in columns.

Figure 337: Show several attribute profile values simultaneously

Click on the button "Close" to close the window with the ADONIS browser.

Support for entering data

ADONIS provides dialogues for entering complex attribute values (or formulas) into the ADONIS Notebooks. The support for entering data in the ADONIS browser will be shown automatically if you double-click on the cell with the attribute profile value to be edited .

> **Hint:** The availabilty of the support for entering data depends on the definition in your application library.

The following supports for entering data can be available:

Edit record attributes (see chap. 4.1.8.1, p. 343)
    to editing attributes of the type "record".

Define colour (see chap. 4.1.8.2, p. 344)
    to select a colour (for the graphical display of an object).

Edit date attributes (see chap. 4.1.8.6, p. 346)
    to edit attributes of the type "Date" .

Edit Date /Date and time attributes (see chap. 4.1.8.6, p. 346)
    to edit attributes of the type "Date and time" .

Edit time attributes (see chap. 4.1.8.8, p. 347)
    to edit attributes of the type "Time".

Add references (see chap. 4.1.8.9, p. 348)
    to create model references in attributes of the type "Interref" .

Define the performer's calendar (see chap. 4.1.8.14, p. 359)
    to define the time of presence of performers.

Define the process calendar (see chap. 4.1.8.15, p. 365)
    to define the appearence profiles of processes.

4.1.12   Copy Attribute Profile

If you want to copy an attribute profile to another attribute profile folder, select this attribute profile and then click on the button "Copy". As soon as you move the mouse pointer onto the list of the model hierarchy, it will be changed into   .

Then click on each attribute profile folder to which the previously selected attribute profile should be copied.

> **Hint:** Since attribute profile names in an application library must be unique, a generated number will be automatically appended to the name when creating the copy of an attribute profile.

4.1.13   Move Attribute Profile

Select the attribute profile you want to move and then click on the button "Move" .As soon as you move the mouse pointer onto the list of the model hierarchy, it will be changed into   .

Click on the attribute profile folder to which the previously selected attribute profile should be moved.

4.1.14 Delete Attribute Profile

Select the attribute profiles you want to delete and then click on the button "Delete".

Before the final deleting of attribute profiles, an appropriate security alert will appear.

If some references to the attribute profiles to be deleted should remain, the window "Deleting attribute profiles - Used attribute profiles" (see fig. 338) will be displayed indicating from which objects in which models these attribute profiles are referenced.
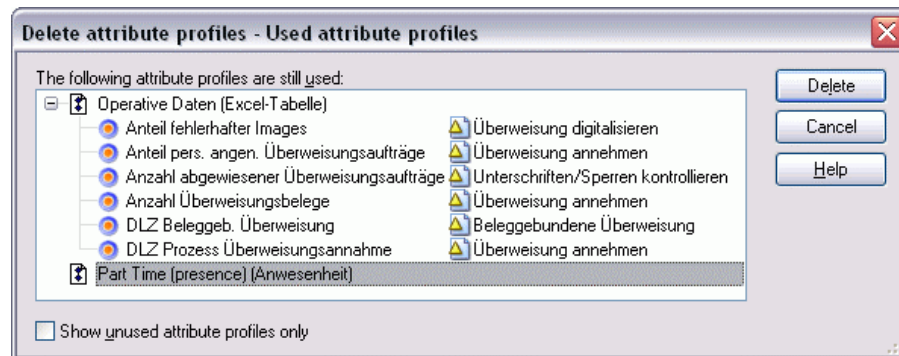

Figure 338: Delete attribute profiles - Used attribute profiles

Select the attribute profiles, which you want to delete despite their usage and click on the OK button.

Activate the option "Show not used attribute profiles only", to only show the attribute profiles which are currently not being referenced.

4.1.15 Show Used Attribute Profiles

The window "Attribute profile - Usage" (see fig. 339) shows all previously selected attribute profiles and the models and objects referencing these attribute profiles.
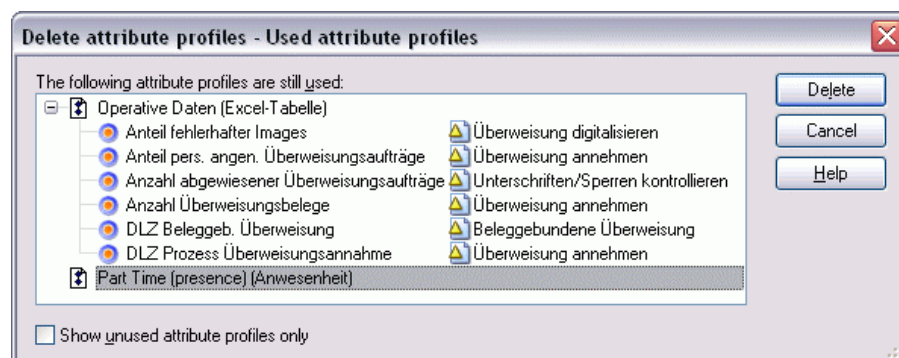
Figure 339: Attribute profile usage

Activate the option "Show not used attribute profiles only", to only show the attribute profiles which are currently not being referenced.

4.1.16 Carry out Queries on Attribute Profiles

In the window "Queries on attribute profiles" (see fig. 340) , you can carry out standardised queries (see chap. 4.1.16.1, p. 374) and user-defined queries (see chap. 4.1.16.2, p. 376).
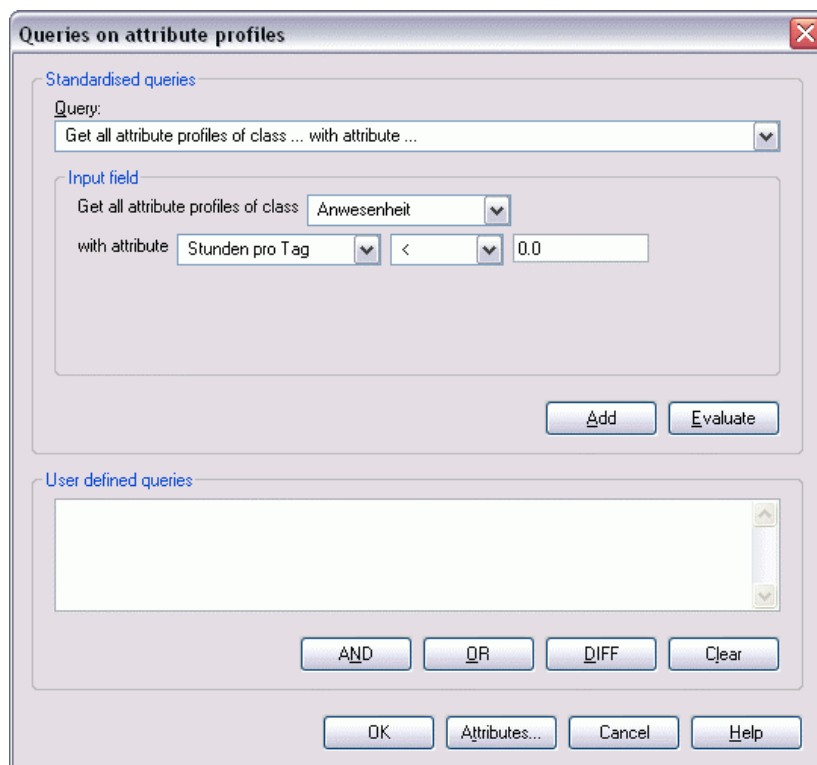
Figure 340: Standardised and user-defined queries on attribute profiles

4.1.16.1   Standardised Queries on Attribute Profiles

To carry out standardised queries, select a standardised query from the list "Query" (see fig. 341). An appropriate text with gaps appears in the input field of the selected query. Enter the required information (e.g. select name of the attribute profile/attribute) or select any value from the class/attribute/comparison operators lists.
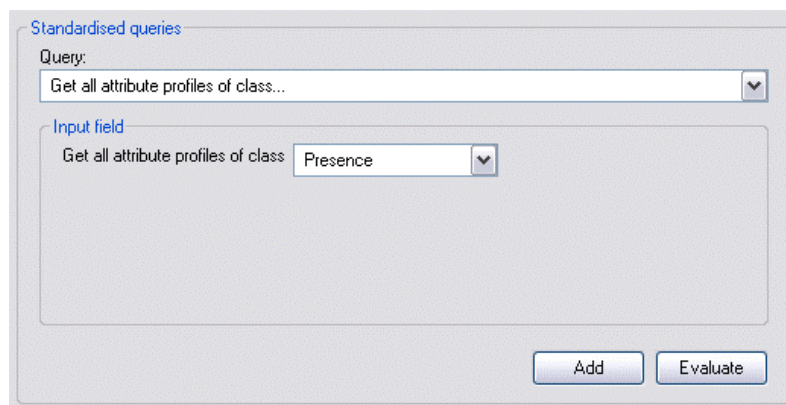

Figure 341: Standardised queries (part of the window "Queries on attribute profiles")

Standardized queries are explained below:

- **Find all attributes of the class** [*Selection*].

  This query displays all attribute profiles of the selected class.

- **Find all attributes of the class** [*Selection*] **with**

  attribute [*Selection*][*comparison operator*][*entry*].

This query displays all attribute profiles of the selected class with a specific attribute value.

Note: First select the class, then the attribute of the class and the comparison operator and enter the comparison value of the attribute.

● **Find all attributes of the class** [*Selection*] **with**

record attribute [*Selection*][*comparison operator*][*Number*]**.**

This query displays all attribute profiles of the selected class, which contain in their given record attribute the number of entries (rows) you have defined.

Note: These standardised queries will only be displayed if record attributes are defined in your application library.

● **Find all attributes of the class** [*Selection*] **with record attribute** [*Selection*] **and with**

column [*Selection*][*comparison operator*][*entry*]**.**

This query displays all attribute profiles of the selected class, which contain in the given table and column a specific attribute value.

Note: These standardised queries will only be displayed if record attributes are defined in your application library.

You can carry out the completed standardised queries by clicking on the **"Evaluate" button**. The query results (see chap. 4.1.16.3, p. 376) will be displayed in the ADONIS browser .

If you click on the **"Add" button** , the standardised query will be transformed into an AQL expression (see chap. 12., p. 593) and shown in the **"user defined query" field** (see fig. 343). From this query you can create a complex user-defined query (see chap. 4.1.16.2, p. 376) by entering further AQL expression and by adding further standardised queries.

By clicking on the **"Attribute" button** (see p. 375) you will define the attributes, which should be added to the name of the attribute profile found. If you select no attribute, the query results will be displayed **without** attribute.

> **Hint:** Standardised queries will be saved after the completion of the query. When recalling the menu item "Queries" the standardised query will be shown as a user-defined query (see chap. 4.1.16.2, p. 376) using AQL expression. This query remains saved until another standardised or user-defined query is entered or you close ADONIS.

> **Hint:** If an AQL expression is registered in the **"user-defined query" field**, you will carry out this user-defined query by clicking on the OK button, otherwise the current selected standardised query will be carried out.

Select Attributes for Result Display

All attributes of attribute profile classes will be shown in the window "Queries - Attribute column selection" (see fig. 342).
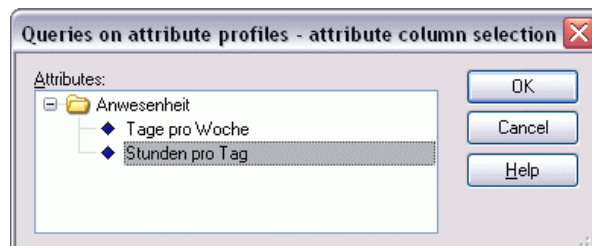

Figure 342: Queries on attribute profiles- Attribute settings

Attributes are always grouped under the attribute profile class in which they are defined. Select the attributes which you want to show in the display of query results in the ADONIS browser (see chap. 5., p. 44) and then click on the OK button.

> **Hint:** The attribute selection remains saved until you select other or additional attributes or you close ADONIS.

4.1.16.2   User-defined Queries on Attribute Profiles

Carry out the creation of user-defined queries within the **"user-defined query" field** (see fig. 343), while connecting standardised queries, defining queries in the query language AQL (see chap. 12., p. 593) or expanding standardised queries with AQL-expressions.
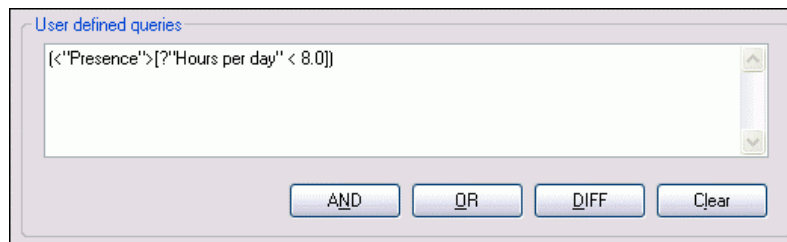
Figure 343: User defined queries (Part of the window "Query on attribute profiles")

You can use and assemble standardised queries (see chap. 4.1.16.1, p. 374) as a support for the definition of user-defined queries.

For this define a standardised query and click on the "Add" button. This will generate an AQL expression, which corresponds to your standardised query. This AQL-expression will be entered in the **"User defined query" field**.

By clicking on the **"AND"**, **"OR"** and **"DIFF"** buttons, you can combine or connect queries. You can also delete the content of the **"user-defined queries" field** by clicking on the **"Cancel" button**.

By clicking on the **"Attributes" button** (see p. 375) you will define the attributes, which should be displayed in addition to the name of the objects or connectors found.

> **Hint:** Standardised queries (see chap. 4.1.16.1, p. 374) will be saved after the completion of the queries. When recalling the "Query" menu item, the standardised query will be displayed as a user-defined query (AQL-expression). This query will remain saved, until you enter another standardised or user-defined query or you close ADONIS.
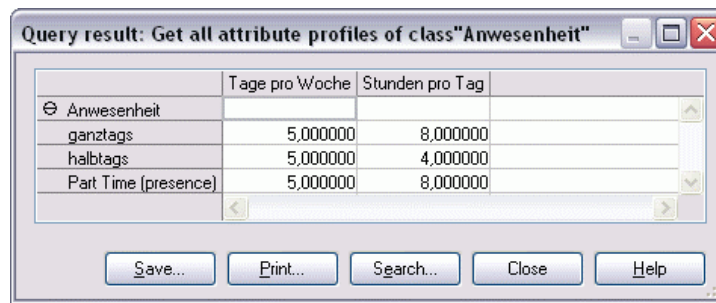
> **Hint:** If an AQL expression is registered in the **"user-definied query" field**, you will carry out this user-defined query by clicking on the OK button, otherwise the current specific standardised query will be carried out.

4.1.16.3   Query Results

The result of the queries will be displayed in the ADONIS browser (see chap. 5., p. 44) relative to their attribute profile class in the window "query result: <*Search criteria*>" (see fig. 344).

> **Hint:** Instead of<*Search criteria*>, the formulated query will be shown in the window "query". This is either the text of the standardised query or an AQL expression of the user-defined query.

The name of each attribute profile found will be shown in the first (grey) column, where the attribute profiles of a class are gathered under the class name of attribute profile. If additionally you have selected attributes for the result display (see p. 375), then the values of these attributes will be displayed, the name of these attributes will be shown in the first (grey) row (see fig. 344).

Figure 344: Queries on attribute profiles - Result display

> **Hint:** The display of record attributes (see p. 378) expands the representation in the ADONIS browser to columns (by integrating record attributes) and rows (by integrating the table content).

Select in the context menu (right mouse button) the menu item "Select attributes" to change the display of attributes (see p. 375).

You can save the query result to a file, print it out or display it as a diagram. Moreover, the search for specific contents is possible.

Furthermore, the attribute values of attribute profiles can be edited directly in the query result browser (see chap. 4.1.9, p. 370).

In addition it is possible to sort (see p. 377) the displayed objects and connectors according to specific attribute values using the context menu (right mouse button).

Sort Results

You can sort the displayed attribute profiles according to their attribute values using the menu item "Sort" in the context menu (right mouse button).

Once you have selected the menu item "Sort", the window "Sort - Attribute selection" (see fig. 345) will be shown.
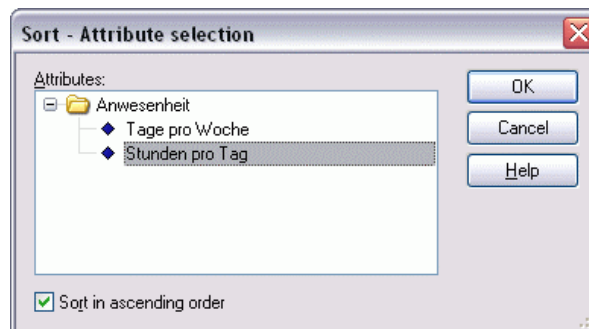
Figure 345: Sort - Attribute selection

Select from the lists the attribute according to which the query result will be sorted. The **"Sort in ascending order" option** enables the choice between ascending and descending order.

> **Hint:** The sort is carried out attribute profile class- related, i.e. the query result will be sorted using the selected attribute within the attribute profile class.

Display Record Attributes

For the result display of the attributes (see fig. 344), the first representation of record attributes is carried out in the ADONIS browser by the symbol $\oplus$. This means, that it is a short display of tables and that the record attributes can be displayed by a click.

Once you have clicked the cell with the symbol $\oplus$ the display of the table will be expanded: Either the column will be expanded to all defined record attributes or the table rows will be integrated in the display as additional rows.

> **Hint:** In the expanded display of a table, all record attributes of an object will be outlined in blue.

The symbol $\ominus$ indicates, that all record attributes are displayed. Clicking on it brings up a compact view of the displayed table.

To **shrink or expand all record attributes**, select in the context menu of the ADONIS browser (right mouse button) the menu item "Shrink all record attributes" or "Expand all record attributes".

5.   Component Management

With the help of the Component Management (see fig. 346) you can extend the user-specific configu-
ration of **ADONIS Standard** (see chap. 5.1, p. 20) or **ADONIS light** (see chap. 5.2, p. 21) as required.
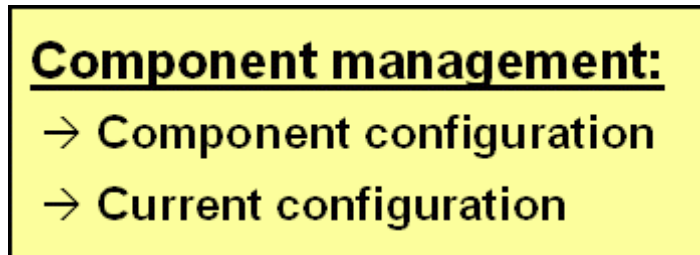

Figure 346: Functionality in the Component Management

The ADONIS light configuration of the ADONIS Business Process Management Toolkit (see chap. 4.,
p. 18), consists of the components "Information Acquisition", "Modelling", "Analysis" and "Im-
port/Export" (without "Transformation"). The standard configuration also includes "Simulation" and
"Evaluation". Using the Administration component, the ADONIS administrator may provide the
ADONIS users with access to additional components (see chap. 4., p. 18) which may have been pur-
chased since ADONIS was originally installed.

The current configuration (see chap. 5.2, p. 382) shows you which components and which elements of
each component are available at the moment.

> **ATTENTION:**   The component configuration and the display of the current configuration
> always refers to the **current** ADONIS database (i.e. the ADONIS database
> which was specified by the administrator when logging onto the Administra-
> tion Toolkit.

The name of the current ADONIS database is displayed in the title bars of the windows within the
Component Management.

To use the services provided by the Component management component of ADONIS, please proceed
according to the following steps:

1. Start the ADONIS Administration Toolkit. The window ADONIS Administration Toolkit
   (see chap. 3., p. 16) will appear.

2. Activate "Component management" by clicking on the corresponding smart-icon 🧩 in the quick-
   access panel.

   Alternatively, you can activate this component by opening the component panel's popup menu.
   Click with the right mouse button on the component panel where the smart-icons for the compo-
   nents can be seen. Then select the menu option "Component Management".

   You can also use the function key <F9> to open the popup menu and the key <m> to then select
   component management.

   Once the Component Management is active, the quick-access panel includes the smart icons for
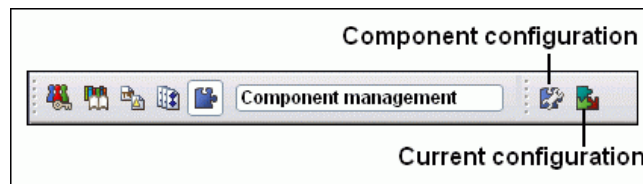   this component (see fig. 346).

Figure 347: Icon bar of the Component Management

5.1 Component Configuration

Component configuration allows you to specify the license specific activation of the components of the ADONIS Business Process Management Toolkit (see chap. 4., p. 18).

Basically, the following components are available in ADONIS:

- **Acquisition**, including the part:
  - Acquisition tables
- **Modelling**, including the part:
  - Attribute profile
- **Analysis**, including the parts:
  - Queries
  - Predefined queries
  - Relation tables
  - Calculation
- **Simulation**, including the parts:
  - Path analysis
  - Capacity analysis
  - Workload analysis (steady state)
  - Workload analysis (fixed time period)
  - Agents
- **Evaluation**, including the parts:
  - Flowmark audit trail evaluation
  - Result evaluation
  - Evaluation queries
- **Import/Export**, including the parts:
  - ADL-import/export
  - Documentation
  - FDL generation
- **Additional components**, including the parts:
  - *Case/4/0-coupling*
  - *ObjectiF-coupling*
  - *Process costs analysis*

- *Dynamic evaluation module*

● **Administration components**, including module:

- Test version

Hint: The modules which are <u>underlined</u> in the list, constitute the package of ADONIS light.

Hint: The modules which are displayed in *italics* in the list above are <u>not</u> included in the standard package of ADONIS.

ATTENTION: To configure new components, a new license number has to be entered. This must be purchased. Your ADONIS consultant will provide you with the new license number.

If you require any additional components, please contact your ADONIS consultant specifying which components and parts are required and also specifying your current customer number. You can find your customer number either on the information sheet which was originally delivered with the installation kit or in the window "Current Configuration" (see chap. 5.2, p. 382)

If you wish to update your current configuration, select the menu option "Component configuration" from the "Components" menu or click on the corresponding smart icon in the quick-access bar.

The window "<*Database Name*>: Component configuration" (see fig. 348) showing your customer number will be displayed. (Instead of <*Database Name*> the name of the current ADONIS database will be shown.)
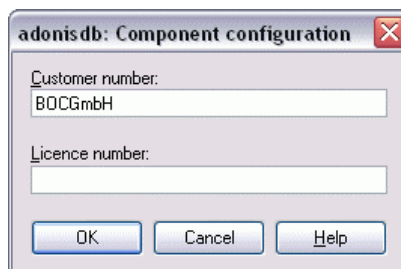

Figure 348: Enter new licence number

The customer number can be seen in the field "Customer number" and cannot be changed.

Hint: Ensure that this customer number has been given to to your ADONIS consultant so that your new license number will match it.

Enter the new license number into the field "License number" and click on the "OK" button. Clicking on the "Cancel" button closes the window and cancels the configuration update.

Once the license number has been entered correctly, an information window appears to tell you that the new configuration has been saved in the ADONIS database. Close the information window by clicking on the "OK" button or by pressing the "Enter" key.

From now on, this new configuration of the ADONIS Business Process Management Toolkit is available to all ADONIS users of the current ADONIS database.

Hint: Access to the components can be restricted for any ADONIS user through the User Administration component (User Settings (see chap. 1.4.6, p. 100)).

5.2 Current Configuration

The current configuration shows you which components or component elements are currently available to the users of the ADONIS Business Process Management Toolkit (see chap. 4., p. 18).

To display the current configuration, activate the Component Management and select the option "Current Configuration" from the "Components" menu. Alternatively, you can click on the smart icon 🖼 in the quick-access panel.

Once you have clicked on the menu item or on the smart icon, the window "Current configuration" (see fig. 349) will appear.
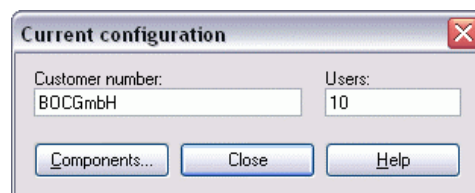
Figure 349: Current configuration - User file

In addition to the customer number and the maximum number of concurrent ADONIS users, the current configuration (see chap. 5.2.1, p. 382) of the ADONIS Business Process Management Toolkit can be seen by clicking on the "Components" button.

5.2.1 Component Availability

The current configuration of the ADONIS Business Process Management Toolkit is displayed in an ADONIS notebook, with each chapter corresponding to a component (see fig. 350).
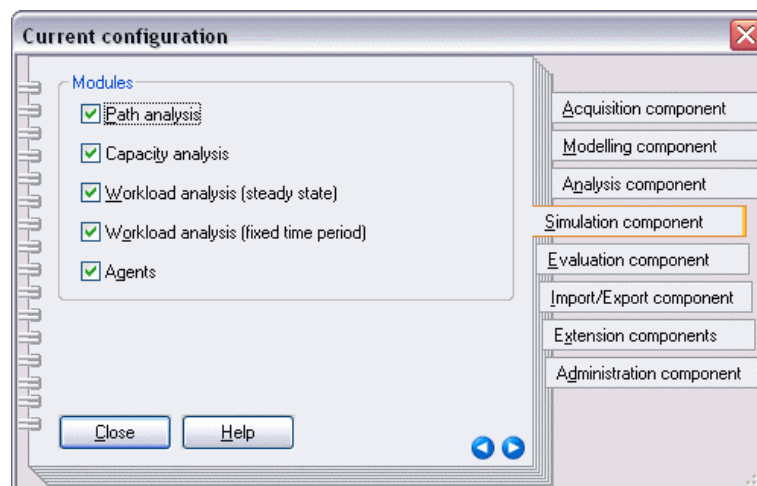

Figure 350: Current configuration - Component availability

The elements of the components (see chap. 5.2.2, p. 383) of the ADONISBusiness Process Management Toolkit which are available for the user are characterised by a tick ☑.

5.2.2   Component Access Rights

Authorisations can be conferred for the following functionality modules of ADONIS. The access to each module must first be basically activated via the licence number. If this is the case, the access for user groups or users can be further restricted.

- **Acquisition**:

  - Acquisition tables : This right enables the start of the Acquisition Component HOMER.

- **Modelling**:

  - Change access status: This right enables the setting of the model attribute "Access status". This is only necessary in specific configurations of ADONIS. The model attribute "Access status" is not used in the standard configuration of ADONIS - the right "Change access status" is not relevant in this case. (Note: There is no need to explicitly activate the right "Change access status".)

  - Change model description: This right enables the setting of the model attributes "Key words", "Description" and "Comment".

  - Change user attributes of a model: This right enables the setting of the model attributes "Model type", "Status", "Reviewed on" and "Reviewed by".

  - Attribute profiles: This right enables to create new attribute profiles and attribute profile groups and to edit existing attribute profiles and attribute profile groups.

- **Analysis**:

  - Queries: This right provides access to carry out standardised AQL queries.

  - Predefined queries: This right enables the carrying out of configuration-specific, predefined AQL queries.

  - Relation tables: This right enables the creation of relation tables. Note that relation tables can be created only in specific configurations of ADONIS. In all other cases the right "Relation tables" is not relevant.

  - Analytic evaluation: This right enables access to carry out the analytical evaluation.

- **Simulation**:

  - Path analysis: This right enables access to carry out the path analysis.

  - Capacity analysis: This right enables access to carry out the capacity analysis.

  - Workload analysis (steady state): This right enables access to carry out the workload analysis (steady state).

  - Workload analysis (fixed time period): This right enables access to carry out the workload analysis (fixed time period).

  - Agents: This right enables the creation of new simulation agents and to change the configuration of existing agents.

- **Evaluation**:

  - FlowMark Audit Trail evaluation: This right enables the evaluation of FlowMark Audit Trails.

  - Comparison of results: This right enables access to carry out comparisons of results on an APF files basis.

  - Evaluation queries: This right enables access to carry out evaluation queries.

- **Import/Export**:

  - ADL import/export: This right enables access to carry out ADL imports and ADL exports.

- Documentation: This right enables access to carry out the documentation generation (HTML and/or RTF).

- FDL generation: This right enables access to carry out FDL generation.

● **Additional components**:

- case/4/0-coupling: This right enables the access to the ADONIS-case/4/0 interface.

- objectiF-Anbindung: This right enables the access to the ADONIS-objectIF interface.

- Process cost analysis: This right enables access to carry out the process cost analysis.

- Dynamic evaluation module: This right enables the execution of "dynamical evaluation modules", for instances of the ADONIS human resources administration component.

● **Administration components**, with the module:

- Test version: This option activates or deactivates the test version status of ADONIS.

- No new databases (only MSDE): If this option is activated, only one ADONIS database can be created and used when using the database system MSDE. This option is normally activated in the stand-alone use of ADONIS with MSDE Runtime.
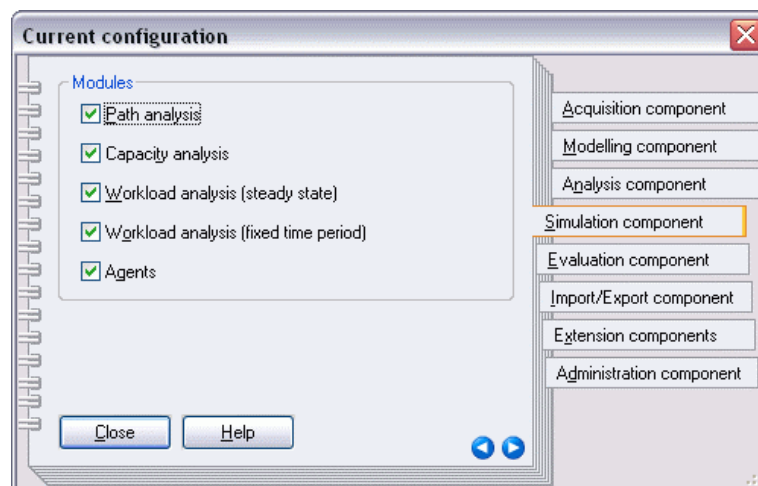

Figure 351: Component access

6.   Error Messages

You will find a detailed description of each error message (including cause and action) in:

- the context menu for each error message by clicking on the help button in the window of the error message,

- in the online documentation (menu item "ADONIS help" in the menu "Help") in the chapter "Error messages",

- in the PDF documentation in the chapter "error messages" (PDF files are to be found on the ADONIS installation CD in the directory \BOOKS).

This chapter lists the error messages that may occur in the ADONIS Administration Toolkit. An error message may belong to one of the following groups:

aadma	**aalkwins**	**aanadlg**	**aanaud**
aap	**aapedit**	**aapqury**	**aapview**
abmpsup	**abrsimpl**	**acalui**	**acard**
acnumchk	**acoexhan**	**acoexman**	**acoexpar**
aconfui	**acregchk**	**adbacc**	**adbadm**
adbsess	**adistrib**	**aeagents**	**aeasubag**
aexpappl	**aexport**	**afile**	**afilemgt**
agdt	**agogo**	**agolay**	**ahaconv**
ahagrob	**ahamot**	**ahanote**	**aicondef**
aimport	**airdom**	**aleo**	**alib2sgm**
alibchk	**alibload**	**alibmgt**	**alibren**
alogin	**amigrat**	**amodmgt**	**anamegen**
anbbrow	**anotebk**	**aoutgen**	**apercalc**
aperout	**aperpar**	**aperress**	**apwchgui**
aqueryed	**aregex**	**arelrdef**	**areposui**
arightmg	**ascope**	**ascript**	**asetsel**
asgmlexp	**asimmap**	**asimtext**	**asrchtlb**
astdcfg	**asysbox**	**ausmgt**	**avedbox**
averdate	**awrjpeg**	**svxwins**	

An error message consists of an error code (in square brackets - []) that is shown at the beginning of the error message and of the message's text (below the error code). The error code is a combination of letters and numbers. The letters represent the error group, the numbers serve to number the messages within an error group. The error messages are listed in alphabetical order according to groups and in ascending order inside each group. Should you have problems finding an error message, please use the online help or check the index.

> **Hint:** The list of errors is ordered alphabetically according to error groups and the groups are numbered in ascending order. The numbering need not necessarily be continuous.

You can close the window containing the error message by clicking on the OK button or by pressing the "Enter" key. Then execute the actions described under "Action".

[aadma-02]

Message: The initialisation of ADONIS failed. Login cancelled.

Cause: Possibly, the connection with the ADONIS database server was aborted.

Action: Repeat the login process. Should the error re-occur, please contact your database or system administrator.

[aadma-04]

Message: "*<"\""%name\"">*" (*<(%type)>*) could not be deleted, because it is currently locked by user *< %user!>* !

Cause: The model listed is currently being edited by the ADONIS user specified and thus cannot be deleted.

Action: The user specified above should stop the editing of the model and close this model so that it can then be deleted.

    (You can also send an appropriate messsage to the user via ADONIS-Mail (Menu "Extras", menu item "Messages").)

[aadma-05]

Message   "*<"\""%name\"">*" (*<(%type)>*) has the access state \"read access\" and cannot be deleted!

Cause: The model listed is loaded as read-only and thus can neither be changed or deleted.

Action: Close the model specified and then attempt to delete it again.

[aadma-06]

Message: An error occurred when deleting in the database!

Cause: It is possible that your connection to the database server has been aborted or the database server is overloaded.

Action: Repeat the deletion. Should the error re-occur, either try again later or else contact your database or system administrator.

[aadma-07]

Message: The library "*<library name>*" could not be loaded! Quit ADONIS.

Cause: The model and model group list belonging to the library could not be loaded .

Action: Quit ADONIS and the restart ADONIS. If the error occurs again, contact your ADONIS administrator.

[aadma-08]

Message: The configuration file "<*file name*>" could not be written.

There are probably no access rights for the write access to the file.

(The configuration change has only been carried out in the database and thus is not complete.)

Contact your system administrator and then carry out the configuration change again.

Cause: When trying to change the configuration, the specified configuration file could not be opened.

Action: Verifiy whether you possess enough write access rights in the target directory and if enough hard disk memory is available. If the file already exists, check whether the file is write protected or already opened.

[aadma-09]

Message: The file "<*file name*>" specified after the parameter -e cannot be opened!

Cause: The file name for the parameter -e to execute an AdoScript is not valid.

Action: Please check the path and name of the file and enter a valid file name.

[aadma-10]

Message: ADONIS can't be used with the desired language <*language*> since

the licence number is not configured for this language.

The login process will be aborted.

Cause: The access rights defined in the licence number do not allow a login in the selected language.

Action: Contact your ADONIS administrator.

[aalkwins-01]

Message: Invalid filename.

Cause: An invalid filename, the name of a write-protected file or a directory that does not exist has been specified for the import file.

Action: Enter a valid name of a file that is not wirte-protected in an existing directory.

[aalkwins-02]

Message: Enter a filename.

Cause: You have not specified a filename for the ADL file to be imported.

Action: Enter the filename of the ADL file.

[aalkwins-08]

Message: The models contained in the application model "*<model name>*" have been deleted. Therefore this application model is no longer valid.

Cause: The application model you have specified for export no longer contains any models.

Action: Specify a different application model which contains the models you require.

[aalkwins-10]

Message: Username must be at least three characters!

Cause: When renaming a username during UDL import, a name of less than three characters was specified.

Action: Please enter a new username which is at least three characters long.

[aalkwins-11]

Message: An unknown model type has been used!

Cause: For the export using AdoScript an unknown model type was provided. The export of models of this type is not possible.

Action: Use the name of an existing model type in the AdoScript.

[aanadlg-04]

Message: The models to be analysed must be defined on the same application library.

Cause: The models selected for analysis are based on different application libraries.

Action: Select a number of models to be analysed which are based on the same application library.

[aanadlg-05]

Message: An error has occurred during the storage!

Cause: An error has occurred during the storage of the model information of the models analysed.

Action: The data medium is possibly write protected (e.g. floppy disk).

[aanadlg-06]

Message: The entered value "*<value>*" is not valid!

    <Format> is expected.

Cause: While creating a standardised query, you have entered a value which is not compatible with the selected attribute.

Action: Enter a value compatible with the entry type.

[aanaud-01]

Message: Error in application library.

Cause: The query definitions in your application library contain errors.

Action: Contact your ADONIS administrator or your ADONIS consultant.

[aanaud-03]

Message: Error in library attribute "*<attribute name>*":

           <Number>. query:

           Unknown model type ("*<model type name>*")!

Cause: An unknown model type has been used in the query in the above-mentioned library at-
           tribute.

Action: Contact your ADONIS administrator.

[aanaud-04]

Message: Error in library attribute "*<attribute name>*":

           <Number>. query:

           Invalid context!

Cause: A key word has been used in a wrong context in the query in the above-mentioned li-
           brary attribute.

Action: Contact your ADONIS administrator.

[aanaud-05]

Message: Error in library attribute "*<attribute name>*":

           <Number>. query:

           Unknown key word ("*<Name>*")!

Cause: An unknown key word has been used in the query in the above-mentioned library at-
           tribute.

Action: Contact your ADONIS administrator.

[aanaud-06]

Message: Error in library attribute "*<attribute name>*":

           <Number>. query:

           Invalid menu item!

Cause: The menu item is missing in the query in the above-mentioned library attribute.

Action: Contact your ADONIS administrator.

[aanaud-07]

Message: Error in library attribute "*<attribute name>*":

             <Number>. query:

             Invalid query!

Cause: The query text is missing in the query in the above-mentioned library attribute .

Action: Contact your ADONIS administrator.

[aanaud-08]

Message: Error in library attribute "*<attribute name>*":

             <Number>. query:

             Invalid AQL expression!

Cause: A syntactical invalid AQL expression is used in the query in the above-mentioned library attribute.

Action: Contact your ADONIS administrator.

[aanaud-09]

Message: Error in library attribute "*<attribute name>*":

             <Number>. query:

             Unknown class ("*<class name>*") in the *<context>*!

Cause: An unknown class is used in the query in the above-mentioned library attribute. The context can be the input fields or the result attributes.

Action: Contact your ADONIS administrator.

[aanaud-10]

Message: Error in library attribute "*<attribute name>*":

             <Number>. query:

             Unknown attribute ("*<attribute name>*") in the *<context>*!

Cause: An unknown attribute is used in the query in the above-mentioned library attribute. The context can be the input fields or the result attributes.

Action: Contact your ADONIS administrator.

[aanaud-11]

Message: Error in library attribute "*<attribute name>*":

             <Number>. query:

Key word missing("<*Name*>")!

Cause:	The named key word is missing in the query in the above-mentioned library attribute.
Action:	Contact your ADONIS administrator.

[aanaud-12]

Message:	Error in library attribute "<*attribute name*>":
	<*Number*>. query:
	Context missing!
Cause:	A required context is missing in the query in the above-mentioned library attribute.
Action:	Contact your ADONIS administrator.

[aanaud-13]

Message:	Error in library attribute "<*attribute name*>":
	<*Number*>. query:
	Input field empty!
Cause:	The input elements are missing in the query in the above-mentioned library attribute.
Action:	Contact your ADONIS administrator.

[aanaud-14]

Message:	Error in library attribute "<*attribute name*>":
	<*Number*>. query:
	AQL expressions empty!
Cause:	The AQL expressions are missing in the query in the above-mentioned library attribute.
Action:	Contact your ADONIS administrator.

[aanaud-15]

Message:	The entered value "<*Value*>" is invalid!
	Expected is <*Format*>.
Cause:	While carrying out a predefined query, you have entered a value, which does not match to the defined input type.
Action:	Enter a value which suits the input type.

[aanaud-16]

Message:	Error in library attribute "<*attribute name*>":

<*Number*>. query:

Result attribute empty!

Cause: A plan without result attributes has been defined in the query in the above-mentioned library attribute.

Action: Contact your ADONIS administrator.

[aanaud-17]

Message: Error in library attribute "<*attribute name*>":

The length of the attribute value exceeds the maximum allowed length of 32.000 characters!

Cause: The definition for the mentioned query attribute exceeds 32.000 characters and therefore can not be saved.

Action: Reduce the attribute value by deleting not used parts of the text.

[aap-01]

Message: The attribute possesses no facet of the type "<*Type name*>".

Cause: The attribute possesses no facet with the names "AttributeProfileRefDomain" and so cannot be used.

Action: Contact your ADONIS administrator.

[aap-02]

Message: The expression is faulty:

"<*LEO error message*>"

Cause: The value of the facet "AttributeProfileRefDomain" is faulty. The LEO expression entered contains a syntactical error. Note the appropriate [aleo] error message.

Action: Rectify the value of the facet "AttributeProfileRefDomain" and take into account the specified [aleo] error message or contact your ADONIS administrator.

[aap-03]

Message: The LEO expression is empty.

Cause: The facet "AttributeProfileRefDomain" is empty.

Action: Rectify the value of the facet "AttributeProfileRefDomain" or contact your ADONIS administrator.

[aap-04]

Message: The key word "APREF" is missing.

Cause: The LEO expression in the facet "AttributeProfileRefDomain" does not contain the required key word "APREF".

Action: Correct the value of the facet "AttributeProfileRefDomain" by adding the key word "APREF" or contact your ADONIS administrator.

[aap-05]

Message: The parameter "c" of the key word "APREF" is missing.

Cause: The parameter "c" of the key word "APREF" is required but has not been found.

Action: Rectify the value of the facet "AttributeProfileRefDomain" by adding the parameter "c" to the key word "APREF" or contact your ADONIS administrator.

[aap-06]

Message: The parameter "c" of the key word "APREF" contains an unknown class name.

Cause: The class entered in the parameter "c" does not exist.

Action: Rectify the value of the facet "AttributeProfileRefDomain" by adding a valid class name or contact your ADONIS administrator.

[aap-07]

Message: The specified class in the parameter "c" of the key word "APREF" is not an attribute profile class.

Cause: The specified class in the parameter "c" is not the name of an attribute profile class but the name of another ADONIS class.

Action: Rectify the value of the facet "AttributeProfileRefDomain" by inserting a valid attribute profile class name or contact your ADONIS administrator.

[aap-08]

Message: The specified class in the parameter "c" of the key word "APREF" is not instanceable.

Cause: The class specified in the parameter "c" may be an attribute profile class, but it has been defined as "abstract" and so cannot be instanced.

Action: Rectify the value of the facet "AttributeProfileRefDomain" by inserting a valid attribute profile class name or set the referenced class on "not abstract" or contact your ADONIS administrator.

[aap-09]

Message: The LEO expression contains too many key words.

Cause: The facet "AttributeProfileRefDomain" can only reference one attribute profile class.

Action: Rectify the value of the facet "AttributeProfileRefDomain" by referencing only one attribute profile class or contact your ADONIS administrator.

[aap-10]

Message: The LEO expression can contain a maximum of one reference.

Cause: An attribute value refers to more than one attribute profile.

Action: Change the attribute value by referencing only one attribute profile or contact your ADONIS administrator.

[aap-11]

Message: The key word "REF" is missing.

Cause: The attribute value is faulty because the key word "REF" is missing.

Action: Change the attribute value by inserting the key word "REF" or contact your ADONIS administrator.

[aap-12]

Message: The parameter "i" of the key word "REF" is missing.

Cause: The attribute value is faulty because the parameter "i" of the key word "APREF" is not available.

Action: Change the attribute value by adding the parameter "i" to the key word "REF" or contact your ADONIS administrator.

[aap-14]

Message: The specified instance in the parameter "i" of the key word "REF" is invalid.

Cause: The referenced attribute profile has not been found.

Action: Change the attribute value so that a valid attribute profile is shown or contact your ADONIS administrator.

[aap-15]

Message: The specified instance in the parameter "i" of the key word "REF" does not have the suitable class.

Cause: The referenced attribute profile is not from the class specified in the facet "Attribute-ProfileRefDomain".

Action: Change the attribute value so that a valid attribute profile is shown or contact your ADONIS administrator.

[aapedit-01]

Message: The entry "*<entry text>*" is not authorised for the attribute "*<attribute name>*"!

<Additional error text>

Would you like to further edit the attribute?

Cause: During the editing of attribute profiles, an invalid attribute value has been entered.

Action: Enter a valid value for this attribute.

[aapedit-02]

Message: The name "*<object name>*" is already assigned to another object of the class "*<Class name>*" !

      <Additional error text>

      Would you like to further edit the attribute?

Cause: When changing the name of an object in the Attribute Profile Management, you have selected a name which is already given for this class .

Action: Select another name for the object.

[aapedit-03]

Message: Not all values could be inserted!

Cause: When pasting in the ADONIS browser, an error has occurred. This cancelled the pasting.

Action: Check if the area to be pasted is not too large and if the attributes go together. Then verify if it is allowed to paste in the area you have selected.

[aapqury-01]

Message: The AQL expression is syntactically not correct!

Cause: The AQL expression you have entered does not correspond with the AQL syntax defined.

Action: Enter a syntactically correct AQL expression.

[aapqury-02]

Message: "*<Attribute profile name>*" is already being edited by another ADONIS user and thus can not be changed!

Cause: The specified attribute profile cannot be changed as it is already being edited by another ADONIS user.

Action: Carry out the changes later.

[aapqury-03]

Message: Error during the copy: Not all values could be inserted!

Cause: During the insertion in the ADONIS browser, an error has occurred. The pasting has been cancelled.

Action: Check if the area to be pasted is not too large and if the attribute types go together. Then check if it is possible to paste in the area you have selected.

[aapqury-04]

Message: Error during the copy: No value could be inserted!

Cause: During the insertion in the ADONIS browser, none of the values to be inserted could be assigned.

Action: Check if the area to be pasted is not to large and if the attribute types go together. Then check if it possible to paste in the area you have selected.

[aapview-01]

Message: Error when following an attribute profile reference:

"<*Model name*>" (<*model type*>)

could not be opened.

The referenced model has meanwhile possibly been deleted.

Cause: The original model of the attribute profile reference cannot be opened, because it has been meanwhile deleted by another ADONIS user or your access right to this model has been withdrawn by the ADONIS administrator.

Action: Speak with other ADONIS users or with your ADONIS administrator.

[abmpsup-01]

Message: File "<*filename*>" cannot be opened!

Cause: One of the specified files cannot be opened with write access. Possibly an incorrect path has been entered or there may be too little space available in the directory or disk on which the file should be stored.

Action: Please check that the file and path were specified correctly and check the space available on the data medium on which the file should be stored.

[abmpsup-02]

Message: No file name specified!

Cause: You wish to save the generated graphic in a file, but have not specified a filename.

Action: Please enter a filename in which the graphic should be stored.

[abrsimpl-06]

Message: The column width must be of a number greater than 0 and smaller than 200!

Cause: You have entered no number or a number outside the valid area (between 0 and 200, excluding limits).

Action: Enter a valid column width.

[acalui-01]

Message: The day profile *<day profile name>* cannot be deleted because it is still being used. Please click on the search button to identify the assigned days.

Cause: You are attempting to delete a day profile which is still assigned to a number of days in the calendar. A day profile can only be deleted when it is no longer assigned to any days.

Action: If you really want to delete the selected day profile, then you must first ensure that it is no longer assigned to any days in the calendar. Using the "Search" button, you can identify the days to which this day profile is still assigned. Using the "Assign" button you can assign a day profile to a number of different days. Alternatively, by double-clicking on a day directly within the calendar, you can change the assigned day profile.

[acalui-02]

Message: Please enter a name for the new day profile.

Cause: You have not specified a name for the day profile you are trying to create. As day profiles are identified by their names, it is essential to enter a unique name.

Action: Enter a valid name for the day profile.

[acalui-03]

Message: The day profile name <day profile name> is not allowed. Either a day profile with this name already exists or the name specified contains restricted characters. Please use a different name.

Cause: An invalid name has been entered for the day profile. Either a day profile with this name already exists or the name specified contains the character "@".

Action: Enter a valid name for the new day profile which does not contain the character "@".

[acalui-04]

Message: The maximum number of day profiles has been reached. Before defining a new day profile, please delete at least one.

Cause: There is a maximum permitted number of day profile (approx. 50). This number has been reached.

Action: Delete at least one of the existing day profiles in order to define a new one.

[acalui-05]

Message: Please select a day profile.

Cause: You have clicked on a button (Change, Delete, Assign or Search) which requires a day profile to be specified before the relevant action can be carried out.

Action: Select a day profile from the list and click on the button again.

[acalui-06]

Message: The input in the start field is not time.

Cause: You have entered a value into the Start field which is not in the correct time format. Note that the required format is "hh:mm:ss", where "hh" is a number between "00" and "23", and both "mm" and "ss" require a number between "00" and "59". A valid value is for example "08:00:00".

Action: Enter a time as described in "Cause" above.

[acalui-07]

Message: The input in the End field is not time.

Cause: You have entered a value into the End field which is not in the correct time format. Note that the required format is "hh:mm:ss", where "hh" is a number between "00" and "23", and both "mm" and "ss" require a number between "00" and "59". A valid value is for example "17:00:00".

Action: Enter a time as described in "Cause" above.

[acalui-08]

Message: The start time cannot be later than the end time.

Cause: When defining time intervals, the start time must be earlier than the end time. Please note that the time intervals within a day must be between "00:00:00" and "23:59:59".

Action: Enter values in the Start and End fields which make up a valid time interval.

[acalui-09]

Message: There are intersections with intervals that are already defined in the day profile.

Cause: You have attempted to define a new time interval which overlaps with existing intervals. The time intervals within a day profile must be distinct - i.e. they cannot overlap with any other time intervals.

Action: Enter a valid time interval which doesn't overlap with any other time intervals. If necessary, delete or change existing time intervals.

[acalui-10]

Message: Please select an interval.

Cause: You have clicked on a button (Change or Delete) which requires a time interval on which to work.

Action: Select the time interval on which you want to carry out the action, and then click on the relevant button again.

[acalui-13]

Message: Please consider that the start date must be before the end date.

Cause: You have entered start and end dates which don't define a time interval. Please note that the time interval must lie within a single year (January - December).

Action: Enter a start and end date which define a valid time interval.

[acalui-14]

Message: The days set is empty. Please activate at least one input type.

Cause: You have not selected an input type ("Day" or "Time interval"). This means that no dates can be selected to which a day profile can be assigned.

Action: Activate at least one input type (by clicking) and then define the days to which the day profile should be assigned.

[acalui-15]

Message: The calendar is too large. Therefore it cannot be stored in the database. Please delete some day profiles.

Cause: There is an internal limit defined for the complexity of the calendar. When a very high number of day profiles are defined (>50) the calendar can become so complex that it can no longer be saved.

Action: Delete any unnecessary day profiles - i.e. day profiles which are not assigned to any days.

[acalui-16]

Message: The distribution is not correct.

Cause: You have entered a value into the input field which does not meet the conventions of the distribution.

Action: Once you select your required distribution type, the syntax is shown in the 'Distribution' field. When using uniform distribution, please be careful that the minimum and maximum bounds are in the correct time format.

[acalui-17]

Message: The maximum number of day profiles has been defined. Before copying a day profile, please delete at least one.

Cause: A limit exists for the maximum number of day profiles which can be defined (approx. 50).

Action: Delete at least one day profile.

[acalui-18]

Message: Please specify the reference object and select at least one day profile.

Cause: In order to copy a day profile into the current calendar, you must first specify the model from which you want to copy, the reference object which contains the day profile you want to copy and finally the actual day profile itself.

Action: Select the reference object and the day profile.

[acalui-19]

Message: A day profile named*<day profile name>* already exists. Therefore it cannot be copied.

Cause: Day profiles must have a unique name within a calendar. Therefore if a day profile with the same name already exists in the calendar to which you are attempting to copy, the system will not allow the new profile to be copied.

Action: If you are sure that you want to copy the new day profile into your calendar, you must first delete the existing day profile with the same name.

[acalui-20]

Message: The calendar has the wrong format. Therefore the day profiles cannot be copied.

Cause: If the calendar of the selected reference object has an invalid format, then no day profiles can be copied.

Action: Select a different reference object. In order to correct the invalid calendar of the reference object, you should open the ADONIS notebook of that object - the calendar will then automatically be corrected to contain standard values.

[acard-02]

Message: "*<Object>*": A maximum of *<number>* connectors is allowed (*<direction>*).

Should the verification of the cardinalities be continued?

Cause: The verification of the cardinalities of the model has determined that the maximum number of allowed connectors in the specified object is too high .

The possibly shown direction ("incoming", "outgoing" or "all together") indicates whether the maximum number of incoming or outgoing connectors has been reached or if the number of connectors in general is too high.

Action: According to the defined cardinality restrictions, only a certain number of connectors is allowed. To adhere to these restrictions, you have to delete connectors which start from or end in the specified object.

[acard-03]

Message: "*<source>*": A maximum of <number> connectors of the type "*<relation class>*" is allowed (<direction>).

Should the verification of the cardinalities be continued?

Cause: The verification of the cardinalities of the model has determined that the maximum number of allowed connectors of the specified relation type in the specified source is too high.

The *source* is about a concrete object, an object of the class displayed a relation between a concrete object and objects of the specified class or about the relation between objects of the class shown and a concrete object.

The possibly shown direction ("incoming", "outgoing" or "all together") indicates whether the maximum number of incoming or outgoing connectors has been reached or if the number of connectors in general is too high.

Action: According to the cardinality restrictions, only a certain number of connectors of the specified relation class are permitted. To adhere to these restrictions, you have to delete connectors which start from or end in the specified object.

[acard-04]

Message: Unexpected value in the attribute "*<attribute>*" ("*<class>*"): "*<error expression>*" is not a valid value. "*<correct expression>*" is expected.

Cause: The specified attribute of the specified class contains an invalid value. The valid expression listed is expected instead.

Action: Contact your ADONIS administrator. The Administration Toolkit should be used to correct the invalid value.

[acard-05]

Message: The model contains *<number>* objects of the class "*<class name>*". A maximum of *<number>* objects are allowed.

Should the verification of the cardinalities be continued?

Cause: The verification of a model has determined that the maximum number of allowed objects of the specific class is too high.

Action: According to the defined cardinality restrictions, only a certain number of objects of the specified class is allowed. To fulfill these restrictions, you must delete the appropriate number of objects of this class.

[acard-06]

Message: The model contains *<number>* objects of the class "*<class name>*". <At least *number*> object(s) is/are required.

Should the verification of the cardinalities be continued?

Cause: The verification of a model has determined that the minimum number of required connectors of the specific class has not been reached.

Action: According to the defined cardinality restrictions, a minimum number of objects of the specified class is required. To fulfill these restrictions, you must create the corresponding number of objects of this class.

[acard-07]

Message: *<Object or class name>*:At least *<number>* connector(s) is/are required (*<direction>*).

Should the verification of the cardinalities be continued?

Cause: The verification of a model has determined that the minimal number of required connectors has not been reached.

The possibly shown direction ("incoming", "outgoing" or "all together") indicates whether the minimum number of incoming or outgoing connectors has not been reached or if the number of connectors in general is too low.

Action: According to the defined cardinality restrictions, a minimum number of connectors is required. To fulfill these restrictions, you must create additional connectors, which start from or end in the specified object.

[acard-08]

Message: *<Object or class name>*: At least*<number>* connector(s) of the type "*<relation class name>*" is/are required (*<direction>*).

Should the verification of the cardinalities be continued?

Cause: The verification of a model has determined that for the minimum number of required connectors of the specified class for the specified object has not been reached.

The possibly shown direction ("incoming", "outgoing" or "all together") indicates whether the minimum number of incoming or outgoing connectors has not been reached or if the number of connectors in general is too low.

Action: According to the defined cardinality restrictions, a minimum number of connectors of the specified relation class is required. To fulfill these restrictions, you must create additional connectors of the relation class, which start from or end in the specified object.

[acard-09]

Message: Error in the assignment of the attribute "*<attribute name>*" ("*<class name>*"): "*<expression>*" is not applicable to the relation class "*<relation class name>*"!

Cause: The specified cardinality restriction is not meaningful.

This message will be delivered for instance if the number of **incoming** connectors are restricted to a class, although the corresponding connectors of this class can only be **outgoing** (and not incoming).

Action: Remove this restriction.

[acnumchk-01]

Message: The key word "DOMAIN" is missing.

Cause: The LEO expression in the facet "AttributeNumericDomain" does not contain the required key word "DOMAIN".

Action: Rectify the value of the facet "AttributeNumericDomain", by adding the key word "DOMAIN" or contact your ADONIS administrator.

[acnumchk-02]

Message: The parameter "message" of the key word "DOMAIN" is missing.

Cause: The LEO expression in the facet "AttributeNumericDomain" does not contain the required parameter "message" for the key word "DOMAIN".

Action: Rectify the value of the facet "AttributeNumericDomain", by adding the parameter "message" to the key word "DOMAIN" or contact your ADONIS administrator.

[acnumchk-03]

Message: After the key word "DOMAIN", only the key word "INTERVAL" is authorised.

Cause: The LEO expression in the facet "AttributeNumericDomain" contains an invalid key word. After the key word "DOMAIN", only the key word "INTERVAL" is authorised.

Action: Rectify the value of the facet "AttributeNumericDomain", by also using the key word "INTERVAL" after the key word "DOMAIN" or contact your ADONIS administrator.

[acnumchk-04]

Message: The parameter "lowerbound" of the key word "INTERVAL" is missing.

Cause: The LEO expression in the facet "AttributeNumericDomain" does not contain the required parameter "lowerbound" for the key word "INTERVAL".

Action: Rectify the value of the facet "AttributeNumericDomain", by adding the parameter "lowerbound' to the key word "INTERVAL" or contact your ADONIS administrator.

[acnumchk-05]

Message: The parameter "upperbound" of the key word "INTERVAL" is missing.

Cause: The LEO expression in the facet "AttributeNumericDomain" does not contain the required parameter "upperbound" for the key word "INTERVAL".

Action: Rectify the value of the facet "AttributeNumericDomain", by adding the parameter "upperbound" to the key word "INTERVAL" or contact your ADONIS administrator.

[acnumchk-06]

Message: The value of the parameter "upperbound" is smaller than the value of the parameter "lowerbound".

Cause: The value of the parameter "upperbound" of the key word "INTERVAL" is smaller than the value of the parameter "lowerbound" of the same key word.

Action: Change the value of the parameter "lowerbound" and "upperbound" so that the value of the parameter "upperbound" is greater than or equal to the value of the parameter "lowerbound" or contact your ADONIS administrator.

[acnumchk-07]

Message: The value is not in one of the specified intervals.

Cause: The specified value is not within the area valid for the attribute.

Action: Change the value so that it is in an interval specified in the facet "AttributeNu-mericDomain".

[acoexhan-01]

Message: No expression!

Cause: The value of the edited EXPRESSION attribute does not contain any expression, i.e. no formula or possible constant. This way no result can be calculated for the EXPRESSION attribute.

Action: Specify an expression for the calculation of the result.

[acoexhan-02]

Message: The result of the expression is not of the type specified in the attribute definition!

Expected type: "*<result type1>*"

Delivered type: "*<result type2>*".

Cause: The expression of the EXPRESSION attribute delivers a result, which is of another type than the one specified in the attribute definition. This is why you cannot use the result.

Action: Specify an expression which delivers a result matching to the type. If necessary, insert a conversion operator (VAL/STR) in this formula .

[acoexhan-03]

Message: The calculated value is too long/big to save in the database!

Cause: The result of an EXPRESSION attribute returned a value which is too long/big to save in the database.

Action: Define an expression resulting in a shorter/smaller value.

[acoexhan-04]

Message: The expression is too long to store in the database!

Cause: The EXPRESSION is too long and therefore cannot be saved into the database.

Action: Define a shorter expression to calculate the result.

[acoexman-01]

Message:	The syntax definition is EXPR type:<ResultType> [expr: [fixed:]<expression>, however the attribute value is "*<attribute value>*"!
Cause:	The value defined for the EXPRESSION attribute is syntactically wrong.
Action:	Change the value so that the syntax instructions come up.

[acoexpar-01]

Message:	Internal error!
Cause:	An unexpected error code has been delivered in the EXPRESSION attribute during the call of a core function. This can occur when a number which is not an ID is given instead of an ID.
Action:	Verify and correct the formula.

[acoexpar-02]

Message:	Attribute "*<attribute name>*" (*<Class name>*) does not exist!
Cause:	In the EXPRESSION attribute, a string, whose value is not an attribute name in the specified class, has been assigned during the call of a function, which expects an attribute name as a parameter.
Action:	Verify and correct the formula.

[acoexpar-03]

Message:	The attribute "*<attribute name>*" (*<class name>*) cannot be accessed because of its types in the expressions!
Cause:	You are trying to sort an attribute of the type RECORD or PROFILEREF.
Action:	Verify and correct the formula.

[acoexpar-04]

Message:	Error in the referenced attribute "*<attribute name>*" of the objects "*<object name>*" (*<class name>*) in "*<model name>*" (*<model type>*)!
Cause:	The EXPRESSION attribute accesses to another EXPRESSION attribute, in which an error is delivered instead of a value.
Action:	Verify and correct the formula of the referenced EXPRESSION attribute specified in the error message.

[acoexpar-05]

Message:	At the moment, the object "*<Object name>*" (*<class name>*) in "*<model name>*" (*<model type>*) cannot be accessed!

Cause: The EXPRESSION attribute attempts to access an object which is not in the memory yet. This conflict can only occur during the loading of a model and will disappear itself after the loading process.

Action: No action required or action is done automatically.

[acoexpar-06]

Message: No object "*<object name>*" (*<class name>*) exists in "*<model name>*" (*<model type>*)!

Cause: In the EXPRESSION attribute, a string, with a value which is not the name of an object of the specified class, has been assigned during the call of a function, which has an object name as parameter.

Action: Verify and correct the formula.

[acoexpar-07]

Message: In the model type "*<model type>*", there is no class "*<class name>*"!

Cause: In the EXPRESSION attribute, a string whose value is not the name of a class in the specified model type, has been assigned during the call of a function, which expects a class name as a parameter.

Action: Verify and correct the formula.

[acoexpar-08]

Message: Access to the specific attribute "*<attribute name>*" is not authorised!

Cause: In the EXPRESSION attribute, you have attempted to sort this attribute, which leads to an endless recursion.

Action: Verify and correct the formula.

[acoexpar-09]

Message: The attribute "*<attribute name>*" (*<class name>*) is not of numerical type!

Cause: In the EXPRESSION attribute, you have attempted to calculate an arithmetical aggregate function via a non numerical attribute.

Action: Verify and correct the formula.

[acoexpar-10]

Message: The ID list is faulty!

Cause: In the EXPRESSION attribute, a string, which contains no or not only IDs, is assigned to a function, which expects a string with a list of IDs as parameter.

Action: Verify and correct the formula.

[acoexpar-11]

Message: *<Function name>* cannot be evaluated without access to the Modelling Component!

Cause: In the EXPRESSION attribute, you have attempted to access the "modelling" message port, which is not available.

Action: None. This error does not occur in the Business Process Management Toolkit.

[acoexpar-12]

Message: No column "*<column name>* exists in the record attribute "*<attribute name>*" (*<class name>*)!

Cause: In the EXPRESSION attribute, a string, whose value is not the name of a column of the specified RECORD attribute, is assigned during the call of a function, which expects as a column name of a RECORD attribute as parameter, .

Action: Verify and correct the formula.

[acoexpar-13]

Message: The column "*<column name>*" of the record attribute "<attribute name>" (*<class name>*) is not of numerical type!

Cause: In the EXPRESSION attribute, you have attempted to calculate an arithmetical aggregate function using a non numerical attribute.

Action: Verify and correct the formula.

[acoexpar-14]

Message: The column "*<column name>*" of the record attribute "*<attribute name>*" (*<class name>*) is not of the type *<type name>*!

Cause: In the EXPRESSION attribute, a RECORD column is accessed, whose attribute type is different from the attribute type expected.

Action: Verify and correct the formula.

[acoexpar-15]

Message: Syntactical error in the AQL expression!

Cause: In the EXPRESSION attribute, a function is called, which expects an AQL expression with a string as parameter. The value assigned is not a syntactically correct AQL expression.

Action: Verify and correct the formula.

[acoexpar-16]

Message: Reference missing in the object "*<Object name>*" (*<class name>*), record attribute "*<attribute name>*", column "*<column name>*" [*<column index name>*]!

Cause: In the EXPRESSION attribute, an INTERREF attribute, which must contain a valid reference, is accessed. However the INTERREF attribute contains either no reference or no valid reference.

Action: Change the specified INTERREF attribute, so that it contains a valid reference.

[acoexpar-17]

Message: The attribute "*<attribute name>*" (*<class name>*) is neither of numerical nor of time type!

Cause: In the EXPRESSION attribute, you have attempted to calculate an arithmetical aggregate function via a non numerical and non time attribute.

Action: Verify and correct the formula.

[acoexpar-19]

Message: The AQL expression cannot be evaluated!

Cause: In the EXPRESSION attribute, you have attempted to evaluate an AQL expression, without having the message port required for it available.

Action: None. This error does not occur in the Business Process Management Toolkit.

[acoexpar-20]

Message: Cyclic dependence when accessing to the attribute "*<attribute name>*" of the object "*<object name>*" (*<class name>*) in "*<model name>*" (*<model type>*)!

Cause: Several EXPRESSION attributes cyclically access another, so that an endless recursion appears.

Action: Verify and correct the formulas of the concerned EXPRESSION attribute.

[acoexpar-21]

Message: In the model type *<model type>* there is no relation type "*<class name>*"!

Cause: In the EXPRESSION attribute, a string, which is not the name of a relation type, is assigned to a function, which expects the name of a relation type as parameter.

Action: Verify and correct the formula.

[acoexpar-22]

Message: The ID list when calling the *<function name>* is empty!

Cause: In the EXPRESSION attribute, an empty string is assigned to a function which expects a non-empty ID list as a parameter. The reason can be, that an INTERREF attribute is accessed in which no valid reference is contained.

Action: Verify and correct the formula and the included INTERREF attributes.

[acoexpar-23]

Message: The attribute "*<attribute name>*" (*<class name>*) is not of the type *<type name>*!

Cause: In the EXPRESSION attribute, an attribute which is different from the expected attribute type, is being accessed using a function.

Action: Verify and correct the formula.

[acoexpar-24]

Message: At least one object is referenced by the INTERREF attribute "*<attribute name>*" of the object "*<object name>*" (*<class name>*) and you cannot access this object as "*<model name>*" (*<model type>*) is not loaded!

Cause: In the EXPRESSION attribute, you have attempted to find IDs referenced objects, without having opened the target model.

Action: Open the target model. If the expression could be correctly calculated once, the last valid result will be delivered instead of this error.

[acoexpar-25]

Message: An object "*<object name>*" (*<class name>*) is referenced by the INTERREF attribute "*<attribute name>*" of the object "*<object name>*" (*<class name>*) and this object does not exist in "*<model name>*" (*<model type>*)!

Cause: In the EXPRESSION attribute, you have attempted to find IDs referencing, no longer existing objects.

Action: Edit the INTERREF attribute concerned or create new objects, so that the INTERREF attribute contain valid references again.

[acoexpar-26]

Message: The INTERREF attribute "*<attribute name>*" of the object "*<object name>*" (*<class name>*) contains errors!

Cause: In the EXPRESSION attribute, you have accessed an INTERREF attribute, which contains an invalid value. This means that the database is corrupt.

Action: The ADONIS database may be recreated.

[acoexpar-27]

Message: The call of the *<function name>* can not be evaluated, as "*<model name>*" (*<model type>*) is not loaded!

Cause:	In the EXPRESSION attribute, you have attempted to sort information from a model which is not opened.
Action:	Open the specified model. If the expression could be correctly calculated once, the last valid result will be delivered instead of this error.

[acoexpar-28]

Message:	The call of the *<function name>* can not be evaluated, as there is no scope or referenced model!
Cause:	You have attempted to evaluate an object-related expression without having specified the scope. This can occur during the call of EVAL_EXPRESSION, when functions have been called, which refer to a specific object id and the parameter object has not been specified.
Action:	Specify the object id when calling EVAL_EXPRESSION.

[acoexpar-29]

Message:	The record attribute "*<attribute name>*" (*<class name>*) contains no row with index *<row number>*!
Cause:	In the EXPRESSION attribute, you have attempted to access a RECORD row which does not exist.
Action:	Verify and correct the formula or edit the specified RECORD attribute.

[acoexpar-30]

Message:	The attribute profile referenced by "*<attribute name>*" (*<class name>*) does not exist!
Cause:	In the EXPRESSION attribute, you have attempted to access to a referenced attribute profile, which has broken the attribute profile reference .
Action:	Verify and correct the formula or the attribute profile reference specified in the error message.

[aconfui-01]

Message:	An invalid customer number has been entered! You must use a combination of upper-case letters (A-Z), lowercase letters (a-z) or numbers (0-9) with a minimum of 5 and a maximum of 17 characters.
	In the expanded mode, you can use a minimum of 5 and a maximum of 24 characters.
Cause:	The customer number does not meet the specified conditions.
Action:	Enter a valid customer number.

[aconfui-03]

Message:	An error has occurred when saving the licence number!

Cause:	The licence number could not be saved to the file. There may not be enough free memory space available or the file is write-protected.
Action:	Create some free memory space or save the licence number to another file.

[aconfui-04]

Message:	No file name has been entered!
Cause:	No file name has been entered!
Action:	Enter a file name.

[aconfui-05]

Message:	Cannot load current configuration!
Cause:	An error occurred while the current configuration was read from the ADONIS database.
Action:	Quit ADONIS and re-start it. Should this error re-occur, contact your system administrator.

[aconfui-10]

Message:	Licence number does not correspond with customer number!
Cause:	The licence number entered cannot be used for configuring components when combined with the customer number listed.
Action:	Check the licence number entered. Should this error still occur, contact your system administrator or your ADONIS consultant.

[aconfui-12]

Message:	No licence number entered!
Cause:	You did not enter a licence number.
Action:	Enter the licence number for the new configuration of components of the ADONIS Business Process Management Toolkit.

[aconfui-13]

Message:	New configuration could not be saved in database.
Cause:	An error occurred when the new configuration was to be saved in the ADONIS database.
Action:	Quit ADONIS and re-start it. Should this error re-occur, contact your database or system administrator.

[aconfui-14]

Message:	The generation of the new licence number is not possible!
Cause:	An unexpected error has occurred during the generation of the licence number.
Action:	Contact your ADONIS administrator.

[aconfui-15]

Message:	No test licence number has been entered!
Cause:	No test licence number has been entered!.
Action:	Enter a test licence number.

[aconfui-16]

Message:	The test licence number could not be decoded!
Cause:	The licence number is not valid.
Action:	Enter a valid licence number.

[aconfui-17]

Message:	You must configure 1-99 users to specify the maximum number of users!
	In the extended mode, you can configure 1-9999 users.
Cause:	An invalid number of users has been entered.
Action:	Enter a valid number of users.

[aconfui-18]

Message:	The Configuration Toolkit cannot be started, since an ADONIS application is already active. Close the other ADONIS application in order to start the Configuration Toolkit.
Cause:	An ADONIS application is already active.
Action:	First close the other ADONIS application. Then start the Configuration Toolkit again.

[acregchk-01]

Message:	The key word "REGEXP" is missing.
Cause:	The LEO expression in the facet "AttributeRegularExpression" doesn't contain the required key word "REGEXP".
Action:	Rectify the value of the facet "AttributeRegularExpression", by adding the key word "REGEXP" or contact your ADONIS administrator.

[acregchk-02]

Message: The parameter "expression" of the key word "REGEXP" is missing.

Cause: The LEO expression in the facet "AttributeRegularExpression" does not contain the required parameter "expression" for the key word "REGEXP". In the parameter "expression", you must have the regular expression, which the attribute values must meet.

Action: Correct the value of the facet "AttributeRegularExpression", by adding the parameter "expression" to the key word "REGEXP", or contact your ADONIS administrator.

[acregchk-03]

Message: The parameter "message" of the key word "REGEXP" is missing.

Cause: The LEO expression in the facet "AttributeRegularExpression" does not contain the parameter "message" required for the key word "REGEXP". In the parameter "message", you must have the description text for the regular expression, which the attribute values must meet.

Action: Correct the value of the facet "AttributeRegularExpression", by adding the parameter "message" to the key word "REGEXP" or contact your ADONIS administrator.

[acregchk-04]

Message: The regular expression in the parameter "expression" is invalid.

Cause: The parameter "expression" in the facet "AttributeRegularExpression" contains an invalid regular expression.

Action: Rectify the value of the facet "AttributeRegularExpression", by giving a valid regular expression as value to the parameter "expression" of the key word "REGEXP" or contact your ADONIS administrator.

[acregchk-05]

Message: The expression does not meet the regular expression in the parameter "expression".

Cause: You have attempted to assign a value, which doesn't correspond to the defined regular expression.

Action: Change the value in such a way, that it meets the regular expression. If necessary, have a look at the help text of the corresponding attribute to find out if there is a description of the possible values.

[adbacc-01]

Message: *<Database error message>*Class: ADBACC, Stmt handle: *<Handle>*, Function: *<Function Name>*

Cause: A general database error occurred. The error message gives information about the cause of the error and contains the database error-code, which can be used to gain further information from the database error documentation.

Action: Exit ADONIS and start it again. If the error re-occurs, please write down the *<Database error message>*, *<Handle>* and *<Function Name>* and contact your system administrator.

[adbadm-01]

Message: *<Database error message>* Class: ADBADM, Stmt handle: *<Handle>*, Function: *<Function Name>*

Cause: A general database error occurred. The error message gives information about the cause of the error and contains the database error-code, which can be used to gain further information from the database error documentation.

Action: Quit ADONIS and start it again. If the error still occurs, please write down the *<Database error message>*, *<Handle>* and *<Function Name>* and contact your system administrator.

[adbsess-01]

Message: The network or database system is not available at the moment. The database connection has been aborted.

In the current session, no further database access is possible.

It is recommended to start the application again.

Cause: An error has occurred in the network connection.

Action: Try to start the application again. Should this error appear again, contact your system administrator.

[adbsess-02]

Message: Because of an unexpected database message, the connection with the database has been aborted.

In the current session, no further database access is possible.

It is recommended to start the application again.

Cause: Demands to the database system have been rejected by an error message.

Action: Click on the button "Show database message" and note the displayed error message and contact a system administrator or an ADONIS administrator.

Further access to the database is only possible once you have restarted the application.

[adistrib-01]

Message: Enter a numeric value for the expected valueperiod;

Cause: No number has been entered for the expected value in a normal distribution.

Action: Enter a numeric value for the expectation.

[adistrib-02]

Message: Enter a number greater than 0 for the standard deviation.

Cause: Either no number or a number less than or equal to 0 has been entered for the standard deviation value of the normal distribution.

Action: Enter a positive number for the standard deviation.

[adistrib-04]

Message: Enter a number greater than 0,00000001 for the expected value.

Cause: Either no number or a number less than or equal to 0.00000001 has been entered for the expected value in the exponential distribution.

Action: Enter an authorised number (i.e. greater than 0.00000001) for the expected value.

[adistrib-05]

Message: Enter a number for both the lower and upper bounds.

Cause: No number has been entered for either the lower or upper bound of the uniform distribution.

Action: Enter a numeric value for the lower and upper bounds.

[adistrib-06]

Message: The lower bound cannot be greater than the upper bound.

Cause: A smaller value has been entered for the upper bound of a uniform distribution compared to that entered for the lower bound.

Action: Enter numeric values for the upper and lower bounds such that the upper bound is greater than the lower bound.

[adistrib-08]

Message: The sum of the probabilities must add up to 1.

Cause: You have entered probability values for the discrete distribution where the sum of all probabilities is not 1.

Action: Specify the probabilities such that they add up to 1.

[adistrib-09]

Message: Enter a valid symbol name.

Cause: An invalid symbol name has been specified for the discrete distribution (i.e. one which contains invalid characters such as @ or is completely numeric.

Action: Enter an alphanumeric value for the symbol.

[adistrib-10]

Message: Enter a unique symbol name.

Cause: You have entered a symbol name which you have already used within this discrete distribution.

Action: Each symbol name within a discrete distribution must be unique - select a different symbol name.

[adistrib-11]

Message: Enter a value between 0 and 1. Please note that format is "0,n" or "0.n" - NOT ".n".

Cause: Either no number, or a number less than zero or greater than 1 has been entered for the probability value in the discrete distribution.

Action: Please enter a number between 0 and 1. (Format 0,n or 0.n).

[adistrib-13]

Message: Syntax error.

Cause: The distribution defined is not correct.

Action: Check the correct syntax for the distribution type.

[adistrib-14]

Message: Invalid number.

Cause: A non-numeric value has been entered as part of the distribution instead of a numeric value.

Action: Enter a numeric value.

[adistrib-15]

Message: Invalid symbol.

Cause: An invalid symbol name has been specified for the discrete distribution (i.e. one which contains invalid characters such as @ or is completely numeric.

Action: Enter an alphanumeric value for the symbol.

[adistrib-16]

Message: Ambiguous symbol.

Cause: A symbol name which is not unique within this discrete distribution has been entered.

Action: Ensure that every symbol name is unique within a discrete distribution.

[adistrib-17]

Message: No object selected!

Cause: No object was selected in resource assignment.

Action: Select the object(s) to be assigned to the resource.

[adistrib-19]

Message: Expression is syntactically incorrect!

Cause: The AQL expression for resource assignment is syntactically incorrect.

Action: Correct the AQL expression or use the input support to create an AQL expression with the correct syntax.

[adistrib-20]

Message: The current performer can only be selected alone.

Cause: You have attempted to use the expression "current performer" with other objects.

Action: Either select "current performer" or the other objects.

[adistrib-23]

Message: Expression syntactically incorrect!

Cause: The AQL expression in the "Performer" field is syntactically incorrect.

Action: Check the AQL syntax or re-enter the performer required using the input support.

[adistrib-28]

Message: Invalid value.

Times must be entered in the format yy:ddd:hh:mm:ss.

Cause: You have entered a time value in an invalid format.

Action: Enter a time value in the correct format (yy:ddd:hh:mm:ss).

[adistrib-29]

Message: Invalid value.

The year may not be higher than 999999.

Cause: Too high a value has been entered for the year.

Action: Enter value less than or equal to 999999 for the year.

[adistrib-30]

Message: Error while creating a new inter-reference:

The maximum number of references for this attribute to model type *<model type>* is *<number>*.

Cause: The maximum number of references for this attribute to models of the specified type has been exceeded.

Action: Delete another reference or contact your ADONIS administrator or consultant if you would like to increase this maximum number.

[adistrib-31]

Message: Error while creating a new inter-reference:!

The maximum number of references for this attribute

to objects of class*<class name>* in model type <model type> is *<number>*.

Cause: The maximum number of object references from this attribute has been exceeded.

Action: Delete any unnecessary object references or contact your ADONIS consultant.

[adistrib-34]

Message: Error when creating a new inter model reference:

The maximum number of references authorised for this attribute is *<number>*!

Cause: You have attempted to create a new inter model reference, although the INTERREF attribute already contains the maximum number of references authorised for this attribute.

Action: If there is an unnecessary reference in this INTERREF attribute, you must delete it first.

[adistrib-35]

Message: The transition condition contains errors:

<Error>

Cause: The transition condition is syntactically not correct.

Action: Correct the error.

[adistrib-36]

Message: The value for the process time contains errors:

The expected format is YY:DDD:HH:MM:SS!

Cause: The format of the process time in comparison within the transition condition is not correct.

Action: Enter a process time in ADONIS time format YY:DDD:HH:MM:SS.

[adistrib-37]

Message: The value for the day time contains errors:

The expected format is hh:mm:ss!

Cause: The format of the day time in comparison within the transition condition is not correct.

Action: Enter a day time in format hh:mm:ss.

[adistrib-38]

Message: The value for "Date-time" is invalid. The value must be in format "YYYY:MM:DD hh:mm:ss" and you must keep within the following value range:

Year: 0001 - 9999, month: 01 - 12, day: 01 - 31

Hour: 00 - 23, minute: 00 - 59, second: 00 - 59.

Cause: You have entered an invalid value in an attribute of the type "Date-time" (i.e. invalid format or outside the authorised value range .

Action: Enter the value using the format "YYYY:MM:DD hh:mm:ss" and observe the above described value range.

[adistrib-39]

Message: The value for "Year" is too big. Values between 0001 and 9999 are authorised.

Cause: In an attribute of type "Date-time", you have entered a value for "Year", which is outside the authorised value range.

Action: Enter a value between 0001 and 9999 for the year.

[adistrib-40]

Message: The value for "Date" is invalid. The value must be in format "YYYY:MM:DD" and you must keep to the following value range:

Year: 0001 - 9999, month: 01 - 12, day: 01 - 31.

Cause: You have entered an invalid value for the date in an attribute of type "Datum" (DATE) (i.e. invalid format or outside the authorised value range.

Action: Enter the value in format "YYYY:MM:DD" and observe the above described value range.

[adistrib-41]

Message: The value for "Days" is too big.

The maximum allowed value is 364 days.

Cause: In an attribute of type "Time", you have entered a value for "Days", which is outside the valid range.

Action: Enter a value between 0 and 364 for the days.

[adistrib-42]

Message: The value for "Hours" is too big.

                 The maximum allowed value is 23 hours.

Cause: In an attribute of type "Time", you have entered a value for "Hours", which is outside the valid range.

Action: Enter a value between 0 and 23 for the hours.

[adistrib-43]

Message: The value for "Minutes" is too big.

                 The maximum allowed value is 59 minutes.

Cause: In an attribute of type "Time", you have entered a value for "Minutes", which is outside the valid range.

Action: Enter a value between 0 and 59 for the minutes.

[adistrib-44]

Message: The value for "Seconds" is too big.

                 The maximum allowed value is 59 seconds.

Cause: In an attribute of type "Time", you have entered a value for "Seconds", which is outside the valid range.

Action: Enter a value between 0 and 59 for the seconds.

[aeagents-02]

Message: Error in a combining function of an agent.

Cause: A result function of type "Animation" or "Special formula" could not be initialised. The agent concerned is configured incorrectly. Perhaps the agent was imported from an ADL file which contained errors.

Action: Cancel the simulation, open the start window of the simulation again and click on the button "Agents". Open the configuration window of each agent marked here as incorrect and correct the configuration errors. As soon as you close the configuration windows again you will be informed if you have overlooked possible configuration errors. Correct these and start the simulation again. **Note:** To avoid similar problems in the long run, you should export the models simulated to an ADL file and inform your ADONIS consultant.

[aeagents-05]

Message: All specified result functions of the current agent could not be saved. (Too many functions have been specified). The configuration window of the current agent will show you which functions were saved successfully.

Cause: You tried to assign too many result functions to an agent. On average, about 15 functions can be assigned to an agent.

Note: The exact number depends on both the types of result functions assigned and the manner in which result functions, which were already assigned, were configured. There is no fixed upper limit for the number of possible result functions in an agent.)

Action: Check which result functions had been assigned successfully and which were not by using the tab cards in the configuration window of the agent and those which were not. Remove unnecessary result functions.

[aeagents-06]

Message: The result names of the following agent are not unique: "*<Agent name>*"

Cause: An agent is configured incorrectly. Perhaps the agent was imported from an invalid ADL file or you used a pre-defined agent, the definition of which is incorrect.

Action: Open this agent's configuration window and change the result name of all result functions so that all result names are unambiguous.

[aeagents-07]

Message: The following result will not be generated: "*<Name of the result function concerned>*: *<Name of the result not generated>*"

Cause: A result, which serves as basis for the result function listed, cannot be generated. The result function may have been configured incorrectly.

Action: Cancel the simulation and then open the start window of the simulation again. Click on the button "Agents". and correct the configuration of all the agents marked as incorrect in the window "Agents overview".

[aeagents-08]

Message: The configuration string of an agent could not be decoded. The following result type was not recognised: "<Unknown result type>"

Cause: An agent is configured incorrectly. Perhaps the agent was imported from an invalid ADL file or you used a predefined agent, the definition of which is incorrect.

Action: Correct the configuration of all the agents marked here as incorrect in the window "Agents Overview" (menu "Edit" - option "Agents").

[aeagents-09]

Message: An agent result cannot be determined due to one of the following reasons: the agent is located in a recursively called process, the agent is located in a subprocess which was called in a parallel way within the execution of one and the same main process.

Result: "*<Result name>*"

(Process: "*<Process name>*")

Cause: The model structure simulated cannot be evaluated by one of the agents created. There are two possible reasons for this:

1. the model in which the agent was created is called recursively. That is, the model calls itself again (maybe via other subprocesses).

2. the agent was created in a subprocess and this subprocess is called from a super or- dinated process in a parallel way.

To put it in a more general context this means that two (or more) instances of a (sub)process which orginate from the same main process instance, are executed simul- taneously.

Action: Avoid recursive model structures or change your model structures accordingly.

Avoid calling subprocesses in a parallel way within a model structure. One way to avoid a parallel structure is to move the agent from the subprocess, which is called in a parallel way, to the directly super ordinated process and assign the subprocess call objects as observed objects to the agent.

[aeagents-10]

Message: Error when evaluating the simulation through an agent. The generation of the following result has been deactivated to avoid further error messages: "*<Result name>*"

Cause: An unexpected error occurred while the result listed was calculated. Generation of this result has been cancelled.

Action: Note the result name listed, export the models simulated to an ADL file and inform your ADONIS consultant.

[aeasubag-02]

Message: The following result generating function of an agent has not been configured completely and is thus deactivated: "*<result name>*"

Cause: You use a pre-defined agent. This agent's definition is incorrect

Action: Cancel the simulation, then open the start window of the simulation again and click on the button "Agents". The window "Agent overview" opens up. Correct the configuration of all agents marked as incorrect here.

    Hint: To avoid this error in the future, you should note down the result name listed (and the agent type concerned, if you know it) and inform your ADONIS consultant. The agent definition in the Library Management of the ADONIS Administration Toolkit is probably in- correct and must be corrected.

[aeasubag-03]

Message: The initialisation of an agent failed. The result "*<result name>*" cannot be initialised be- cause "*<name>*" is not a valid name. Please contact your ADONIS administrator.

Cause: You are using a predefined agent. An invalid name was used within the definition of this agent.

Action: Note down the result and the name listed (and also the agent type concerned, if you know it) and inform your ADONIS consultant. The agent definition in the Library Man- agement of the ADONIS Administration Toolkit must be corrected.

[aeasubag-04]

Message: The following result cannot be generated because not all results on which this result is based are available:

"*<Name of the Result Function Concerned>*"

Cause: The calculation formula in the result function listed is incorrect. It references other result functions which do not exist

Action: Open the configuration window of the concerned Result Function and correct the calculation formula.

[aeasubag-05]

Message: The following result cannot be generated because of a syntax error in the required formula:

"*<Result function name>*"

Cause: The calculation formula for the result function listed is incorrect. It contains a syntax error.

Action: Open the configuration window of the agent concerned and correct the calculation formula.

[aeasubag-06]

Message: Some specified calendars are too complex. The following result cannot be generated:

"*<Result name>*"

Cause: The calendars of performers, resources and/or agents in the models simulated are too complex. The result listed(which is always of the type "Workload") therefore cannot be calulated

Action: Use a smaller number of different day profiles in the calendars.

[aeasubag-07]

Message: "*<Result name>*"

This result cannot be generated. (Possibly other results which are required for the generation are not available under the current settings.)

Cause: An agent's result could not be generated. This may be for one of the following reasons:

1. The result listed could not be initialised. In this case, you already received an appropriate error message at the beginning of the simulation.

2. The result was calculated based on other results. One or more of these other results is not available.

Either an error occurred while the result not available was being calculated (in this case, you already received an appropriate error message at the beginning of the simulation), or the result not available cannot be displayed under the current settings of the result representation (That is the most probable cause.)

Action: Depending on the cause:

1. Correct the error that was originally reported.

2. Change the settings concerning the result representation.

Please note that some results can only be represented "per *<time period>*" (such as "per year"). As long as the option "per process" is selected, these results will not be available. Then display the results of the agent concerned again.

[aeasubag-08]

Message: The formula for the result "*<result name>*" contains too many different result names. The formula cannot be evaluated.

Cause: A calculation formula may contain at most 26 different results. This upper limit was exceeded by the calculation formula of the result function listed.

Action: Simplify the calculation formula.

[aexpappl-01]

Message: The attribute "Documentation configuration" contains errors (row *<no.>*)!

The format model has been ignored, since a required attribute is missing: *<attribute name>*

Cause: The content of the library attribute "Documentation configuration" is faulty: An attribute of the class "__LibraryMetaData__" of the BP library is referenced but does not exist.

Action: Contact your ADONIS administrator.

[aexpappl-06]

Message: Error during the creation of the help file.

Cause: These temporary files are created in the current directory or in the directory at which the enviroment variable TEMP is pointing.

Action: Ensure that the enviroment variable TEMP exists and points to a directory in which enough memory space is available and in which you are allowed to create files.

[aexpappl-08]

Message: The file *<File name>* does not exist!

Cause: During the ADONIS customising, it was determined that one or more files (e.g. graphic logo files) should be copied from the installation directory to the target directory. The file specified in the error message could not be opened.

Action: Verify your ADONIS installation using the file list and create or complete if necessary missing or damaged files.

> **Hint:** The documentation process will continue despite this error but the generated documentation contents will not contain data related to the damaged or missing files e.g. a logo.

[aexpappl-10]

Message: The attribute "documentation configuration" is faulty!

The name of the format model is missing, the format model has been ignored.

Cause: In the attribute "documentation configuration", no name was specified during the definition of a format model (using a FORMAT or an EXPORT element).

Action: Correct this error using the Administration Toolkit or contact your ADONIS administrator.

[aexpappl-13]

Message: The attribute "documentation configuration" contains errors (row *<no.>*)!

The element "*<element name>*" is faulty.

Cause: In the attribute "Documentation configuration", an invalid name was entered.

Action: Correct the spelling of the element name or delete the invalid element using the Administration Toolkit or contact your ADONIS administrator.

[aexpappl-14]

Message: The attribute "Documentation configuration" contains errors (row *<no.>*)!

SOURCE elements must always be inside an EXPORT element.

Cause: In the attribute "documentation configuration", an element was entered at the wrong position.

Action: Correct the error using the Administration Toolkit or contact your ADONIS administrator.

[aexpappl-15]

Message: The attribute "Documentation configuration" contains errors (row *<no.>*)!

The SOURCE element "*<element name>*" does not exist.

Cause: In the attribute "Documentation configuration", an invalid element name was given to a SOURCE element.

Action: Correct the element name using the Administration Toolkit or contact your ADONIS administrator.

[aexpappl-16]

Message: The attribute "Documentation configuration" contains errors (row *<no.>*)!

LIBRARY elements must be inside SOURCE "Model" elements.

Cause: In the attribute "Documentation configuration", an element was entered at the wrong position.

Action: Correct the error using the Administration Toolkit or contact your ADONIS administrator.

[aexpappl-17]

Message: Error in the attribute "Documentation configuration".

The attribute "*<attribute name>*" does not exist.

Cause: The content of the library attribute "Documentation configuration" is faulty: An attribute from the class "__LibraryMetaData__" of the BP library is referenced but does not exist.

Action: Contact your ADONIS administrator.

[aexpappl-18]

Message: Error in the attribute "Documentation configuration".

Characteristics *<Characteristics name>*. The attribute *<attribute name>* has the wrong type.

Cause: In the library attribute "Documentation configuration", an attribute of the class "__LibraryMetaData__" of the BP library is referenced but has the wrong type.

Action: Contact your ADONIS administrator.

[aexpappl-19]

Message: Error in the attribute "Documentation configuration".

Characteristics *<characteristics name>*.

The value "*<value denomination>*" is not allowed in the attribute "*<attribute name>*".

Valid values are: *<value area>*

Cause: In the library attribute "Documentation configuration", an enumeration attribute of the class "__LibraryMetaData__" of the BP library is referenced but its value area contains invalid values.

Action: Contact your ADONIS administrator.

[aexpappl-20]

Message: Error in the attribute "Documentation configuration".

The value "*<value denomination>*" is not allowed in the attribute "*<attribute name>*".

Valid values are: *<value area>*

Cause: In the library attribute "Documentation configuration", an enumeration attribute of the class "__LibraryMetaData__" of the BP library is referenced but its value is not in the valid area.

Action: Contact your ADONIS administrator.

[aexpappl-21]

Message: Error in the attribute "Documentation configuration".

Characteristics *<characteristics name>*.

The attribute "*<attribute name>*" is not a class attribute.

Cause: The content of the attribute "Documentation configuration" is faulty: An attribute from the class "__LibraryMetaData__" of the BP library is referenced but is not a class attribute.

Action: Contact your ADONIS administrator.

[aexpappl-22]

Message: The file "*<file name>*" could not be opened!

Cause: A file required for the documentation generation could not be opened.

Action: Verify your ADONIS installation using the file list and replace or complete if necessary missing or damaged files.

[aexpappl-23]

Message: The file "*<file name>*" could not be overwritten! Make sure that the directory to which the file should be written does exist and that you own the required write access rights.

Cause: The target file specified for the documentation could not be opened with write access.

Action: Verify the write access to the target directory, in which the file will be generated. You can create documentation only in directories for which you own write access. Then verify, if a file with the specified name exists already in the target directory and delete this file if necessary.

[aexpappl-25]

Message: Invalid Value:

The value "*<Value>*" could not be saved.

Cause: The configuration of the documentation generation is faulty.

Action: Contact your ADONIS administrator.

[aexpappl-26]

Message: Error in the application library:

No documentation export was defined for the menu item "*<name>*".

Cause: No format model exists in the attribute "Documentation configuration" for the menu item called (using a FORMAT or EXPORT element).

Action: Control the spelling of the menu items called and if necessary correct errors or adapt the attribute "documentation configuration" using the Administration Toolkit or contact your ADONIS administrator.

[aexpappl-27]

Message: The class "__LibraryMetaData__" does not exist.

Cause:	The class "__LibraryMetaData__" required for the documentation generation does not exist in the BP library .
Action:	Contact your ADONIS administrator.

[aexpappl-28]

Message:	The user settings could not be saved to the database, since the user profile exceeds the maximum length.
Cause:	The database attribute to save the user-specific settings is not large enough to store all settings.
Action:	The user settings could not be saved to the database.

Hint: The settings carried out for the documentation are lost after you close ADONIS. But the settings are valid for the documentation which will be immediately generated.

[aexpappl-29]

Message:	An error has occurred during the conversion of the library attribute "Documentation configuration": The attribute "<*attribute name*>" could not be created in the class "_Library MetaData_".
Cause:	An attribute with the specified name already exists in the class "_Library MetaData_".
Action:	Contact your ADONIS administrator.

[aexpappl-30]

Message:	Error in the command "EXEC_ACFILTER"! The attribute "<*attribute name*>" does not exist within the class "_Library MetaData_".
Cause:	The specified attribute is referenced within the AdoScript command EXEC_ACFILTER, but does not exist.
Action:	Contact your ADONIS administrator.

[aexpappl-31]

Message:	The converted library attribute "Documentation configuration" could not be saved to the database as it is too long.
Cause:	The configuration of the Documentation Component is too big to be converted.
Action:	Carry out a manual conversion of the library attribute or simplify the configuration by deleting single menu items.

[aexpappl-32]

Message:	Error in the application library:
	No settings dialogue has been defined.

Cause: No format model (using the DIALOG element) exists in the attribute "documentation configuration" for the called menu item.

Action: Correct the error if necessary (adapt the attribute "documentation configuration" using the Administration Toolkit) or contact your ADONIS administrator.

[aexpappl-33]

Message: The settings for the attribute- and class filter could not be saved, as the length of the settings exceeds the maximum length allowed for the attribute "*<Attribute name>*" of the class "__LibraryMetaData__".

Cause: The settings you have defined in the attribute and class filter cannot be saved to the ADONIS database, because of their size.

Action: When carrying out the settings in the attribute and class filter, use the predefined mode (button "Load mode") to reduce the complexity and therefore the size.

[aexpappl-34]

Message: Error in the attribute "documentation configuration",

Characteristics "*<characteristics>*".

The value "*<value>*" of the attribute "*<attribute name1>*" has not been defined in the attribute "*<attribute name2>*".

Cause: The specified value can also be the name of an attribute (*attribute name2*) which contains the domain.

Action: Define the value in the domain or change the value of the attribute to a value from the domain.

[aexpappl-35]

Message: The target directory "*<directory name>*" is not a valid directory.

Cause: The specified directory name is invalid.

Action: Specify a valid directory name.

(The characters **/ \ : * ? < >** and **|** are not allowed in directory names.)

[aexpappl-36]

Message: No LOG file could be created!

Documentation generation will continue.

Cause: The creation of the LOG file is wrong.

Action: Verify if a file with this name exists already and if it is write-protected or already opened.

If necessary, verify if you possess write rights on the target directory and if there is enough memory available in the hard disc.

[aexpappl-37]

Message: The file "*<file name>*" is referenced in the "documentation configuration" with "copy", but has not been specified in the "requirefile" area.

Contact your ADONIS administrator.

Cause: The specified file has been listed in the library attribute "documentation configuration" as a copy parameter and not as a "requirefile" parameter.

Action: In the ADONIS Administration Toolkit, define the specified file in the library attribute "documentation configuration" as a "requirefile" parameter or contact your ADONIS administrator.

[aexpappl-38]

Message: The ADONIS installation directory cannot be used as a target directory.

Cause: You have selected the ADONIS installation directory as the target directory for the documentation generation or the export. This is not allowed for security reasons.

Action: Select another target directory. If you have write access rights only on the ADONIS installation directory, create a sub directory in it and select a file as target file in this sub directory.

[aexpappl-39]

Message: The format of the font table "<file name>" is not valid.

The default font table will be used.

Cause: A codepage conversion table you wanted to use contains an error and is thus invalid.

Action: Contact your ADONIS administratorParallelity

[aexport-01]

Message: Unable to open export file.

Cause: The file to be exported could not be created, since the data carrier/medium specified (e.g. floppy disk) is write-protected or does not contain enough free space.

Action: Remove the write-protection from the data carrier/medium or create some new space or assign a valid file name and restart the export.

[aexport-05]

Message: No models were exported.

Cause: Due to errors during the loading process no models were exported.

Action: Quit ADONIS and restart it. If the error message re-occurs during the next attempt, please inform your database administrator.

[aexport-06]

Message: Unknown model type.

Cause: An unknown model type was encountered during the export process. Possibly, the connection to the ADONIS database was aborted.

Action: Quit ADONIS and re-start it. If the error message re-occurs during the next attempt, please inform your database administrator.

[aexport-07]

Message: Unknown library type.

Cause: An unknown library type was encountered during the export process. Possibly, the connection to the ADONIS database was aborted.

Action: Quit ADONIS and re-start it. If the error message re-occurs during the next attempt, please inform your database administrator.

[aexport-08]

Message: Error in export: *<Error>*

Cause: The error listed occurred during the export process. Possibly, the connection to the ADONIS database has been aborted.

Action: Quit ADONIS and re-start it. If the error message re-occurs during the next attempt, please inform your database administrator.

[aexport-10]

Message: Error writing to file "*<filename>*".

Cause: During the export process an error occurred while writing to the file specified.

Action: There is possibly not enough space left on the data carrier/medium. If there is still enough space, start the export process again.

[aexport-11]

Message: Error in library "*<library name>*".

Do you want to continue anyway?

Cause: During the export process an error occurred either while a library was about to be loaded or while determining the libraries assigned.

Action: Cancel the export process and re-start it. If you continue the export process, the ADL file will not contain the relevant model groups access information.

[aexport-12]

Message: Error when exporting files belonging to the library!

Cause: When loading a file belonging to the library from the ADONIS database, a serious error has occurred. The connection to the database may be aborted.

Action: Close ADONIS, start it again and repeat the export. Should the message appear again, contact your database administrator.

[afile-01]

Message: The file <*file name*> could not be opened with read access.

Cause: An error occurred, when attempting to open the specified file with read access.

Action: Verify, whether the specified directory as well as the specified file exist and whether you possess the required access rights to this file.

[afile-02]

Message: The file <*file name*> could not be opened with write access.

The file may be used by another program.

Cause: An error occurred, when attempting to open the specified file with write access.

Action: Verify, whether the specified file is currently opened, whether the specified directory as well as the specified file exist and whether you possess the required access rights for this file.

[afile-03]

Message: Error when reading the file "<*file name*>".

Cause: An error occurred when reading the specified file.

Action: Repeat the process. Should the error appear again, manually copy the specified file to the target directory. If successful, verify your ADONIS installation using the file list. If necessary, reinstall ADONIS.

[afile-04]

Message: Error when reading the file "<*file name*>".

Cause: An error occurred when writing the specified file to the target directory.

Action: Verify if there is enough memory space available in the target directory. If the target directory is a network drive, check if there is still a connection. Then repeat the process.

[afile-05]

Message: The file "<*file name*>" could not be deleted.

Cause: An error has occurred when deleting the specified file.

Action: Verify, if the specified directory and/or the specified file exist, if you have the required rights to delete this file and if the file is already opened.

[afile-06]

Message: The file "*<file name>*" does not exist.

Cause: The specified file could not be found.

Action: Verify if the specified directory and/or the specified file exist and whether you have the required rights to access this file.

[afile-07]

Message: The file "*<file name>*" exists already .

Cause: You have attempted to create or copy the specified file.

Action: The file will be created only if the already existing file has been renamed or deleted.

[afile-08]

Message: The file "*<file name>*" cannot be closed.

Cause: An error has occurred when trying to close the specified file.

Action: Repeat the process or contact your ADONIS administrator.

[afile-09]

Message: No temporary file could be created.

Cause: An error has occurred when trying to create a temporary file in the temporary folder.

Action: Verify if enough memory space is available in the hard disk, if the system variable "TEMP" is set and if you have the required access rights for the directory specified in the system variable "TEMP".

[afile-10]

Message: The file name "*<file name>*" is invalid.

Cause: The specified file name is not in a correct format or contains invalid characters.

Action: When entering file names, make sure that total length of paths and file names does not exceed 120 figures and no more than seven directories are contained in the path. Moreover, the characters **/ \ : * ? < >** and **|** are not valid.

[afile-11]

Message: The file "*<file name>*" could not be renamed.

Cause: An error has occurred when attempting to rename the specified file.

Action: Verify if a file with the same name already exists . If necessary assign another file name.

[afile-12]

Message: The file "*<file name>*" could not be moved.

Cause: An error has occurred when attempting to move the specified file.

Action: Verify if the specified directory exists and if a file with the same name is contained in this directory.

[afile-13]

Message: The specified database directory is invalid.

Cause: The database directory you have specified could not be found.

Action: Verify the specified database directory and correct your entry accordingly.

[afile-14]

Message: The file "*<file name>*" could not be accessed, since it is not open!

Cause: You have tried to access a file which is not open.

Action: Open the file before you access it.

[afile-16]

Message: The file "*<file name>*" already exists for the library "*<library name>*".

Cause: You have tried to import or to copy the specified file to the folder of the specified application library.

Action: The file can only be imported or copied with the specified name once the file already existing in the folder of the application library has been renamed or deleted.

[afile-17]

Message: The name of the file "*<file name>*" is too long.

Cause: The file *<file name>* could not be stored in the database becuase the name of the file is too long.

Action: Adjust the length of the file name accordingly. Note that the length of the external file name within the library (including the file extension) and the length of the library name must not exceed 249 characters.

[afilemgt-01]

Message: No file name has been entered!

Cause: You have not specified a file name for the export file.

Action: Enter the name of the export file.

[afilemgt-02]

Message: The specified directory name is invalid!

Cause: For the export of a file, you have specified a path which does not exist.

Action: Enter a valid (=existing) path for the export.

[afilemgt-03]

Message: The file <*file name*> does not exist!

Cause: The specified path and/or file name for the import does not exist.

Action: Enter the correct file name and/or path of an existing file.

[afilemgt-04]

Message: An error has occurred when importing the file "<*file name*>" to the database.

    The import will be aborted.

Cause: During the import, a database or a network error has occurred.

Action: If a database error message has also previously suddenly been shown, note the error information and contact a system administrator.

    Restart the application and repeat the process.

[afilemgt-05]

Message: An error has occurred during the export of the file "<*file name*>" from the database.

    The export will be aborted.

Cause: During the export, a database or a network error has occurred.

Action: If a database error message has also previously suddenly been shown, note the error information and contact a system administrator.

    Restart the application.

[afilemgt-06]

Message: An error has occurred when renaming the file "<*file name*>".

    The file has not been renamed.

Cause: When renaming the file, a database or network error has occurred.

Action: If a database error message has also previously suddenly been shown, note the error information and contact a system administrator.

    Restart the application and repeat the process.

[agdt-01]

Message: An error has occurred when sorting the start model!

Cause: The model could not be sorted. Connectors are possibly missing in the model to be arranged.

Action: Check the model to be arranged for missing connectors and repeat the process. Should the error appear again, close ADONIS and start it again. If the error is not corrected, contact your ADONIS administrator.

[agdt-02]

Message: Positions could not be assigned!

Cause: An error occurred, while positions were being assigned

Action: Quit ADONIS and start it again. Should this error re-occur, contact your ADONIS administrator.

[agdt-03]

Message: Bendpoints could not be inserted!

Cause: An error occurred while the bendpoints were being inserted.

Action: Should this error re-occur, align the connectors without inserting bendpoints or contact your ADONIS administrator.

[agdt-04]

Message: Positions could not be assigned with bendpoints!

Cause: An error occurred while positions with bendpoints were being assigned.

Action: Quit ADONIS and start it again. Should this error re-occur, contact your ADONIS administrator.

[agdt-05]

Message: Method median-down could not be applied!

Cause: An error occurred while the median-down method was being applied.

Action: Quit ADONIS and start it again. Should this error re-occur, align the connectors without applying the median-down method or contact your ADONIS administrator.

[agdt-06]

Message: Median-up method could not be applied!

Cause: An error occurred while the median-up method was being applied.

Action: Quit ADONIS and start it again. Should this error re-occur, align the connectors without applying the median-up method or contact your ADONIS administrator.

[agdt-07]

Message: Pairwise exchange could not be applied!

Cause: An error occurred while pair wise exchange was being applied.

Action: Quit ADONIS and start it again. Should this error re-occur, align the connectors without applying pairwise exchange or contact your ADONIS administrator.

[agdt-08]

Message: Pendulum-down method could not be applied.

Cause: An error occurred while the pendulum-down method was being applied.

Action: Quit ADONIS and restart it. Should this error re-occur, align the connectors without applying the pendulum-down method or contact your ADONIS administrator.

[agdt-09]

Message: Pendulum-up method could not be applied!

Cause: An error occurred while the pendulum-up method was being applied.

Action: Quit ADONIS and start it again. Should this error re-occur, align the connectors without applying the pendulum-up method or contact your ADONIS administrator.

[agdt-10]

Message: Horizontal mirror could not be applied!

Cause: An error occurred while applying the horizontal mirror.

Action: Quit ADONIS and start it again. Should this error re-occur, contact your ADONIS administrator.

[agdt-11]

Message: Vertical mirror could not be applied!

Cause: An error occurred while applying the vertical mirror.

Action: Quit and restart ADONIS. Should this error re-occur, contact your ADONIS administrator.

[agdt-12]

Message: It was not possible to arrange downwards!

Cause: An error occurred while arranging downwards.

Action: Quit and restart ADONIS. Should this error re-occur, contact your ADONIS administrator.

[agdt-13]

Message: It was not possible to arrange upwards!

Cause: An error occurred while arranging downwards.

Action: Quit and restart ADONIS. Should this error re-occur, contact your ADONIS administrator.

[agdt-14]

Message: It was not possible to arrange to the right!

Cause: An error occurred while arranging to the right.

Action: Quit and restart ADONIS. Should this error re-occur, contact your ADONIS administrator.

[agdt-15]

Message: It was not possible to arrange to the left!

Cause: An error occurred while arranging to the left.

Action: Quit ADONIS and start it again, contact your ADONIS administrator.

[agdt-16]

Message: The rubber band method could not be applied!

Cause: An error occurred while applying the rubber band method.

Action: Quit and restart ADONIS. Should this error re-occur, contact your ADONIS administrator.

[agdt-18]

Message: Cycle minimisation could not be applied!

Cause: An error occurred while applying cycle minimisation.

Action: The model contains too many cycles and therefore may not be arranged in the current form. Try to simplify the model and to reduce the number of cycles.

[agdt-19]

Message: New model could not be written!

Cause: An error occurred while the newly arranged model was about to be transferred.

Action: Quit and restart ADONIS. Should this error re-occur, contact your ADONIS administrator.

[agdt-20]

Message: At least one relation couldn't be inverted according to the parameter value!

Cause: An error occured while attempting to turn the relations of the model according to the parameter value.

Action: Change the parameter values of the alignment function or contact your ADONIS administrator.

[agdt-22]

Message: Process-hierarchy could not be generated because two objects would have the same name!

Cause: An attempt was made to create an object with a name currently assigned to an existing object.

Action: Delete the model, into which the process hierarchy was to be generated and then start generating the process hierarchy again.

[agdt-23]

Message: Could not position one object of the model to be arranged!

Cause: An error occurred while positioning the object.

Action: Quit and restart ADONIS. Should this error re-occur, contact your ADONIS administrator.

[agdt-24]

Message: Could not number the objects of the model!

Cause: An error occurred, while the model objects were being numbered.

Action: Quit and restart ADONIS. Should this error re-occur, contact your ADONIS administrator.

[agdt-26]

Message: The selected model contains no model object to be arranged! No arrangement could be carried out.

Cause: The arrangement function has been used on a model which contains no object and therefore has been closed.

Action: Use the arrangement function only on models with objects you want to re-place.

[agdt-27]

Message: The selected model contains no model references to be displayed as a process-hierarchy!

Cause: You selected a model without model references for generating a graphical process hierarchy.

Action: Insert the necessary model references or select a different model.

[agdt-28]

Message: The selected model involves no objects for numbering! Cannot number objects.

Cause: You selected a model which does not contain any objects or objects which are not defined for numbering when attempting to number the objects.

Action: Model the object to be numbered or select a different model.

[agdt-29]

Message: Name missing after the key word "*<profile>*"!

Cause: In the configuration of the arrangement function, a profile and a name have been defined. The profile is either an arrangement profile **PROFILE**, a hierarchy profile **HIERPROFILE** or an enumeration profile **ENUMPROFILE**.

Action: Once you have specified a profile in the configuration of the arrangement function, you need to give a name to the profile. Specify the name between double quotation marks or if necessary contact your ADONIS administrator.

[agdt-30]

Message: Faulty initialising with expression for the key word "*<element>*"!

Cause: The specified element could not be initialised. The error appears, when an element of the same name has already been defined, when a previous definition refers to the same model type/the same class, relation etc. or when the specified model type/the specified class/relation class does not exist.

Element is one of the following syntax elements:

PROFILE, **CLASSPAIRPAR**, **HIERPROFILE**, **HIERmodel type**, **ENUMPROFILE**, **ENUMmodel type**, **ENUMCLASS** or **ENUMREL**.

Action: Verify the configuration of the arrangement function and if necessary contact your ADONIS administrator.

[agdt-31]

Message: Wrong key word "*<key word>*"!

Cause: The specified key word is not authorised in this context or makes no sense here.

Action: Verify the configuration of the arrangement function and if necessary contact your ADONIS administrator.

[agdt-32]

Message: Faulty value after the key word "*<key word>*"!

Cause:	This error appears for one of the following reasons:
	The name of the model type has not been specified or is invalid (for **HIERmodel type**, **ENUMmodel type**, **CLASSmodel type**, **DEFmodel type** or **CLASSPAIRmodel type**).
	The name of the attribute, the class or the relation is missing or is invalid (for **CLASSPAR**, **CLASSPAIRPAR**, **HIERATTRIB**, **HIERCLASS**, **HIERUSECLASS**, **HIERUSEREL**, **HIERUSEATTRIB**, **ENUMCLASS** or **ENUMREL**).
	The parameter **dist:** for **DOUBLEDP** has not been specified.
	The parameter **vertdist:** for **CHNGSIZE** has not been specified.
	The parameter **use:** for **HIERPROFILE** or **usetype:** for **HIERmodel type** has not been specified, the profile or the model type does not exist.
Action:	Check/complete the configuration of the application function and if necessary contact your ADONIS administrator.

[agogo-01]

Message:	Error in the value of attribute "*<attribute name>*" (*<context>*):
	The keyword "GRAPHREP" must appear first but the attribute value begins with "*<Text>*"!
Cause:	The specified attribute contains errors.
Action:	This error can be fixed in the ADONIS Administration Toolkit or please contact your ADONIS administrator.

[agogo-02]

Message:	Error in the value of attribute "*<attribute name>*" (*<context>*):
	"*<Text>*" is not a keyword!
Cause:	The value assigned to the specified attribute during customising contains errors.
Action:	This error can be fixed in the ADONIS Administration Toolkit or please contact your ADONIS administrator.

[agogo-03]

Message:	Error in the value of attribute "*<attribute name>*" (*<context>*):
	"*<Text>*" is not the name of an attribute!
Cause:	The value assigned to the specified attribute during customising contains errors.
Action:	This error can be fixed in the ADONIS Administration Toolkit or please contact your ADONIS administrator.

[agogo-04]

Message:	Error in the value of attribute "*<attribute name>*" (*<context>*):
	<Keyword> cannot be used together with COMPOUND!

Cause: The value defined during customising of the attribute listed contains the syntax error specified.

Only LINE, POLYLINE and CURVE can be used together with COMPOUND.

Action: Correct the error in the Library Management of the ADONIS administration Toolkit or contact your ADONIS administrator.

[agogo-05]

Message: Error in the value of attribute "*<attribute name>*" (*<context>*):

Illegal numeric parameter in *<keyword>* element!

An integer value > 0 has to be specified.

Cause: The value defined during customising of the attribute listed contains the syntax error specified.

The number of points in the case of POLYLINE / POLYGON must be greater than 0.

Action: Correct the error in the Library Management of the ADONIS administration Toolkit or contact your ADONIS administrator.

[agogo-06]

Message: Error in the value of attribute "*<attribute name>*" (*<context>*):

*<Keyword>*without IF!

Cause: The value defined during customising of the attribute listed contains the syntax error specified.

ELSIF, ELSE or ENDIF used without previous IF.

Action: Correct the error in the Library Management of the ADONIS administration Toolkit or contact your ADONIS administrator.

[agogo-07]

Message: Error in the value of attribute "*<attribute name>*" (*<context>*):

ENDIF is missing!

Cause: The value defined during customising of the attribute listed contains the syntax error specified.

Action: Correct the error in the Library Management of the ADONIS administration Toolkit or contact your ADONIS administrator.

[agogo-08]

Message: Error in the value of attribute "*<attribute name>*" (*<context>*):

<keyword> is just defined for relations!

Cause: The value defined during customising of the attribute listed contains the syntax error specified.

EDGE, START, MIDDLE or END in the definition of the graphic representation of a class.

Action: Correct the error in the Library Management of the ADONIS administration Toolkit or contact your ADONIS administrator.

[agogo-10]

Message: Error in the value of attribute "*<attribute name>*" (*<context>*):

IF with multiple ELSE branches!

Cause: The customised attribute value contains the syntax/semantics error described.

A defined IF sequence contains more than one ELSE branches. While multiple ELSIF statements are possible, only one ELSE condition is allowed.

Action: Fix the error in the Library Management of the ADONIS Administration Toolkit or contact your ADONIS administrator.

[agogo-11]

Message: Error with the value of the attribute "*<attribute name>*" (*<context>*):

<Keyword> without block - element sequence enclosed in curly braces is missing!

Cause: The customised attribute value contains the syntax/semantics error described.

FOR or WHILE loops without element sequence in curly braces ("**{ .. }**").

Action: Fix the error in the Library Management of the ADONIS Administration Toolkit or contact your ADONIS administrator.

[agogo-12]

Message: Error with the value of the attribute "*<attribute name>*" (*<context>*):

ENDWHILE without WHILE!

Cause: The customised attribute value contains the syntax/semantics error described.

ENDWHILE was used without preceeding **WHILE** statement.

Action: Fix the error in the Library Management of the ADONIS Administration Toolkit or contact your ADONIS administrator.

[agogo-13]

Message: Error with the value of the attribute "*<attribute name>*" (*<context>*):

"*<Name>*" is not the name of a model type group!

Cause: The customised attribute value contains the syntax/semantics error described.

The model type group name specified in the HOTSPOT definition of the Method Diagram does not comply with any existing name of a model type group.

Action: Fix the error in the Library Management of the ADONIS Administration Toolkit or contact your ADONIS administrator.

[agolay-03]

Message: Error in the value of attribute "*<attribute name>*" (*<context>*):

"*<text>*" is not a keyword!

Cause: The value defined during customising of the attribute listed contains the syntax error specified.

Action: Correct the error in the Library Management of the ADONIS Administration Toolkit or contact your ADONIS administrator.

[agolay-04]

Message: Error in the value of the attribute "*<attribute Name>*" (*<context>*):

Missing layout for "*<keyword>*"!

Cause: The value defined during customising of the attribute listed contains the syntax error specified.

HEAD, BITMAP, PAGE or FOOT, without having previously introduced LAYOUT.

Action: Correct the error in the Library Management of the ADONIS Administration Toolkit or contact your ADONIS administrator.

[agolay-05]

Message: Error in the value of attribute "*<attribute name>*" (*<context>*):

Region "*<keyword>*" defined twice!

Cause: The value defined during customising of the attribute listed contains the syntax error specified.

HEAD, BITMAP, PAGE or FOOT definition twice in LAYOUT.

Action: Correct the error in the Library Management of the ADONIS Administration Toolkit or contact your ADONIS administrator.

[agolay-06]

Message: Error in the value of the attribute "*<attribute name>*" (*<context>*):

Missing region for "*<keyword>*"!

Cause: The value defined during customising of the attribute listed contains the syntax error specified.

TEXT, ATTR or FONT, without having previously introduced a region (HEAD, ...).

Action: Correct the error in the Library Management of the ADONIS Administration Toolkit or contact your ADONIS administrator.

[agolay-07]

Message: Error in the value of the attribute "*<attribute name>*" (*<context>*):

Necessary region "PAGE" is missing in layout "*<Name>*"!

Cause:	The value defined during customising of the attribute listed contains the syntax error specified.
	A layout must always contain a PAGE definition.
Action:	Correct the error in the Library Management of the ADONIS Administration Toolkit or contact your ADONIS administrator.

[agolay-08]

Message:	Error in the value of the attribute "*<attribute name>*" (*<context>*):
	At layout "*<name of layout>*" specified attribute "*<attribute name>*" is not existing or is of the wrong type!
Cause:	The customised attribute value contains the syntax/semantics error described.
	In the page layout definition *<name of layout>* the GraphRep attribute *<attribute name>* was specified. However, this attribute does not exist in the class "__ModelTypeMetaData__".
Action:	Fix the error in the Library Management of the ADONIS Administration Toolkit or contact your ADONIS administrator.

[ahaconv-01]

Message:	"*<Token ()>*" is not a key word!
Cause:	There is a syntactical error in a class attribute "__Conversion__", which is required for the transformation of objects to another class.
Action:	Verify and correct the "__Conversion__" attribute.

[ahaconv-02]

Message:	"*<Name>*" is not the name of class that can be instanced!
Cause:	A not existing class is specified in a class attribute "__Conversion__", which is required for the transformation of objects to another class.
Action:	Verify and correct the "__Conversion__" attribute.

[ahaconv-03]

Message:	The attribute "*<attribute name>*" (*<class name>*) does not exist!
Cause:	A not existing attribute is specified in a class attribute "__Conversion__", which is required for the transformation of objects to another class.
Action:	Verify and correct the "__Conversion__" attribute.

[ahaconv-04]

Message:	The attribute "*<attribute name 1>*" (*<class name 1>*) and "*<attribute name 2>*" (*<class name 2>*) are not of the same type!

Cause: A conversion indication, which is not realisable, is specified in a class attribute "__Conversion__", which is required for the transformation of objects to another class.

Action: Verify and correct the "__Conversion__" attribute.

[ahaconv-05]

Message: The attributes are not based on the same record class:

"*<Attribute name 1>*" (*<Class name 1>*): "*<Record class 1>*"

"*<Attribute name 2>*" (*<Class name 2>*): "*<Record class 2>*"

Cause: In the class attribute "__Conversion__", used for converting objects from one class to another, a statement has been made which cannot be executed.

Action: Check and correct the "__Conversion__" attribute.

[ahaconv-06]

Message: The attributes are not based on the same attribute profile class:

"*<Attribute name 1>*" (*<Class name 1>*): "*<Attribute profile class 1>*"

"*<Attribute name 2>*" (*<Class name 2>*): "*<Attribute profile class 2>*"

Cause: In the class attribute "__Conversion__", used for converting objects from one class to another, a statement has been made which cannot be executed.

Action: Check and correct the "__Conversion__" attribute.

[ahaconv-07]

Message: The following attribute cannot be converted as not all enumeration values are defined in the target attribute:

From "*<Attribute name 1>*" (*<Class name 1>*) to "*<Attribute name 2>*" (*<Class name 2>*)

Cause: In the class attribute "__Conversion__", used for converting objects from one class to another, a statement has been made which cannot be executed.

Action: Check and correct the "__Conversion__" attribute.

[ahaconv-08]

Message: The following attribute cannot be converted as the name of the target attribute is already assigned to another object:

From "*<Attribute name 1>*" (*<Class name 1>*) to "*<Attribute name 2>*" (*<Class name 2>*)

Cause: In the class attribute "__Conversion__", used for converting objects from one class to another, a statement has been made which cannot be executed. The name used for the instance to be converted cannot be assigned as this name has already been assigned to an object of this class in the model.

Action: Check and correct the "__Conversion__" attribute.

[ahagrob-01]

Message: "*<text>*" is not a keyword!

Cause: In the class attribute "Allowed objects" of a swimlane class an unknown keyword is defined.

Action: Correct the error in the Library Management of the ADONIS Administration Toolkit or contact your ADONIS administrator.

[ahagrob-02]

Message: "*<class name>*" is not the name of a class!

Cause: In the class attribute "Allowed objects" of a swimlane class the name of a undefined class is used with the INCL or EXCL keyword.

Action: Correct the error in the Library Management of the ADONIS Administration Toolkit or contact your ADONIS administrator.

[ahagrob-03]

Message: Objects of class "*<class name>*" may be placed over multiple swimlanes.

Therefore class "*<class name>*" cannot be excluded from swimlane class "*<class name of swimlane>*"!

Cause: In the class attribute "Allowed objects" of the specified swimlane class the specified class is excluded although this class can be placed above several swimlanes. Because of the invalid combination (exclusion in the swimlane class - class allowed to be placed above several swimlanes) Klasse in mehreren Schwimmbahnen erlaubt) the exclusion is ignored.

Action: Correct the class attribute "Allowed objects" in the Library Management of the ADONIS Administration Toolkit or contact your ADONIS administrator.

[ahamot-01]

Message: Error in the value of attribute "*<attribute name>*" (*<context>*):

"*<text>*" is not a keyword!

Cause: The value defined during customising of the attribute listed contains the syntax error specified.

Action: Correct the error in the Library Management of the ADONIS Administration Toolkit or contact your ADONIS administrator.

[ahamot-02]

Message: Error in the value of attribute "*<attribute name>*" (*<context>*):

Missing model type for "*<keyword>*"!

Cause: The value defined during customising of the attribute listed contains the syntax error specified.

INCL, **EXCL**, **OR_ASSIGN**, **AND_ASSIGN** or **MODE**, without having previously introduced model type.

Action: Correct the error in the Library Management of the ADONIS Administration Toolkit or contact your ADONIS administrator.

[ahamot-03]

Message: Error in the value of attribute "*<attribute name>*" (*<context>*):

Name for "*<keyword>*" is missing!

Cause: The value defined during customising of the attribute listed contains the syntax error specified.

Name missing after **model type**, **MODE**, **OR_ASSIGN** or **AND_ASSIGN**.

Action: Correct the error in the Library Management of the ADONIS Administration Toolkit or contact your ADONIS administrator.

[ahamot-04]

Message: Error in the value of attribute "*<attribute name>*" (*<context>*):

Base model type "*<name>*" is not defined!

Cause: The value defined during customising of the attribute listed contains the syntax error specified.

The model type (name after **from:**, **OR_ASSIGN** or **AND_ASSIGN**) was not defined.

Action: Correct the error in the Library Management of the ADONIS Administration Toolkit or contact your ADONIS administrator.

[ahamot-05]

Message: Error in the value of attribute "*<attribute name>*" (*<context>*):

Base mode "*<name>*" is not defined!

Cause: The value defined during customising of the attribute listed contains the syntax error specified.

The mode (name after **from:**, **OR_ASSIGN** or **AND_ASSIGN**) was not defined.

Action: Correct the error in the Library Management of the ADONIS Administration Toolkit or contact your ADONIS administrator.

[ahamot-06]

Message: Error in the value of attribute "*<attribute name>*" (*<context>*):

"*<name>*" is not the name of a class or relation class that can be instanced!

Cause: The value defined during customising of the attribute listed contains the syntax error specified.

The name after **INCL** or **EXCL** is not a class or relation name.

Common mistake: Trying to include or exclude a class which does not belong to the model type at all, while a mode is defined.

Action: Correct the error in the Library Management of the ADONIS Administration Toolkit or contact your ADONIS administrator.

[ahamot-08]

Message: The file "*<file name>*" could not be found!

Cause: The Bitmap file assigned to a model type does not exist.

Action: Create the appropriate file or correct the file name.

[ahanote-01]

Message: Error in the value of attribute "*<attribute name>*" (*<context>*):

"*<text>*" is not a keyword!

Cause: The value defined during customising of the attribute listed contains the syntax error specified.

Action: Correct the error in the Library Management of the ADONIS Administration Toolkit or contact your ADONIS administrator.

[ahanote-02]

Message: "*<attribute name>*" is not the name of an attribute of this class!

Cause: The specified attribute is contained in the class attribute "AttrRep", but it is not defined for this class.

Action: Enter in the class attribute "AttrRep" only the attribute, which is also defined in this class.

[ahanote-04]

Message: The attribute "*<attribute name>*" is contained several times!

Cause: In the class attribute "AttrRep" for the assembly of the ADONIS notebook, the mentioned attribute is defined more than once. In the ADONIS notebook, an attribute cannot be displayed more than once.

Action: Delete in the class attribute "AttrRep" the concerned ATTR elements, so that the specified attribute appears only once.

[ahanote-05]

Message: The user "*<User name>*" has no access rights for the notebook!

Cause: You have not sufficient rights to access the ADONIS notebook of the object/relation.

Action: Contact your ADONIS administrator.

[aicondef-01]

Message: Cannot load icon configuration!

Cause: Possibly the connection to the ADONIS database server has been aborted.

Action: Repeat the action. Should the error re-occur, please contact your database or system administrator.

[aicondef-02]

Message: Cannot save icon configuration!

Cause: Possibly the connection to the ADONIS database server has been aborted.

Action: Repeat the action. Should the error re-occur, please contact your database or system administrator.

[aicondef-03]

Message: Unknown user!

Cause: Possibly the connection to the ADONIS database server has been aborted.

Action: Repeat the action. Should the error re-occur, please contact your database or system administrator.

[aicondef-04]

Message: Cannot load quick-access bar configuration from application library!

Cause: Possibly the application library is defective.

Action: Repeat the action. Should the error re-occur, please contact your ADONIS administrator.

[aimport-01]

Message: Unable to open import file.

Cause: The import file could not be opened; the data medium (e.g. floppy disk) could not be accessed.

Action: Check the medium and re-start the import.

[aimport-02]

Message: Import of file *<path+file name>*:

error in line *<#*:

<semantic/syntax error>

The import process will be aborted.

Cause: During the import, the specified file contains a semantically or syntactically incorrect definition. This might be due to a default in the specified file or to manual changes within the file.

Action: The content of the file cannot be imported. If necessary contact your ADONIS consultant.

[aimport-03]

Message: Import of file <*path+file name*>: contains libraries which are not allowed.

Cause: The ADL-file contains library definitions, which are not allowed.

Action: The models included in the file cannot be imported.

[aimport-04]

Message: A serious error occurred while deleting the erroneous library. Possibly the connection to the database has been aborted. Please exit ADONIS.

Cause: An error occurred during deletion of an erroneous library.

Action: Exit ADONIS and inform your database administrator.

[aimport-05]

Message: A serious error occurred while deleting the erroneous model. Possibly the connection to the database has been aborted. Please exit ADONIS.

Cause: A serious error occurred during the deletion of an erroneous model.

Action: Exit ADONIS, re-start it and redo the import. If the error re-occurs, please contact your database administrator.

[aimport-06]

Message: A serious error occurred while loading the old model. Possibly the connection to the database has been aborted. Please exit ADONIS.

Cause: After an error occurred during the model import (option "Paste into existing models"), the attempt to reload the old model led to a serious error.

Action: Exit ADONIS, re-start it and redo the import. If the error re-occurs, please contact your database administrator.

[aimport-07]

Message: A serious error occurred while restoring the old model. Possibly the connection to the database has been aborted. Please exit ADONIS.

Cause: After an error occurred during the model import (option "Overwrite existing models"), the attempt to restore the model to be overwritten led to a serious error.

Action: Exit ADONIS, re-start it and redo the import. If the error re-occurs, please contact your database administrator.

[aimport-08]

Message: Import of file<*file name*>: File contains models which are not allowed.

Cause: The ABL file contains model definitions, which are not allowed.

Action: The libraries included in the file cannot be imported.

[aimport-09]

Message: Import of file <*file name*>: invalid format.

Cause: You tried to import an ADONIS library or an ADL file which is not compatible with your ADONIS version. ADONIS libraries are not directly exchangeable within different ADONIS versions and/or different operating systems.

Action: Please contact your ADONIS consultant.

[aimport-10]

Message: Import of file <*file name*>:

The read-only model <*model name*> cannot be changed by import.

Cause: You tried to change a read-only model via model import (option "Overwrite existing models" or "Paste into existing models????").

Action: If the model was loaded read-only, please close it and import the model again. If you have read-only access for the model, please contact your ADONIS administrator.

[aimport-11]

Message: No application libraries are defined in the database. Therefore user import is impossible.

Cause: You tried to import users, without previously importing an application library.

Action: Import the application library needed and re-start the user import.

[aimport-12]

Message: Error when opening the protocol file!

Cause: An error has occurred when opening the protocol file. The data medium (e.g. floppy disk) is possibly write-protected.

Action: Specify another protocol file name and/or control the write-protection of the data medium.

[aimport-13]

Message: Error when writing the protocol file "*<file name>*"!

Cause: When creating the import protocol file an error has occurred.

Action: The data medium is possibly full.

[aimport-14]

Message: The model group "*<model group name>*" does not exist!

 Would you like to skip the setting of the corresponding access right and continue the import?

Cause: The model group referenced for the UDL import does not exist, i.e. in the UDL file, the access right defined to the specified model group cannot be set.

Action: Click on the "Yes" button to skip the setting of the access rights of the specified model group and continue the import.

 Click on the "All" button to skip the setting of the access rights of all non existing model groups and continue the UDL import.

 If you click on the "No" button, the UDL import will be aborted.

[aimport-15]

Message: The libraries contain no classes! An import is therefore not possible.

Cause: The libraries of the imported ABL file contain no classes.

Action: Select another ABL file.

[aimport-16]

Message: Import of the file*<file name>*:

 The user has no access to the model "*<model name>*"!

 This cannot be changed by the import.

Cause: You have attempted to change a model you can not access by ADL import into the data base. This is not permitted.

Action: Select another import option (e.g. rename) or do not import the specified model.

[aimport-17]

Message: Unknown target library!

 The connection to the database is possibly aborted.

 Exit ADONIS.

Cause: When searching for information in the library, into which the ADL file should be imported, a serious error has occurred.

Action: Exit ADONIS, restart it and repeat the import. Should the message re-appear, contact your database administrator.

[aimport-18]

Message: The ADL file is possibly from a versioned library.

The import of such a file to a non-versioned library is only possible by explicitly specifying the option 'import-versioned-file'.

Cause: You have tried to import an ADL file from a versioned library to a non versioned library. In the silent mode of the ADL import via an AdoScript, this is only possible by explicitly specifying the option 'import-versioned-file'.

Action: Insert the option 'import-versioned-file' to the AdoScript.

[aimport-19]

Message: The ADL file is possibly from a versioned library.

The import of such a file with the option 'Transfer version number to the model name' is only possible by explicitly specifying the option 'import-versioned-file'.

Cause: When searching for information of the library to which the ADL file should be imported, a serious error has occurred.

Action: Insert the additional option 'import-versioned-file' to the AdoScript.

[aimport-20]

Message: Invalid target model group!

Cause: A target model group must be specified for the ADL import via an AdoScript in silent mode, when models or model groups should be imported. You have specified either no or invalid target model groups.

Action: Specify a valid target model group.

[aimport-22]

Message: Invalid target attribute profile group!

Cause: A target attribute profile group must be specified for the ADL import via an AdoScript in silent mode, when attribute profile groups should be imported. You have specified either no or an invalid target attribute profile group.

Action: Enter a valid target attribute profile group.

[aimport-24]

Message: A serious error has occurred when re-creating the old attribute profile.

The connection with the database is possibly aborted.

Exit ADONIS.

Cause: After an error in the attribute profile import (option "Overwrite attribute profiles") you have tried to re-create the overwritten attribute profile, which has led to a serious error.

Action: Exit ADONIS, restart it and repeat the import. If the message appears again, inform your database administrator.

[aimport-25]

Message: A serious error has occurred when deleting the faulty attribute profile.

The connection with the database is possibly aborted.

Exit ADONIS.

Cause: After an error in the attribute profile import, you have tried to delete the faulty attribute profile, which has led to a serious error.

Action: Exit ADONIS, restart it and repeat the import. If the message appears again, inform your database administrator.

[aimport-26]

Message: Import of the file *<ABL file name>*:

Error when saving the file belonging to the library "*<file name>*"!

Cause: A serious error has occurred during the library import when saving the specified file to the ADONIS database.

Action: Exit ADONIS, restart it and repeat the import. Note that the length of the external file name within the library (including the file extension) and the length of the library name must not exceed 249 characters in total.

If the message appears again, inform your database administrator.

[aimport-27]

Message: System user administration could not be initialised!

Cause: You have been trying to import system users into an ADONIS database with Single-Sign-on functionality, but the operating system (Windows ME/98/95) does not support Single-Sign-on.

Action: To import system users into a ADONIS database use an operating system supporting Single-Sign-on (i.e. Windows XP/2000/NT 4.0).

Please pay attention to the software pre-requisites for Single-Sign-on within the ADONIS installation manual.

[airdom-01]

Message: The LEO expression is faulty:

"*<LEO error message>*"

Cause: The LEO expression in the facet "AttributeInterRefDomain" does not correspond with the LEO syntax. Pay attention to the LEO error message.

Action:	Rectify the value of the facet "AttributeInterRefDomain" by adapting theLEO expression or contact your ADONIS administrator.

[airdom-02]

Message:	The LEO expression must contain at last one domain definition.
Cause:	The LEO expression contains no domain definition. At least one domain definition is always required.
Action:	Rectify the value of the facet "AttributeInterRefDomain", by adapting the LEO expression or contact your ADONIS administrator.

[airdom-03]

Message:	Only "MODREF" and "OBJREF" are valid keywords.
Cause:	An invalid key word has been specified. Valid key words are "MODREF" and "OBJREF".
Action:	Rectify the value of the facet "AttributeInterRefDomain" by adapting the LEO expression, so that it contains only the key words "MODREF" and "OBJREF" or contact your ADONIS administrator.

[airdom-04]

Message:	Model references and object references cannot be mixed.
Cause:	The key words "MODREF" and "OBJREF" cannot be used mixed within an attribute.
Action:	Rectify the value of the facet "AttributeInterRefDomain" by adapting the LEO expression, so that the key words "MODREF" and "OBJREF" don't appear together or contact your ADONIS administrator.

[airdom-05]

Message:	Internal core error.
Cause:	An internal error has occured.
Action:	Contact your ADONIS administrator.

[airdom-06]

Message:	The specified class does not exist.
Cause:	The specified class name is not valid in the library.
Action:	Rectify the value of the facet "AttributeInterRefDomain" by adapting the LEO expression so that it contains a valid class name or contact your ADONIS administrator.

[airdom-07]

Message:	The specified model type does not exist.
Cause:	The specified model type is unknown in the current library.
Action:	Rectify the value of the facet "AttributeInterRefDomain" by adapting the LEO expression, so that it contains a valid model type name or contact your ADONIS administrator.

[airdom-08]

Message:	The specified class does not exist in the specified model type.
Cause:	The specified class does exist, but is not visible in the specified model type.
Action:	Rectify the value of the facet "AttributeInterRefDomain" by adapting the LEO expression, so that it contains a valid class name or contact your ADONIS administrator.

[airdom-09]

Message:	The maximum number of referenced is not valid.
Cause:	The parameter "max" contains an invalid value. Valid values must be greater than 0.
Action:	Rectify the value of the facet "AttributeInterRefDomain" by adapting the LEO expression or contact your ADONIS administrator.

[aleo-01]

Message:	Character expected after backslash '\'!
	<context>
Cause:	The attribute listed (context) was changed during customising (or in the ADL file) and contains the syntax error described.
Action:	Correct the error described or contact your ADONIS administrator.

[aleo-02]

Message:	Closing quotation mark " is missing!
	<context>
Cause:	The attribute listed (context) was changed during customising (or in the ADL file) and contains the syntax error described.
Action:	Correct the error described or contact your ADONIS administrator.

[aleo-03]

Message:	Invalid character: *<character>*.
	key word (next element) or Name (next attribute) expected!

<Context>

Cause: The attribute listed (context) was changed during customising (or in the ADL file) and contains the syntax error described.

Action: Correct the error described or contact your ADONIS administrator.

[aleo-04]

Message: *<figure>* ':' expected!

<context>

Cause: The attribute listed (context) was changed during customising (or in the ADL file) and contains the syntax error described.

Action: Correct the error described or contact your ADONIS administrator.

[aleo-05]

Message: Value expected after ':' !

<context>

Cause: The attribute listed (context) was changed during customising (or in the ADL file) and contains the syntax error described.

Action: Correct the error described or contact your ADONIS administrator.

[aleo-06]

Message: *<Text>*: Measure unknown!

<context>

Cause: The attribute listed (context) was changed during customising (or in the ADL file) and contains the syntax error described.

Action: Correct the error described or contact your ADONIS administrator.

[aleo-08]

Message: *<Text>*: key word expected!

<context>

Cause: The attribute listed (context) was changed during customising (or in the ADL file) and contains the syntax error described.

Action: Correct the error described or contact your ADONIS administrator.

[aleo-09]

Message: Unauthorised figure after closing quotation marks '"'!

<context>

Cause: The attribute listed (context) was changed during customising (or in the ADL file) and contains the syntax error described

Action: Correct the error described or contact your ADONIS administrator.

[aleo-10]

Message: Unauthorised figure after a bracket closing an expression ')'!

<context>

Cause: The attribute listed (context) was changed during customising (or in the ADL file) and contains the syntax error described

Action: Correct the error described or contact your ADONIS administrator.

[aleo-13]

Message: Operand expected!

<context>

Cause: The attribute listed (context) was changed during customising (or in the ADL file) and contains the syntax error described

Action: Correct the error described or contact your ADONIS administrator.

[aleo-14]

Message: Closing bracket ')' is missing!

<context>

Cause: The attribute listed (context) was changed during customising (or in the ADL file) and contains the syntax error described

Action: Correct the error described or contact your ADONIS administrator.

[aleo-15]

Message: Operator expected!

<context>

Cause: The attribute listed (context) was changed during customising (or in the ADL file) and contains the syntax error described

Action: Correct the error described or contact your ADONIS administrator.

[aleo-16]

Message: Closing bracket ')' without opening bracket '('!

<context>

Cause:	The attribute listed (context) was changed during customising (or in the ADL file) and contains the syntax error described
Action:	Correct the error described or contact your ADONIS administrator.

[aleo-17]

Message:	Unauthorised figure!
	<context>
Cause:	The attribute listed (context) was changed during customising (or in the ADL file) and contains the syntax error described
Action:	Correct the error described or contact your ADONIS administrator.

[aleo-18]

Message:	Division by 0!
	<context>
Cause:	The described runtime error has occurred during the evaluation of an expression, which is contained in the value of the specified attribute.
Action:	Correct the error described or contact your ADONIS administrator.

[aleo-19]

Message:	Over flow!
	<context>
Cause:	The described runtime error has occurred during the evaluation of an expression, which is contained in the value of the specified attribute.
Action:	Correct the error described or contact your ADONIS administrator.

[aleo-20]

Message:	This expression shall not contain a variable!
	<context>
Cause:	The described runtime error has occurred during the evaluation of an expression, which is contained in the value of the specified attribute.
Action:	Correct the error described or contact your ADONIS administrator.

[aleo-21]

Message:	Type incompatibility!
	<function call>
	<context>

Cause: The described runtime error has occurred during the evaluation of an expression, which is contained in the value of the specified attribute.

Action: Correct the error described or contact your ADONIS administrator.

[aleo-22]

Message: <*Text*>: Unknown function!

<*context*>

Cause: The described runtime error has occurred during the evaluation of an expression, which is contained in the value of the specified attribute.

Action: Correct the error described or contact your ADONIS administrator.

[aleo-23]

Message: Wrong parameter number!

<*context*>

Cause: The described runtime error has occurred during the evaluation of an expression, which is contained in the value of the specified attribute.

Action: Correct the error described or contact your ADONIS administrator.

[aleo-24]

Message: <*Text*>: Unknown operator!

<*context*>

Cause: The described runtime error has occurred during the evaluation of an expression, which is contained in the value of the specified attribute.

Action: Correct the error described or contact your ADONIS administrator.

[aleo-25]

Message: <*Text*>: Lowercase letters in a key word!

<*context*>

Cause: The described runtime error has occurred during the evaluation of an expression, which is contained in the value of the specified attribute.

Action: Correct the error described or contact your ADONIS administrator.

[aleo-26]

Message: Under flow when accessing to string!

<*context*>

Cause:	The described runtime error has occurred during the evaluation of an expression, which is contained in the value of the specified attribute.
Action:	Correct the error described or contact your ADONIS administrator.

[aleo-27]

Message:	Over flow when accessing to string!
	<context>
Cause:	The described runtime error has occurred during the evaluation of an expression, which is contained in the value of the specified attribute.
Action:	Correct the error described or contact your ADONIS administrator.

[aleo-28]

Message:	*<figure>*: unauthorised figure after a key word!
	<context>
Cause:	The attribute listed (context) was changed during customisation (or in the ADL file) and contains the syntax error described.
Action:	Correct the error described or contact your ADONIS administrator.

[aleo-29]

Message:	Unauthorised figure after an attribute name!
	<context>
Cause:	The attribute listed (context) was changed during customisation (or in the ADL file) and contains the syntax error described.
Action:	Correct the error described or contact your ADONIS administrator.

[aleo-30]

Message:	Unauthorised figure after an attribute value!
	<context>
Cause:	The attribute listed (context) was changed during customisation (or in the ADL file) and contains the syntax error described.
Action:	Correct the error described or contact your ADONIS administrator.

[aleo-31]

Message:	The target of a value assignment must be a variable!
	<context>

Cause: The described runtime error has occurred during the evaluation of an expression, which is contained in the value of the specified attribute.

Action: Correct the error described or contact your ADONIS administrator.

[aleo-32]

Message: Call of the function error()!

<context>

Cause: The described runtime error has occurred during the evaluation of an expression, which is contained in the value of the specified attribute.

Action: Correct the error described or contact your ADONIS administrator.

[aleo-34]

Message: Closing bracket '}' is missing!

<context>

Cause: The attribute listed (context) was changed during customisation (or in the ADL file) and contains the syntax error described.

Action: Correct the error described or contact your ADONIS administrator.

[aleo-35]

Message: Closing bracket '}' without opening bracket '{'!

<context>

Cause: The attribute listed (context) was changed during customisation (or in the ADL file) and contains the syntax error described.

Action: Correct the error described or contact your ADONIS administrator.

[aleo-36]

Message: The value for *<attribute name>* is missing or is of the wrong type!

Expected: *<Attribute type1>*, found: *<Attribute type2>*.

<context>

Cause: The attribute listed (context) was changed during customisation (or in the ADL file) and contains the syntax error described.

Action: Correct the error described or contact your ADONIS administrator.

[aleo-37]

Message: Over flow when concatenating two strings!

The maximum possible length of a string contains 65529 characters.

Cause:	The string made from the concatenation is too long to be saved to a string variable.
Action:	Change the concerned command or expression, so that the created string will not be too long.

[aleo-38]

Message:	Overflow while calculating an expression!
	The expression is too complex and cannot be calculated.
Cause:	The expression has a nesting depth and/or a number of operands greater than 250.
Action:	Simplify the expression or contact your ADONIS administrator.

[aleo-39]

Message:	*<Name of variable>*: is not initialised!
	<Name of function>
Cause:	Calling the function *<Name of function>* changes an array (areplace, aappend). However the variable *<Name of variable>* is not defined as an array.
Action:	Initialize the variable listed as an array before the function call or contact your ADONIS administrator.

[aleo-40]

Message:	Parameter list without function!
Cause:	A part of the expression is interpreted as a parameter list without a function name in front of it.
Action:	Correct the error described or contact your ADONIS administrator.

[alib2sgm-01]

Message:	The file could not be opened.
Cause:	The file specified for the storage of browser contents could not be opened.
Action:	The data carrier may be write protected. Delete the write protection and try to save the browser contents again. If necessary contact your system administrator.

[alib2sgm-02]

Message:	No file name has been entered!
Cause:	You have not specified a file name for the export file of a library.
Action:	Enter the name of an export file.

[alib2sgm-03]

Message: The directory name entered is invalid!

Cause: You have entered a not existing path for the export of a library.

Action: Enter a valid (=existing) path for the export.

[alibchk-01]

Message: The facet "*<facet name>*" is faulty!

Cause: A facet, which should immediately be assigned a value, is empty.

Action: Contact yourADONIS administrator.

[alibchk-02]

Message: unknown type of the facet "*<facet name>*": *<TypNr>*.

Cause: A facet contained in the library has an invalid type.

Action: Note the facet name and type and contact your ADONIS administrator.

[alibchk-03]

Message: Attribute values of type STRING cannot contain more than 3699 characters.

Cause: The value of a STRING attribute is too long.

Action: Reduce the value entered.

[alibchk-04]

Message: Attribute values of type LONGSTRING cannot contain more than 32000 characters.

Cause: The value of a LONGSTRING attribute is too long.

Action: Reduce the value entered.

[alibchk-05]

Message: Attribute values of type INTEGER must be integer.

Cause: The value of an INTEGER attribute is not integer.

Action: Verify the value and correct typing errors.

[alibchk-06]

Message: Attribute values of type DOUBLE can only be floats.

Cause: The value of a DOUBLE attribute is not in the valid format.

Action: Verify the value and correct typing errors .

[alibchk-07]

Message: Attribute values of type TIME must have the following format:

                    YY:DDD:HH:MM:SS (Year:day:hours:minutes:seconds)

Cause: The value of a TIME attribute is not in the valid format.

Action: Correct the attribute value.

[alibchk-08]

Message: The facet "*<facet name>*" is not assigned a value.

                    It must be assigned with a name of a record class.

Cause: The specified facet is not assigned with a valid value.

Action: Contact yourADONIS administrator.

[alibchk-09]

Message: The name "*<record name>*" of the facet "*<facet name>*" does not correspond with any
                    existing record class.

Cause: The specified facet is not assigned a valid value.

Action: Contact yourADONIS administrator.

[alibchk-10]

Message: The facet "*<facet name>*" must be assigned with a positive integer value.

Cause: The specified facet is not assigned a valid value.

Action: Contact your ADONIS administrator.

[alibchk-11]

Message: The value (*<value>*) of the attribute "*<attribute name>*" is not in the valid value area.

Cause: The specified attribute value is not in the authorised value area.

Action: Correct the attribute value, so that it is in the valid value area.

[alibchk-12]

Message: The attribute "*<attribute name>*" of the record class or attribute profile class "*<name>*"
                    is not valid.

Cause: No valid notebook string has been specified for the specified record.

Action: Contact your ADONIS administrator.

[alibchk-13]

Message: The attribute "*<attribute name>*" referenced via the submodel pointer does not exist in the class "*<class name>*" !

Cause: The attribute referenced via the submodel pointer doesn't exist.

Action: Contact your ADONIS administrator.

[alibchk-14]

Message: In the class "*<class name>*", the attribute "*<Attributname>*" referenced via the sub-model pointer is not of the type INTERREF!

Cause: In the specified class, the attribute referenced via the submodel pointer is not of the type INTERREF!

Action: Contact yourADONIS administrator.

[alibchk-15]

Message: There is no attribute "*<attribute name>*" defined in the class "*<class name>*" !

Cause: A required class attribute is not defined.

Action: Contact your ADONIS administrator.

[alibchk-16]

Message: The value of the facet "*<facet name>*" is too long ("*<n>*" characters)! Facet values cannot be longer than "*<m>*" characters.

Cause: The value of a facet is too long.

Action: Contact your ADONIS administrator.

[alibchk-17]

Message: Error in library attribute "Relation evaluations":

               <Error>.

Cause: The specified error has been discovered in the library attribute "Relation evaluations".

Action: Contact your ADONIS administrator.

[alibchk-18]

Message:	Error in library attribute "Sim result-mapping":
	<Error>.
Cause:	The specified error has been discovered in the library attribute "Sim result-mapping".
Action:	Contact your ADONIS administrator.

[alibchk-19]

Message:	Attribute values of type EXPRESSION cannot contain more than 3600 characters.
Cause:	The value of an EXPRESSION attribute is too long.
Action:	Reduce the value entered.

[alibchk-23]

Message:	The entry "*<entry>*" is contained several times in the facet "*<facet name>*".
Cause:	The entry displayed cannot appear more than one time in the specified facet and this entry is contained more than once.
Action:	Correct the specified facet accordingly.

[alibload-01]

Message:	The library "*<library name>*"could not be loaded. Possibly the connection to the database was lost.
Cause:	The library listed could not be loaded from the database.
Action:	Repeat the action. Should the error re-occur, close/quit ADONIS, re-start it and repeat the action again. Should the error still continue to occur, please contact your system administrator.

[alibload-02]

Message:	Initialisation of the application library "*<library name>*" has failed. Please check the attributes of all libraries which are attached to this application library.
Cause:	The library attributes of a library which is part of the application library listed contains errors.
Action:	Please check the library attributes of all libraries which are part of the application library listed. You can execute this check using the Administration Toolkit (component "Library Management"). Correct all errors which are reported.
Message:	Initialisation of the application library "*<library name>*" has failed:
	<Number of error message>
	<Error description>
Cause:	The attributes of the library contain errors.

Action:	Please check the attributes of the libraries shown. Within the message window a further error code (*Number of an error message*) will be shown specifying the type of error (*Error description*). To deal with this error please refer to the error groups ahamot, asimmap or aleo.
	If necessary contact your ADONIS consultant.
Message:	Initialisation of the application library "*<library name>*" has failed due to the following error:
	Error in relation redefinition.
Cause:	The attributes in the relation redefinition of the library contain errors.
Action:	Check the attributes of the library.
	If necessary contact your ADONIS consultant.
Message:	Initialisation of the application library "*<library name>*" has failed due to the following error:
	Unexpected error (Internal error code: *<Error code>*).
Cause:	The attributes of the library contain errors.
Action:	Note the error and contact your ADONIS consultant.

[alibload-03]

Message:	The library "*<library name>*" is currently being edited by the user "*<user name>*" and therefore cannot be loaded.
Cause:	Another ADONIS administrator is editing the mentioned library at the same time.
Action:	Wait until the library is released or speak with the other ADONIS administrator about it.
	(You can also send an appropriate message to the specified user via the ADONIS mail (menu "Extras", menu item "Messages").)

[alibmgt-01]

Message:	One imported library could not be saved in the database! Please quit ADONIS.
Cause:	An error occurred while the library imported was about to be saved. It may be a problem with the network.
Action:	Quit ADONIS, restart it and repeat the library import. Should the message appear again, please contact your system administrator.

[alibmgt-05]

Message:	This library is the standard application library. You cannot delete it!
Cause:	You have attempted to delete the standard application library. The standard application library cannot be deleted.
Action:	The standard application library cannot be deleted.

[alibmgt-07]

Message: No file name was entered!

Cause: You did not enter a file name for the export file of a library.

Action: Enter the name of the export file.

[alibmgt-08]

Message: The directory name entered is invalid!

Cause: You entered a path that does not exist for exporting a library.

Action: Enter a valid (= existing) path for the export.

[alibmgt-15]

Message: The file *<file name>* does not exist!

Cause: The path entered and/or the file name of the ABL file for the import does not exist.

Action: Enter the file name and/or the path of an existing ABL file.

[alibmgt-17]

Message: The library *<library name>* is assigned to the following users: *<user name_1, .., user name_n>*!

Cause: You tried to delete the application library listed which is assigned to the ADONIS users listed.

Action: Change the settings of the ADONIS users listed in the User Management component. You can either assign a different application library to these users or delete them from the user list.

[alibmgt-19]

Message: There are (application) models based on the library *<library>*!

Cause: You tried to delete the application library listed. However, models which are based on this application library still exist.

Action: Delete the models based on the application library listed before you delete the application library. If you wish to re-use the models later, export them as well as the application library prior to the deletion.

[alibmgt-23]

Message: The library could not be saved!

Cause: The connection to the ADONIS database server may have been lost or the server may have broken down.

Action: Quit ADONIS and restart it. Should the message appear again, please contact your database administrator.

[alibmgt-24]

Message: Invalid input for attribute *<attribute name>*:*<value>*!

Cause: You entered the value listed into the ADONIS notebook but it does not match the type of the attribute listed.

Action: Please enter a valid value.

[alibmgt-26]

Message: The class attributes of the library "*<library name>*" contain errors!

Cause: The notebook definitions of the library listed are invalid. This may cause run-time errors when the library is used in the ADONIS Business Process Management Toolkit.

Action: Close the window containing the error message by clicking on the "OK" button and save the error protocol then shown. Contact your ADONIS consultant and give him the error protocol so that the errors in the notebook definitions can be fixed.

[alibmgt-28]

Message: The attributes of the library *<library name>* contain errors! This may produce runtime errors in the ADONIS business process management Toolkit. Please pay attention to the following information.

Cause: The attributes in the library listed contain invalid values.

Action: Close the window containing the error message by clicking on the "OK" button and save the error protocol which will then be shown. Correct the library attributes (the syntax of which is described in the ADONIS administration. if necessary contact your ADONIS consultant and give him the error protocol so that the errors in the attribute definitions can be fixed.

[alibmgt-29]

Message: The file *<file name>* does not exist!

Cause: An invalid name was entered for the import file.

Action: Please enter a valid file name.

[alibmgt-31]

Message: The application library *<library name>* cannot be deleted because of one or more defined model groups!

Cause: One or more model groups could not be deleted automatically.

Action: Repeat deleting the application library listed. Should the message appear again, delete the model groups in the Model Management component or contact your ADONIS administrator.

[alibmgt-32]

Message: The application library *<library name>* cannot be deleted!

Cause: An internal error occurred while you tried to delete the application library.

Action: Please contact your ADONIS consultant.

[alibmgt-33]

Message: The business process library *<library name>* cannot be deleted!

Cause: A severe internal error occurred while you tried to delete the business process library.

Action: Please contact your ADONIS consultant.

[alibmgt-34]

Message: The Working Environment library *<library name>* could not be deleted since the user *<user name>* is currently logged in!

Cause: The Working Environment library could not be deleted since it is currently being edited by the specified user.

Action: Delete the specified library once the specified user has finished to edit this library.

(You can also send an appropriate message to the specified user via the ADONIS mail (menu "Extras", menu item "Messages").)

[alibmgt-36]

Message: The application library "*<library name>*" could not be deleted!

Cause: When attempting, to delete the application library, a serious internal error has occurred.

Action: Contact your ADONIS administrator.

[alibmgt-37]

Message: The library "*<library name>*" could not be deleted, because there are still files in the database assigned to this library!

Cause: Deleting the library specified is currently not possible, as there still exists external files assigned to this library.

Action: Delete the external files using the file management before deleting the application library or activate the option "Delete external files" when deleting the application library.

[alibmgt-38]

Message: Cannot delete application library "*<library name>*" because the model "*<model name>*" is locked by the user "*<user name>*"!

Cause: Deleting the library specified is currently not possible, as the model mentioned - which is based on that library - is currently in use by the user specifed and thus locked.

Action: Repeat the procedure later, after the user closed the model.

[alibren-01]

Message: The library "*<current library name>*" could not be renamed because

the new name "*<new library name>*" is not unique!

Cause: The name you selected for renaming cannot be used because another library in the same database already bears this name.

Action: Choose another name for one of the two libraries.

[alogin-05]

Message: Data base "*<database name>*" does not exist!

Cause: A database of the name you entered has not been catalogued on your computer.

Action: Check the database name entered or apply to your database administrator to have the respective database catalogued.

[alogin-06]

Message: ADONIS could not connect to the database! Please try again.

Cause: The connection to the ADONIS database server may have been lost or the server may have broken down.

Action: Repeat the action. Should the error occur again, please contact your system administrator.

[alogin-09]

Message: The user *<user name>* is already logged into ADONIS! The login process was stopped.

Cause: A user may not be logged in more than once. A user of the user name you entered is already logged in.

Action: Login using a different user name or wait until the user logged in logs out of ADONIS.

[alogin-10]

Message: Login impossible - a user name must be specified!

Cause: You did not enter a user name into the field 'Login'.

Action: Please enter a user name.

[alogin-11]

Message: Login impossible - a password must be specified!

Cause: You did not enter a password into the field 'Password'.

Action: Please enter the user's password.

[alogin-12]

Message: Login impossible - a database name must be specified!

Cause: You did not enter a database name into the field 'database Name'.

Action: Please enter a database name.

[alogin-13]

Message: Wrong password specified for user < *user name*> of database <*database name*>!

Cause: The password you entered is not the correct one for the user and database listed.

Action: Please enter the correct password.

[alogin-14]

Message: In the database <*database name*>, there is no user <*user name*>!

Cause: The user name you entered is not part of the list of possible users for the database listed.

Action: Please enter a user name that is permitted for the database listed or ask the ADONIS administrator to enter the user name you used before into the user list of the ADONIS database.

[alogin-15]

Message: The user <*user name*> has no execution access for the Business Process Management Toolkit!

Cause: The user of the user name you entered has no execution access for the Business Process Management Toolkit.

Action: Enter the user name of an ADONIS user who has execution access or ask your ADONIS administrator to grant you execution access.

[alogin-16]

Message: The user <*user name*> has no execution access for the Administration Toolkit!

Cause:	The user of the user name you entered has no execution access for the Administration Toolkit.
Action:	Enter the user name of an ADONIS user who has execution access or ask your ADONIS administrator to grant you execution access.

[alogin-17]

Message:	The database system is not valid!
Cause:	The database system you entered does not exist.
Action:	Enter one of the database systems permitted (parameter -sDB2, -sORACLE, -sINFORMIX or -sSQLSERVER).

[alogin-18]

Message:	ADONIS could not connect to the database - task stopped! Please contact your system administrator.
Cause:	The connection to the ADONIS database server may have been lost or the ADONIS database server may have broken down.
Action:	Repeat the action. Should the message occur again, please contact your system administrator.

[alogin-19]

Message:	At the moment the maximum number of *<number>* users is logged in to the ADONIS database *<data base name>*. Do you want to log into another ADONIS database?
Cause:	You tried to log into an ADONIS database which had the maximum number of users permitted already logged in.
Action:	Click on the "Yes" button to log into another ADONIS database or the "No" button to cancel.

[alogin-20]

Message:	The ADONIS database *<database name>* is configured for single user mode and a user is already logged in at the moment! Do you want to log into another ADONIS database?
Cause:	You tried to log into an ADONIS database which is configured for single user mode (maximum number of users permitted: 1).
Action:	Click on the "Yes" button to log into another ADONIS database or the "No" button to cancel the ADONIS message.

[alogin-21]

Message:	The selected database cannot be accessed from this application!
	The database *<database name>* was created with the version *<version number1>* You currently use *<version number2>*.

Note: You may migrate your models using the import/export tool.

Do you want to login to another ADONIS database?

Cause: You tried to log into a database which was not generated by your current ADONIS version (*version number2*).

Action: Generate a new database using your current ADONIS version. Then migrate the models and libraries from the database with the help of the ADONIS version listed (*version number1*). If necessary contact your ADONIS administrator.

[alogin-22]

Message: A character set error has been detected with the selected database!

Please contact your ADONIS administrator.

Do you want to login to another ADONIS database?

Cause: The database system and your ADONIS client use different character sets.

Note: If the database system is "Microsoft SQL Server", then there are probably different ODBC settings regarding the option "Transform OEM into ANSI" on the various clients.

Action: Contact your ADONIS administrator.

Hint: When the database system "Microsoft SQL Server" is used, the option "Transform OEM into ANSI" (program "ODBC32" in the Windows system directory "Settings") must be set to the same value for all clients with ADONIS applications. This value must also agree with the one on that client from which the ADONIS database was installed.

[amigrat-03]

Message: An error occurred during import of a user group/user.

Cause: The UDL file to be imported is incorrect.

Action: Please contact your ADONIS administrator.

[amodmgt-10]

Message: The user-specific configuration could not be loaded.

Cause: An error occurred while the user-specific configuration was about to be loaded.

Action: Quit ADONIS and restart it. Should this error re-occur, please contact your ADONIS administrator.

[amodmgt-11]

Message: Error in the value of the attribute "*<attribute name>*" (*<class name>*):

"*<Value>*" is not an instance attribute name of the same class!

Cause: The class attribute of the class listed is filled with an illegal value. The value must be the name of an (instance) attribute of the same class.

Action: Please inform your ADONIS administrator

[amodmgt-12]

Message:	There are no classes defined in the library "*<library name>*".
Cause:	No classes are defined in the specified library.
Action:	Inform your ADONIS administrator.

[amodmgt-13]

Message:	The BP library of the application library "*<library name>*" is defective!
Cause:	Your application library is defective.
Action:	Inform your ADONIS administrator.

[amodmgt-14]

Message:	The WE library of the application library "*<library name>*" is defective!
Cause:	Your application library is defective.
Action:	Inform your ADONIS administrator.

[anamegen-01]

Message:	The definition of the name generation is erroneous: Class "*<class name>*" does not have an attribute "*<attribute name>*"!
Cause:	The attribute mentioned is not defined in the shown class and thus cannot be used for the name generation.
Action:	When define a name generation, choose an existing attribute.

[anbbrow-01]

Message:	Not all values could be inserted!
Cause:	When inserting in the ADONIS browser, an error has occurred. The insertion has been aborted.
Action:	Check if the area to be pasted is not too large and if the attribute types go together. Then check if the insertion in the area you have selected is allowed.

[anbbrow-02]

Message:	The entry "*<entry text>*" is not authorised for the attribute "*<attribute name>*"! *<Additional error text>* Do you want to further edit the attribute?
Cause:	When editing in a record attribute an invalid attribute value has been entered.

Action: Enter a valid value for this attribute.

[anotebk-01]

Message: An error occurred while attempting to save the contents of the notebook!

Cause: Either the filename under which you are saving is invalid or there is not enough space on the disk or drive to which you are saving.

Action: Check filename and available space.

[anotebk-02]

Message: An error occurred while attempting to print the contents of the notebook!

Cause: Possibly there is an error with your printer configuration.

Action: Contact your system administrator.

[anotebk-03]

Message: Starting the process

<program name>

failed!

Cause: The value referenced in the PROGRAMCALL attribute cannot be resolved successfully as an external program.

Action: Possibly the "Path" environment variable is incomplete or the program has not been installed.

[aoutgen-01]

Message: Invalid filename.

Cause: An invalid filename was given to which the browser contents should be saved.

Action: Enter a valid filename.

[aoutgen-02]

Message: The file could not be opened.

Cause: The file specified into which the browser contents should be saved could not be opened.

Action: Possibly the disk is read-only. Remove the access restriction and attempt to save the browser contents again. Contact your system administrator if necessary.

[aoutgen-03]

Message: Error writing to file.

Cause: The file specified into which the browser contents should be saved could not be written.

Action: There is possibly no longer enough space on your disk. Contact your system administrator if necessary.

[aoutgen-04]

Message: Error when starting the postprocessors.

(command line: <*command*>)

Cause: When saving the browser contents in the format RTF or HTML, an error has occurred during the start of the postprocessor (jade).

Action: Verify if the postprocessor is installed in the ADONIS installation directory or contact your ADONIS administrator.

[apercalc-01]

Message: At least one model is not archived. The monthly closure will be aborted.

Cause: To execute a monthly closure, all models concerned must previously have been archived. This is not the case.

Action: Archive all concerned models.

[aperout-01]

Message: The current result display is empty!

Cause: No results are available for the current settings.

Action: Select other settings.

[aperpar-01]

Message: Error in the dynamic evaluation module "<*module name*>": The key word '<*key word name*>' is invalid or is used at an invalid location.

Cause: Your ADONIS application library contains errors.

Action: Contact your ADONIS administrator. The value of the library attribute "dynamic evaluation module" must be corrected.

[aperpar-02]

Message: Error in the dynamic evaluation module "<*module name*>": After the key word '<*key word name*>', the name of the key figure is missing.

Cause: Your ADONIS application library contains errors.

Action: Contact your ADONIS administrator. The value of the attribute "dynamic evaluation module" must be corrected.

[aperpar-03]

Message: Error in the dynamic evaluation module "*<module name>*": No formula is defined for the key figure '*<key figurename>*' .

Cause: Your ADONIS application library contains errors.

Action: Contact your ADONIS administrator. The value of the library attribute "dynamic evaluation module" must be corrected.

[aperpar-04]

Message: Error in the dynamic evaluation module "*<module name>*": The definition of the key figure '*<key figure name>*' is incomplete. The correct definition of the formula is missing behind the key word '*<key word name>*' .

Cause: Your ADONIS application library contains errors.

Action: Contact your ADONIS administrator. The value of the library attribute "dynamic evaluation module" must be corrected.

[aperpar-05]

Message: Error in the dynamic evaluation module "*<module name>*": The definition of the key figure '*<key figure name>*' contains errors. The expression specified behind the key word '*<key word name>*'

'*<expression>*

is syntactically incorrect.

Cause: Your ADONIS application library contains errors.

Action: Contact your ADONIS administrator. The value of the library attribute "dynamic evaluation module" must be corrected.

[aperpar-06]

Message: Error in the dynamic evaluation module "*<module name>*": The definition of the key figure '*<key figure name>*' contains errors. A result file type has been specified as a formula behind the key word '*<key word name>*'.

Cause: Your ADONIS application library contains errors.

Action: Contact your ADONIS administrator. The value of the library attribute "dynamic evaluation module" must be corrected.

[aperpar-07]

Message: Error in the dynamic evaluation module "*<module name>*": The redefinition of the function '*<function name>*' contains errors. No valid function indicator has been specified behind the key word '*<key word name>*'.

Cause: Your ADONIS application library contains errors.

Action: Contact your ADONIS administrator. The value of the library attribute "dynamic evaluation module" must be corrected.

[aperpar-08]

Message: Error in the dynamic evaluation module "*<module name>*": The following key figures are not defined, although they are referenced in at least one formula:

<Name of the key figures>

Cause: Your ADONIS application library contains errors.

Action: Contact your ADONIS administrator. The value of the library attribute "dynamic evaluation module" must be corrected.

[aperpar-09]

Message: Error in the dynamic evaluation module "*<module name>*": The definition of the key figure '*<key figure name>*' contains errors. The expression '*<value>*' specified behind the key word '*<key word name>*' is invalid.

Cause: Your ADONIS application library contains errors.

Action: Contact your ADONIS administrator. The value of the library attribute "dynamic evaluation module" must be corrected.

[aperpar-10]

Message: Error in the dynamic evaluation module "*<module name>*": The name of the following key figures is not unambiguous:

<name of the key figures>

Cause: Your ADONIS application library contains errors.

Action: Contact your ADONIS administrator. The value of the library attribute "dynamic evaluation module" must be corrected.

[aperpar-11]

Message: No key figures are defined for the dynamic evaluation module "*<module name>*".

Cause: Your ADONIS application library contains errors.

Action: Contact your ADONIS administrator. The value of the library attribute "dynamic evaluation module" must be corrected.

[aperpar-12]

Message: Error in the dynamic evaluation module "*<module name>*": The expression "*<value>*" is not defined.

Cause: Your ADONIS application library contains errors.

Action: Contact your ADONIS administrator. The value of the library attribute "dynamic evaluation module" must be corrected.

[aperpar-13]

Message: Error in the dynamic evaluation module "*<module name>*" behind the key word "*<key word name>*": "*<value>*" does not exist.

Cause: Your ADONIS application library contains errors.

Action: Contact your ADONIS administrator. The value of the library attribute "dynamic evaluation module" must be corrected.

[aperpar-14]

Message: Error in the dynamic evaluation module "*<module name>*" behind the key word "*<key word name>*": "*<value>*" has the wrong type (expected type: "*<Type name>*").

Cause: Your ADONIS application library contains errors.

Action: Contact your ADONIS administrator. The value of the library attribute "dynamic evaluation module" must be corrected.

[aperpar-15]

Message: An unknown dynamic evaluation module has been discovered.

Cause: Your ADONIS application library contains errors.

Action: Contact your ADONIS administrator. The value of the library attribute "dynamic evaluation module" must be corrected.

[aperpar-16]

Message: The current configuration contains errors.

No dynamic evaluation module has been defined.

Cause: Your ADONIS application library contains errors. The library attribute "Dynamic evaluation module" contains a value which does not define a correct dynamic evaluation module.

Action: Contact your ADONIS administrator. The value of the library attribute "Dynamic evaluation module" has to be corrected.

[aperress-03]

Message: The calculation of the key figure "*<key figure name>*" is cyclic.

Cause: Your ADONIS application library contains errors.

Action: Contact your ADONIS administrator. The value of the library attribute "dynamic evaluation module" must be corrected.

[aperress-04]

Message: The key figure "*<key figure name>*" could not be calculated for "*<object name>*" . Reason: *<reason>*

Cause: The specified key figure can not be calculated. Your ADONIS Application Library could possibly be defective.

Action: Contact your ADONIS administrator.

[aperress-05]

Message: The key figure "*<key figure name>*" could not be calculated, since the evaluation of the formula '*<formula value>*' is wrong.

Cause: Your ADONIS application library contains errors.

Action: Contact your ADONIS administrator. The value of the library attribute "dynamic evaluation module" must be corrected.

[aperress-06]

Message: The key figure "*<key figure name>*" could not be calculated, since the time basis belonging to it could not be found.

Cause: Your ADONIS application library contains errors.

Action: Contact your ADONIS administrator. The value of the library attribute "dynamic evaluation module" must be corrected.

[apwchgui-01]

Message: Password must not be shorter than three characters.

Cause: You entered a password shorter than three characters.

Action: Enter a password which is at least three characters in length.

[apwchgui-02]

Message: Passwords are not consistent.

Cause: You entered different passwords in the two input fields.

Action: Enter the same password in both input fields.

[apwchgui-03]

Message: The old password is wrong.

Cause: You entered an incorrect password.

Action: Enter your previous password correctly.

[apwchgui-04]

Message: ADONIS could not connect to the database. Task stopped.

Cause: The connection to the ADONIS database server may have been interrupted.

Action: Repeat the action. Should the error re-occur, please contact your system administrator.

[apwchgui-05]

Message: The new password contains restricted characters. Please use characters in the ASCII (resp. ANSI) range 32 to 126 only (exceptions: ' \ { } and @).

Cause: You entered illegal restricted characters or umlauts when entering the password.

Action: The characters permitted for the password are only the numbers **0 to 9**, the letters **a to z** and **A to Z**, blanks and the symbols **! # $ % & ( ) * + , - . / : ; < = > ? @ [ ] ^ _ { | } ~**. Umlauts, ß and other symbols must <u>not</u> be used. Please enter the password according to the restrictions above.

[apwchgui-06]

Message: The new password does not match the password rule:

<Rule>

Cause: The password you entered does not match the password rule defined by the ADONIS administrator

Action: For the password certain rules have been defined.

The characters permitted for the password are only the numbers **0 to 9**, the letters **a to z** and **A to Z**, blanks and the symbols **! # $ % & ( ) * + , - . / : ; < = > ? @ [ ] ^ _ { | } ~**. Umlauts, ß and other symbols must <u>not</u> be used.

Please enter a password according to the restrictions above.

[aqueryed-01]

Message: The value is already in the list!

Cause: You entered an enumeration value which already exists.

Action: Please enter a value that does not yet exist.

[aqueryed-03]

Message: Invalid position on creating a reference!

Cause: The selected position on creating a reference is invalid.

Action: Select another position or click repeatedly on the button "other" to be shown the valid positions for a reference.

[aqueryed-04]

Message: The menu item "*<name of the menu item>*" already exists!

Cause: Within the query, the specified menu item appears repeatedly. This is not allowed.

Action: Give another name to the menu item.

[aregex-01]

> **ATTENTION:**   The error message [aregex-01] contains different text parts which are all shown below. Further information can be retrieved from the chapter "Regular Expressions" (see chap. 4., p. 534).

Message: The entered search pattern is not a valid regular expression!

Please note the following information:

<error-specific message>

Cause: The search pattern entered does not meet the syntactic requirements of regular expressions.

Action: Please correct the entry accordingly.

Message: The entered search pattern is not a valid regular expression!

Wrong masking (trailing '\').

Cause: The search pattern entered does not meet the syntactic requirements of regular expressions.

Action: Please correct the entry using the masking (back slash) properly.

Message: The entered search pattern is not a valid regular expression!

Missing bracket ('[' without ']' or vice versa).

Cause: The search pattern entered does not meet the syntactic requirements of regular expressions.

Action: Please correct the entry using the brackets properly.

Message: The entered search pattern is not a valid regular expression!

Missing bracket ('(' without ')' or vice versa).

Cause: The search pattern entered does not meet the syntactic requirements of regular expressions.

Action: Please correct the entry using the brackets properly.

Message: The entered search pattern is not a valid regular expression!

Missing bracket ('{' without '}' or vice versa).

Cause: The search pattern entered does not meet the syntactic requirements of regular expressions.

Action: Please correct the entry using the brackets properly.

Message: The entered search pattern is not a valid regular expression!

Wrong numerical range specified (not a number, number too large, more than two numbers or first number larger than second).

Cause: The search pattern entered does not meet the syntactic requirements of regular expressions.

Action:	Please correct the entry using the numerical range properly (e.g. {2,3}).
Message:	The entered search pattern is not a valid regular expression!
	Invalid range specified.
Cause:	The search pattern entered does not meet the syntactic requirements of regular expressions.
Action:	Please correct the entry using a valid range (e.g. [a-z]).
Message:	The entered search pattern is not a valid regular expression!
	Missing expression before '?', '*' or '+'.
Cause:	The search pattern entered does not meet the syntactic requirements of regular expressions.
Action:	Please correct the entry using an expression before '?', ' *' oder '+'.
Message:	The entered search pattern is not a valid regular expression!
	Invalid collating element.
Cause:	The search pattern entered does not meet the syntactic requirements of regular expressions.
Action:	Please correct the entry using a valid collating element.
Message:	The entered search pattern is not a valid regular expression!
	Invalid character class.
Cause:	The search pattern entered does not meet the syntactic requirements of regular expressions.
Action:	Please correct the entry using a valid character class (e.g. [:alpha:]).
Message:	The entered search pattern is not a valid regular expression!
	Invalid backreference number.
Cause:	The search pattern entered does not meet the syntactic requirements of regular expressions.
Action:	Please correct the entry using a backreference number to an existing pattern.

[aregex-02]

Message:	Not enough memory!
	Search could not be performed.
Cause:	The search cannot be executed because of not enough free memory.
Action:	Exit other programs and try again.

[arelrdef-02]

Message:	Error in library attribute "*name*" ("*library name*"):
	"*name*" is not a valid relation name!
Cause:	There are no existing relations with the specified name.
Action:	Change the corresponding library attribute so that it uses an existing relation.

[arelrdef-03]

Message:	Error in library attribute "*name*" ("*library name*"):
	There is an error in attribute "*attribute name*" !
Cause:	An error has occurred in the specified attribute. Possible causes:
	The attribute has been specified without its class or relation.
	The specified attribute does not exist in the specified class or relation.
	The attribute type specified in the parameter "type" is invalid.
	The value specified in the parameter "value" could not be assigned to the new attribute.
Action:	The following error actions are possible:
	Before you specify an attribute, you must specify the class/relation it belongs to.
	Verify the name of the attribute, since it has not been found in the previously specified class/relation.
	Verify if the value specified in the parameter "type" is available in the following list: "integer", "double", "string", "distribution", "time", "enumeration", "enumerationlist", "longstring" or "programcall".
	Change the value of the parameter "value" in such a way that it meets the type specified in the parameter "type".

[arelrdef-04]

Message:	Error in library attribute "*name*" ("*library name*"):
	There is an error in the facet "*facet name*" !
Cause:	An error has occurred in the specified facet. Possible causes:
	The facet has been specified without its belonging attribute.
	The parameter "value" has not been specified.
	The specified facet does not exist in the specified attribute.
	The value specified in the parameter "value" could not be assigned to the facet.
Action:	The following error actions are possible:
	Before you specify a facet, you must specify the belonging attribute.
	Verify if the parameter "value" has been specified.
	Verify the name of the facet, since it has not been found in the previously specified attribute.
	Change the value of the parameter "value" in such a way that it corresponds to the type of the facet.

[arelrdef-05]

Message:	Error in library attribute "*name*" ("*library name*"):
	"*class name*" is not a valid class name!
Cause:	When creating a new relation or changing an existing meta model class, an invalid class name has been specified.

Action: Verify the name of the origin and target classes defined for the relation. Check the names of the meta model classes, which should be changed.

[areposui-01]

Message: "*name*" already exists. Select another name!

Cause: An attribute profile or an attribute profile group with the specified name already exists. Attribute profiles must have globally unambiguous names (i.e. you cannot have an attribute profile with the same name in no other attribute profile group), the name of attribute profile groups must only be unambiguous inside your own group.

Action: Specify an unambiguous name.

[areposui-02]

Message: "*attribute profile (groups) name*" is currently being edited by another ADONIS user and so cannot be edited!

Cause: You have attempted to edit the displayed attribute profile or the displayed attribute profile group, but the profile or the profile group is currently edited by another user.

Action: Wait until the other user has finished the editing. If necessary, you can send an express-message to the user.

[areposui-05]

Message: The entry "*value*" is not allowed for the attribute "*name*"

Cause: You have entered an invalid attribute value.

Action: Enter a valid value.

[areposui-06]

Message: The changes could not be saved! Update the attribute profile list and execute the action again.

(Error code: *code*)

Cause: The attribute profile or the attribute profile groups concerned have perhaps in the meanwhile been deleted by another user. The connection to the database may also be interrupted.

Action: Update the attribute profile list and then carry out the action again. Pay attention to the error messages (e.g. database error), which could also be deleted. If the problem reoccurs, note the specified error code and contact your ADONIS administrator.

[areposui-08]

Message: The attribute profile has already been deleted by another ADONIS user. Update the attribute profile list.

Cause: The attribute profile has already been deleted by another ADONIS user.

Action: Update the attribute profile list.

[areposui-09]

Message: "*name with version number*" already exists. Select another version number!

Cause: An attribute profile with this name and version number already exists.Attribute profiles must have globally unambiguous names (i.e. there shall not be an attribute profile group with the same name in another attribute profile group).

Action: Give another name or another version number.

[areposui-10]

Message: Not all attribute profile groups could be deleted. Some of the attribute profiles contained in the groups are possibly already used and so could not be deleted. New attribute profiles may also have been created meanwhile by another ADONIS user in the group concerned. Update the attribute profile list and try again to delete the groups.

Cause: You have attempted to delete one or more attribute profile groups. This is not possible according to the specified reasons.

Action: Update the attribute profile list and then try again to delete the groups.

[areposui-11]

Message: The attribute profile group "*name*" is currently updated by another ADONIS user. This is why no new attribute profiles could be created at the moment. Try again later.

Cause: An other user is currently editing the current attribute profile group. To avoid inconsistencies, it is not possible in the meantime to create new attribute profiles in this attribute profile group.

Action: Wait until the other user has finished his editing and try it again.

[areposui-12]

Message: The selected attribute profiles are not based on the same class!

Cause: The attribute profiles selected for editing are not instances from the same class.

Action: For the editing, select attribute profiles from only one class.

[areposui-13]

Message: The move in the attribute profile group "*<group>*" is wrong!

This can be due to the following causes:

- the attribute profile group has meanwhile been deleted by another user.

- you have attempted to move a super ordinated group to a group which is one of its subordinated groups.

- An attribute profile group with the same name as the attribute profile group to move already exists in the target group.

If necessary update the attribute profile list and repeat the action.

Cause: 1. The attribute profile group has been meanwhile deleted by another user.

2. You have attempted to move a subordinated group to one of its subordinated groups.

3. An attribute profile group with the same name as the attribute profile group to move already exists in the target group.

Action: Depending on the cause, the move to the selected target group is under impossible conditions.

[areposui-14]

Message: The attribute profile group concerned has been already deleted by another ADONIS user.

Update the attribute profile list and then repeat the action.

Cause: The attribute profile group concerned does not exist any more.

Action: Update the attribute profile list and repeat the action for as long as necessary.

[arightmg-01]

Message: Could not define new user group!

Cause: An error occurred while a user group was about to be created.

Action: Quit ADONIS and restart it. Should this error re-occur, please contact your system administrator.

[arightmg-02]

Message: Could not delete selected user group!

Cause: An error occurred while a user group was about to be deleted.

Action: Quit ADONIS and restart it. Should this error re-occur, please contact your system administrator.

[arightmg-03]

Message: No new name was entered!

Cause: You did not enter a new name when renaming the group.

Action: Please enter a new name when renaming the group.

[arightmg-05]

Message: An error has occurred when loading the user group containing the user !

Cause: Not all user groups could be loaded.

Action: Quit ADONIS and restart it. Should this error re-occur, please contact your ADONIS administrator.

[arightmg-06]

Message: All user groups to which the user is assigned could not be loaded!

Cause: An error occurred while the user groups to which the user is assigned were about to be loaded.

Action: Quit ADONIS and restart it. Should this error re-occur, please contact your ADONIS administrator.

[arightmg-07]

Message: An error has occurred when creating the model group *<model group name>*!

Cause: An error occurred while a new model group was about to be defined/generated.

Action: Update the model group list and repeat the action. Should the error reappear, exit ADONIS and re-start the application. Should the error appear again, contact your ADONIS administrator.

[arightmg-08]

Message: An error has occurred when deleting the model group *<model group name>*!

Cause: An error occurred while deleting a model group.

Action: Update the model group list and repeat the action. Should the error appear again, exit ADONIS and re-start the application. Should then error occur repeatedly, contact your ADONIS administrator.

[arightmg-11]

Message: An error has occurred when assigning the user *<user name>* to the user group *<user group name>*!

Cause: An error occurred while users were about to be assigned to a user group.

Action: Quit ADONIS and restart it. Should this error re-occur, please contact your ADONIS administrator.

[arightmg-12]

Message: An error has occurred when assigning the user group *<user group name>* to the user *<user name>*!

Cause: An error occurred while user groups were about to be assigned to a user.

Action: Quit ADONIS and restart it. Should this error re-occur, please contact your ADONIS administrator.

[arightmg-13]

Message: An error has occurred during the assignment of *<model name>* (*model type*) to model group *<model group name>*!

The model may have possibly been deleted by another user .

Cause: An error has occured during the assignment of a model to a model group, as this model group may have meanwhile been deleted by another user.

Action: Close the error message - the model group list will automatically be updated - and then repeat the action. Should the error appear again, exit ADONIS and restart the application. Should the error occur repeatedly, contact your ADONIS administrator.

[arightmg-17]

Message: An error has occurred when assigning the user group to the *<user group name>* model group *<model group name>*!

Cause: The access rights of the user group could not be set to the model group. The user or model group has possibly been deleted by another ADONIS user in the meanwhile.

Action: Update the user group list or the model group list and repeat the action. Should the error appear again, exit ADONIS and restart the application. Should the error occur repeatedly, contact your ADONIS administrator.

[arightmg-19]

Message: The selected user group *<user group name>* is not empty and cannot be deleted!

Cause: A user group which still contains users may not be deleted.

Action: Remove all user references from the user group, either by changing the assignment of users or by deleting users from the ADONIS database. Then start the deletion process again.

[arightmg-20]

Message: The selected model group *<model group name>* is not empty and cannot be deleted!

Cause: A model group, which still contains other model groups or models, may not be deleted.

Action: Remove all model references and model groups from the model group, either by changing the assignment of models or by deleting the models - or model groups, respectively - from the ADONIS database. Then start the deletion process again.

[arightmg-21]

Message: A model group with the name *<model group name>* already exists!

Cause: All model groups contained in a model group must have a non-ambiguous name.

Action: Update the model group list to be aware of changes of other ADONIS users. Rename the model group with a new name that is non-ambiguous within the hierarchically superordinated group and restart the action.

[arightmg-22]

Message: A user group with the name *<user group name>* already exists!

Cause: Each user group must have a non-ambiguous name.

Action: Rename the user group with a new non-ambiguous name and start the action anew.

[arightmg-23]

Message: An error has occurred when renaming the user group *<user group name>* could not be renamed!

Cause: An error occurred while one of the user groups was about to be renamed.

Action: Quit ADONIS and restart it. Should this error re-occur, please contact your ADONIS administrator.

[arightmg-24]

Message: An error has occurred when renaming the model group *<model group name>*!

Cause: An error occurred while a model group was about to be renamed.

Action: Update the model group list and repeat the action. Should the error appear again, exit ADONIS and re-start the application. Should the error appear repeatedly, contact your ADONIS administrator.

[arightmg-25]

Message: An error has occurred when moving the reference to the model *<model name>* (*<model type>*)!

Cause: An error occurred while a model was about to be moved.

Action: Update the model group list and repeat the action. Should the error appear again, exit ADONIS and restart the application. Should the error appear repeatedly, contact your ADONIS administrator.

[arightmg-27]

Message: An error has occurred when copying the reference in the model *<model name>* (*<model type>*) to the model group *<model group name>*!

Changes have possibly been carried out by another user. Update the model group list.

Cause: An error has occurred when copying a model reference. The target model group may have been deleted by another user in the meanwhile.

Action: Update the model group list and repeat the action. Should the error appear again, exit ADONIS and re-start the application. Should the error appear repeatedly, contact your ADONIS administrator.

[arightmg-29]

Message: When moving the model group *<model group name>* into model group *<target-model group name>* an error has occurred!

Cause: An error has occurred when moving a model group.

Action: Update the model group list and repeat the action. Should the error appear again, exit ADONIS and re-start the application. Should the error appear repeatedly, contact your ADONIS administrator.

[arightmg-30]

Message: The model group *<target model group name>* already contains a model group with the name *<model group name>*!.

Cause: A model group has been moved into a group which already contains a group of the same name.

Action: Please move the model group to a different group.

[arightmg-31]

Message: Cannot create a new main model group!

Cause: An error occurred while a new main model group was about to be created.

Action: It is not possible to create new main model groups in the Business Process Management Toolkit. Contact your ADONIS administrator, who can create new main model groups with the Administration Toolkit.

If this error occurs in the Administration Toolkit, update the model group list and repeat the action. Should the error re-occur, exit ADONIS and re-start the application. Should the error appear repeatedly, contact your ADONIS administrator.

[arightmg-32]

Message: An error has occurred when deleting the reference of a model *<model name>* (*model type*)!

Cause: An error occurred while a model reference was about to be deleted.

Action: Update the model group list and repeat the action. Should the error appear again, exit ADONIS and re-start the application. Should the error appear repeatedly, contact your ADONIS administrator.

[arightmg-33]

Message: The target model group *<model group name>* is write protected and cannot be changed!

Cause: You do not have sufficient rights to change the respective target model group you wish to change.

Action: Contact your ADONIS administrator. He is responsible for assigning access rights on models and model groups to users.

[arightmg-34]

Message: The reference to *<model name>* (*model type*) cannot be deleted, since the model is no longer contained in any model group!

Cause: There shall always be at least one reference to each model.

Action: Delete the last reference to a model using the Administration Toolkit or contact your ADONIS administrator.

[arightmg-35]

Message: The model *<model name>* (*model type*) cannot be assigned to a model group!

Cause: The specified model could not be assigned to the selected model group.

Action: Update the model group list and repeat the action. Should the error occur again, exit ADONIS and restart the application. Should the error appear again, contact your ADONIS administrator.

[arightmg-36]

Message: The reference from the model *<model name>* (*model type*) cannot be reassigned, since the model is currently opened with write access by the user *<user name>* !

Cause: The model is currently being edited by another user. The reference cannot be moved.

Action: Make sure that the user which is currently editing the model, closes it or you can move the model reference later.

[arightmg-37]

Message: The reference to the model *<model name>* (*model type*) cannot be deleted, since the model is currently opened with write access by the user *<user name>*!

Cause: The model is currently edited by another user. The reference cannot be deleted.

Action: Make sure that the user which is currently editing the model, closes it or you can delete the model reference later.

[arightmg-38]

Message: The reference to the model "*<model name>*" (*<model type>*) cannot be copied, since the model is currently opened with write access by the user *<user name>*!

Cause: The model is currently edited by another user. The reference cannot be copied.

Action: Make sure that the user which is currently editing the model, closes it or you can copy the model reference later.

  (You can also send an appropriate message to the specified user via ADONIS mail (menu "Extras", menu item "Message").)

[arightmg-39]

Message: The model group "*<model group name>*" cannot be moved, since the following models it contains are currently opened with write access by the following users:

<model name> (*<model type>*) - *<user name>*

Cause: Models from the specified model groups are being edited by other users. The model group cannot be moved.

Action: Make sure that the user, who is currently editing the model, closes it or you can move the model group later.

(You can also send an appropriate message to the specified user via ADONIS mail (menu "Extras", menu item "Message").)

[arightmg-40]

Message: The reference to the model "*<model name>*" (*<model type>*) cannot be assigned, since the model is currently opened with write access by the user *<user name>*!

Should the action be continued because of the expanded administrator rights?

Cause: The model is currently being edited by another user. The model reference cannot be reassigned.

Action: Make sure that the user, who is currently editing the model, closes it or you can arrange the model reference later.

(You can also send an appropriate message to the specified user via ADONIS mail (menu "Extras", menu item "Message").)

[arightmg-41]

Message: The reference to the model "*<model name>*" (*<model type>*) cannot be deleted, since the model is currently opened with write access by the user *<user name>*!

Should the action be continued because of the expanded administrator rights?

Cause: The specified model is currently being edited by another user. The model reference cannot be deleted.

Action: Make sure that the user, who is currently editing the model, closes it or you can delete the model reference later.

(You can also send an appropriate message to the specified user via ADONIS mail (menu "Extras", menu item "Message").)

[arightmg-42]

Message: The reference to the model "*<model name>*" (*<model type>*) cannot be copied, since the model is currently opened with write access by the user *<user name>*!

Should the action be continued because of the expanded administrator rights?

Cause: The specified model is currently edited by another user. The model reference cannot be copied.

Action: Make sure that the user, who is currently editing the model, closes it or you can copy the model reference later.

(You can also send an appropriate message to the specified user via ADONIS mail (menu "Extras", menu item "Message").)

[arightmg-44]

Message: Insufficient rights for at least one of the model groups selected!

Cause: You have no write access on at least one of the model groups selected.

Action: Update the model group list and/or select only the model groups for which you have write access rights.

[arightmg-45]

Message: You have no write access to the super ordinated model group of the model "*<model name>*"! Update the model group list or select another model group.

Cause: You have no write access to the super ordinated model group of the selected model!

Action: Update the model group list or select another model group.

[arightmg-46]

Message: You do not have sufficient rights to delete or move the model group "*<model group name>*"!

Cause: You have no write access rights to the specified model group and therefore cannot delete or move it.

Action: If necessary, contact your ADONIS administrator, to obtain write access rights to the specified model group.

[arightmg-47]

Message: The user group structure has been changed by another "ADONIS" user.

Update the user group list!

Cause: The user group structure has been changed by another "ADONIS" user.

Action: Update the user group list!

[arightmg-48]

Message: The model group "*<model group name>*" cannot be moved to a specific subgroup.

Cause: You have attempted to move a model group to one of your subgroups.

Action: Move the specified model group to another model group, which is not a subgroup of the specified model group.

[arightmg-49]

Message:	The reference to the model *<model name>* (*model type*) cannot be moved, since you do not have enough access rights in the target model group.
Cause:	You have attempted to move a model reference to a model group, in which you have no write access rights.
Action:	You cannot move the reference to the selected model group. If necessary, contact your ADONIS administrator to adapt your rights.

[ascope-01]

Message:	The library could not be saved.
Cause:	Due to an internal error the library changed could not be saved.
Action:	Please inform your ADONIS consultant.

[ascope-02]

Message:	The character "*<character>*" must not be used.
Cause:	The new value by which you wish to extend the value range contains an illegal character.
Action:	Please enter a value which does not contain the illegal character.

[ascope-03]

Message:	This value is already contained in the list.
Cause:	You want to extend the value range of an attribute by a value which is already part of the value range.
Action:	The value is already part of the value range. Therfore, you do not need to add it. Just continue with your work.

[ascope-04]

Message:	This library does not contain classes with extendable attributes.
Cause:	The library selected contains no classes with attributes of which the value ranges could be extended.
Action:	Please select a different library. If you should need extendable attributes in your current library, contact your ADONIS consultant.

[ascope-05]

Message:	*<Library name>*:
	The new attribute value range is too large. All changes will be cancelled and the previous attribute value range will be restored.

Cause:	You have extended the value range of attributes in the library listed. The new value range of at least one attribute is too large to be stored/saved in the database.
Action:	Extend the value ranges to a little lesser extent. Instead, you may also contact your ADONIS consultant.

[ascript-01]

Message:	*<Token>* is neither a key word nor a defined procedure!
Cause:	The AdoScript contains no command, which is either defined as internal or as a procedure.
Action:	Check and correct the AdoScript.

[ascript-02]

Message:	*<Token>* without CASE or OBJECTIF!
Cause:	The attribute contains an element, in which a preceding CASE or OBJECTIF element is missing.
Action:	Check and correct the AdoScript.

[ascript-03]

Message:	*<Token>* without ITEM!
Cause:	The attribute contains AdoScript commands, which are not assigned to a menu item.
Action:	Check and correct the AdoScript.

[ascript-05]

Message:	*<Token1>* without *<Token2>*!
Cause:	The AdoScript contains a branching or a loop end, in which the opening token is missing.
Action:	Check and correct the AdoScript.

[ascript-06]

Message:	*<Token1>* without *<Token2>*!
Cause:	The AdoScript contains a branching or a loop end, in which the ending token is missing.
Action:	Check and correct the AdoScript.

[ascript-09]

Message: CC without message element!

Cause: The AdoScript contains a CC call, which is not followed by a message element.

Action: Check and correct the AdoScript.

[ascript-10]

Message: CALL with defective attribute function: "*<Text>*"!

Cause: The value of the parameter function is not a valid function specification.

Action: Check and correct the AdoScript.

[ascript-11]

Message: No function "*<function name* is existing in the DLL "*<file name>*">"!

Cause: Using CALL in the AdoScript, you have attempted to call a function which is not defined in a DLL.

Action: Check and correct the AdoScript.

[ascript-12]

Message: "*<Token>*" is not a command in message port "*<message port>*"!

Cause: In the AdoScript, using SEND or CC, a command has been sent to the specified message port, where it is not defined .

Action: Check and correct the AdoScript.

[ascript-13]

Message: message port "*<Name>*" does not exist!

Cause: In the AdoScript, using SEND or CC, a command has been sent to the specified, non existing message port.

Action: Check and correct the AdoScript.

[ascript-14]

Message: defective call of *<procedure name>*!

Cause: When calling this procedure, the parameters specified are not the exact same parameters that are specified in the procedure definition.

Action: Check and correct the AdoScript.

[ascript-15]

Message:	PROCEDURE without definition element!
Cause:	The AdoScript contains a PROCEDURE element, which is not followed by a definition element.
Action:	Check and correct the AdoScript.

[ascript-16]

Message:	Type incompatibility in a procedure call!
	<function call>
	<context>
Cause:	When calling the procedure, at least one parameter is not of the type defined in the procedure definition.
Action:	Check and correct the AdoScript at the specified place.

[ascript-17]

Message:	The old syntax (IF...ENDIF) cannot be used together with the new syntax (IF...{...}) !
Cause:	The AdoScript contains both constructs of old and of new syntax.
Action:	Check and correct the AdoScript.

[ascript-18]

Message:	Starting the process
	<Process name>
	failed!
Cause:	The START command has been called from an AdoScript but the program could not be started.
Action:	Check and if necessary correct the AdoScript. Make sure that the program to be started exists in the respective directory.

[ascript-19]

Message:	LEO call with unauthorised index (set-cur-elem-index:*<index>*)!
	Number of parsed elements: *<number>*
Cause:	The LEO command has been called in an AdoScript with the index specified with set-current-index outside the authorised area (from 0 to (number of parsed elements) - 1).
Action:	Check and correct the AdoScript.

[ascript-20]

Message:	LEO call with *<indication >* without previous parse!
Cause:	The LEO command has been called in an AdoScript, where no text to be parsed has been specified. However, this must be the case for each call of LEO.
Action:	Check and correct the AdoScript.

[ascript-21]

Message:	The external DLL "*<DLL name>*" could not be loaded!
Cause:	Within an AdoScript the specified DLL file is called from a CALL command although the file dos not exist or is not accessible.
Action:	Enable access to the specified DLL file or correct the CALL command.

[ascript-22]

Message:	Abnormal terminition of function "*<Funktionsname>*" in DLL"*<DLL name>*"!
	Please contact your ADONIS consultant if this problem persists.
Cause:	Within an AdoScript an executed CALL command was aborted due to a fatal error within the specified DLL file.
Action:	Correct the CALL command or update the DLL file with a corrected version.

[ascript-23]

Message:	Double definition of ON_EVENT *<event name>*!
Cause:	In the attribute "external linking" there are more than one event handlers defined for one event.
Action:	Check and correct the AdoScript.

[ascript-25]

Message:	PROCEDURE *<procedure name>* is currently being executed and therefore cannot be overwritten!
Cause:	An AdoScript with the specified procedure is executed. The procedure exists already and is executed simultaneously. Therefore the procedure overwrites itself.
Action:	Ensure that the specified procedure is defined only once (globally).

[asetsel-01]

Message:	The parameter of the selected arrangement function could not be changed!
Cause:	An error occurred while attempting to change the parameter of the selected arrangement function.

Action: Close ADONIS and re-start it. If this error re-occurs, contact your ADONIS administrator.

[asetsel-02]

Message: No quotation marks may be used in the name of the arrangement function and the name may not be blank!

Cause: Quotation marks were specified in the name of the arrangement function or the name was left blank.

Action: Enter a new name.

[asetsel-04]

Message: The new arrangement function could not be added!

Cause: An error occurred when attempting to add a new arrangement function.

Action: Close ADONIS and re-start it. If this error re-occurs, contact your ADONIS administrator.

[asetsel-05]

Message: The arrangement function could not be deleted!

Cause: An error occurred when attempting to delete an arrangement function.

Action: Close ADONIS and re-start it. If this error re-occurs, contact your ADONIS administrator.

[asetsel-08]

Message: An arrangement function with this name already exists !

Cause: A unique name must be entered for each arrangement function.

Action: Retry the action but enter a different name.

[asetsel-10]

Message: The process hierarchy function is not defined for *<model type>*!

Cause: The arrangement function is not defined for this model type.

Action: Open a model or create a model for which it is possible to build a hierarchy. If the model type required for this is not known, then contact your ADONIS administrator, who is competent in the configuration of the (not user-definable) process hierarchy.

[asgmlexp-01]

Message: Error while writing the export data.

Cause: An error occured while writing the data. Possibly there is not enough space on the disk.

Action: Check if there is enough space on the disk where the data is stored or on the disk where the "TEMP" folder is located.

[asgmlexp-02]

Message: Cannot load model.

Please refresh the model list.

Cause: One of the models selected for export could not be loaded. Possibly this model was renamed or deleted by another user.

Action: Click on the button "Refresh" before selecting your models.

[asgmlexp-05]

Message: Model *<model name>*, Instance *<instance name>*:

The subprocess called does not exist.

The program will continue.

Do you want to be notified of further identical errors?

Cause: The specified instance refers to a model which cannot be loaded from database (does not exist).

Action: Correct the model reference.

[asgmlexp-06]

Message:

Model *<model name>*, instance *<instance name>*:

The referenced object does not exist.

The program will continue.

Do you want to be notified of further identical errors?

Cause:

The specified instance refers to a model inter-reference. This model no longer exists.

Action:

Correct the model reference.

[asgmlexp-07]

Message:

Model *<model name>*, instance *<instance name>*:

Invalid value for program and/or parameter.

The program will continue.

Do you want to be notified of further identical errors?

Cause:

The content of an attribute of type "Programcall" (e.g. External documentation) is invalid.

Action:
> Correct the attribute.

[asgmlexp-08]

Message: Model *<model name>*, instance *<instance name>*:

Error when determining the target model.

The process will be continued.

Should messages of this type still be shown?

Cause: The specified instance has an inter model relation with no target instance.

Action: Correct the defective reference accordingly.

[asgmlexp-09]

Message: Error in the attribute "documentation configuration".

Invalid sorting mode: "*<sorting mode>*"

Cause: In the library attribute, an invalid value has been specified as sorting mode.

Action: Repair the error using the Administration Toolkit or contact your ADONIS administrator.

[asgmlexp-10]

Message: Error in the attribute "documentation configuration".

Invalid target directory for the referenced documents: "*<target directory>*"

Cause: The specified target directory does not exist and cannot be created.

Action: Verify you have write access rights in the target directory and if there is enough memory space in the target directory.

[asgmlexp-12]

Message: Error in the attribute "documentation configuration".

Invalid model type: "*<model type>*"

Cause: The specified model type does not correspond to any model type in the library.

Action: Verify the exact spelling of the model type. Correct the error using the Administration Toolkit or contact your ADONIS administrator.

[asgmlexp-13]

Message: Error in the attribute "documentation configuration".

No default LIBRARY element is defined in a SOURCE Block.

Cause:	In the library attribute "documentation configuration", no standard settings are defined in an documentation export.
Action:	Correct the error using the Administration Toolkit or contact your ADONIS administrator.

[asgmlexp-14]

Message:	Error in the attribute "documentation configuration".
	The value "*<value>*" in the LIBRARY element of the row *<no.>* is not a whole number.
Cause:	The value specified in the library attribute "documentation configuration" not a whole number.
Action:	Correct the error using the Administration Toolkit or contact your ADONIS administrator.

[asgmlexp-15]

Message:	Model *<model name>*, instance *<instance name>*:
	Create program and/or parameter: An external file was referenced but could not be found.
	The process will be continued.
	Should messages of this type still be shown?
Cause:	In an attribute, either an external file has been referenced, which has been meanwhile deleted or moved or an internet address has been entered.
Action:	Correct the reference in the model or copy the missing file to the specified directory. If you have specified an internet address in attributes click on the "No" button to suppress messages of this type in the future.

[asgmlexp-16]

Message:	Program error: *<error code>*(*<error number>*).
Cause:	A general program error has occurred.
Action:	Note the error code and error number and contact your ADONIS administrator.

[asgmlexp-17]

Message:	Error in the attribute "Documentation configuration".
	No LIBRARY element has been defined for the model type "*<model type name>*".
Cause:	No element "LIBRARY" has been defined in the library attribute "Documentation configuration" of an element EXPORT.
Action:	Add an element LIBRARY for the specified model type.

[asgmlexp-18]

Message: File "*<file name>*" could not be copied.

Cause: The file could either not be found or not be created.

Action: Verify if the file in the specified directory exists, if there is enough free hard disk memory available and if you have the required access rights.

[asgmlexp-19]

Message: Graphic for the model "*<model name>*" could not be generated.

Cause: There is either not enough available memory space in the target directory or you do not have sufficient access rights to the target directory.

Action: Free up memory space or have your access rights extended. Alternatively select another target directory before the generation without the above-mentioned restrictions.

[asgmlexp-20]

Message: The specified path "*<path>*" is invalid for the referenced documents.

The process will be continued.

Cause: The path contains invalid characters or an invalid structure.

Action: Make sure during the entry of path names, that the total amount does not exceed 120 characters and that no more than seven directories are contained in the path. Moreover the characters **/ \ : * ? < >** and **|** are not allowed.

[asimmap-01]

Message: The sim mapping could not be interpreted correctly.

Cause: A non-identifiable error occurred during the interpretation of the sim mapping.

Action: Check the sim mapping in the library configuration in the ADONIS administration Toolkit and correct possible errors or contact your ADONIS administrator.

[asimmap-02]

Message: Syntax errors were found in the sim mapping.

Cause: There is a syntax error in the sim mapping. Perhaps a closing inverted comma is missing or not all of the keywords are written in capital letters.

Action: Check the sim mapping in the library configuration in the ADONIS administration Toolkit and correct possible errors or contact your ADONIS administrator.

[asimmap-03]

Message: The expression "*<indicator>*" (library attribute "sim mapping") contains errors.

Cause:	An invalid indicator has been used in the sim mapping.
Action:	Correct the specified indicator in the library attribute "sim mapping" or contact your ADONIS administrator.

[asimmap-04]

Message:	The attribute "*<attribute name>*" of the sim mapping is corrupt.
Cause:	The sim mapping is incomplete.
Action:	Add the attribute listed to the sim mapping or contact your ADONIS administrator.

[asimmap-05]

Message:	No entry parameters have been defined. Contact your "ADONIS" administrator to define entry parameter combinations in the library attribute "sim mapping".
Cause:	Your ADONIS application library contains errors. The attribute "sim mapping" must be corrected.
Action:	Contact your ADONIS administrator.

[asimtext-01]

Message:	The library attribute "SimText" is not defined.
Cause:	The library attribute "SimText" is not defined.
Action:	Define the library attribute "SimText" in the Administration Toolkit. If necessary, contact your ADONIS administrator.

[asimtext-02]

Message:	The library attribute "SimText" is not defined. The expression "*<Expression>*" is incomplete or undefined.
Cause:	The expression listed could not be found in the library attribute "Simtext". The library attribute "Simtext" of the BP library either contains a value which is incorrect or incomplete or is not defined.
Action:	Check the library attribute "Simtext" in the Administration Toolkit and correct the definition of the expression listed. If necessary, contact your ADONIS administrator.

[asrchtlb-01]

Message:	Saving the contents of the model select box failed!
Cause:	An illegal file name was entered in the "Save as" window of the model select box.
Action:	Please enter a file name that is permitted and make sure that there is sufficient storage space on the data medium.

[astdcfg-01]

Message: Error when creating the *<%mm>*:

The class "*<class name>*" could not be created!

Super class: "*<super class name>*"

Error code: *<error code>*

ADONIS will be closed!

Cause: The configuration of your ADONIS database contains errors.

Action: Contact your ADONIS administrator.

[astdcfg-02]

Message: Error when creating the *<%mm>*:

The relation class "*<relation class name>*" could not be created!

From class: "*<starting class name>*"

To class: "*<target class name>*"

Error code: *<error code>*

ADONIS will be closed!

Cause: The configuration of your ADONIS database contains errors.

Action: Contact your ADONIS administrator.

[astdcfg-03]

Message: Error when creating the *<%mm>*:

The attribute "*<attribute name>*" could not be created!

Owner: "*<owner name>*"

Error code: *<error code>*

ADONIS will be closed!

Cause: The configuration of your ADONIS database contains errors.

Action: Contact your ADONIS administrator.

[astdcfg-04]

Message: Error when creating the *<%mm>*:

No value could be assigned to the attribute "*<attribute name>*"!

Owner: "*<owner name>*"

Value: "*<value>*"

Error code: *<error code>*

ADONIS will be closed!

Cause: The configuration of your ADONIS database contains errors.

Action: Contact your ADONIS administrator.

[astdcfg-05]

Message: Error when creating the *<%mm>*:

No value could be assigned to the facet "*<facet name>*"!

Attribute: "*<attribute name>*"

Value: "*<value>*"

Error code: *<error code>*

ADONIS will be closed!

Cause: The configuration of your ADONIS database contains errors.

Action: Contact your ADONIS administrator.

[asysbox-01]

Message: System user groups could not be loaded.

Cause: When loading system user groups from the system an error occurred.

Action: Please try again and if necessary contact your ADONIS consultant.

[asysbox-02]

Message: System users could not be loaded.

Cause: When loading system users from the system an error occurred.

Action: Please try again and if necessary contact your ADONIS consultant.

[asysbox-03]

Message: System users of the system user group "*<system user group name>*" could not be loaded.

Cause: When loading system users from the system group an error occurred.

Action: Please try again and if necessary contact your ADONIS consultant.

[asysbox-04]

Message: The domain "*<Domain name>*" could not be initialised.

Cause: Within the current system environment no domain could be found or you do not have access rights for the existing domains.

Action: Log off from the system and login to a domain of which your computer is a member.

<u>**Note:**</u> You CANNOT load domains when you are logged in at the local domain of your computer or your computer is not a member of a work group and therefore not member of a domain.

If necessary please contact your ADONIS consultant.

[asysbox-05]

Message: The list of system domains could not be loaded.

Cause: Within the current system environment no domain could be found or you do not have access rights for the existing domains.

Action: Log off from the system and login to a domain your computer is a member.

    Note: You CANNOT load domains when you are logged in locally or your computer is not a member of a work group and therefore not a member of a domain.

    If necessary please contact your system administrator.

[ausmgt-01]

Message: No application library was selected!

Cause: You did not select any of the application libraries listed.

Action: Select by mouse-click the application library you wish to assign to the ADONIS user.

[ausmgt-02]

Message: Passwords are not consistent!

Cause: Your entry for re-confirming the password does not match the password entry.

Action: Enter the same password twice.

    (**Note:** The system distinguishes between capital and small letters!)

[ausmgt-03]

Message: Password must not be shorter than three characters!

Cause: You entered a password shorter than three characters.

Action: Enter a password which is at least three characters in length.

[ausmgt-05]

Message: The user already exists!

Cause: You tried to add an ADONIS user who is already stored in the ADONIS database.

Action: Enter a different user name for the ADONIS user to be added.

[ausmgt-13]

Message: The user "*<user name>*" does not exist!

Cause: The ADONIS user has possibly been already deleted by another ADONIS administrator.

Action: Close the error message - the user list will automatically be updated - and repeat the action. Should the error appear again, close ADONIS and restart the application. Should the error appear again, contact your ADONIS administrator.

[ausmgt-14]

Message: The user name must have at least three characters.

Cause: You entered a password shorter than three characters.

Action: Enter a password which is at least three characters in length.

[ausmgt-15]

Message: The password contains restricted characters!

Please only use alphanumerical characters (a-z, A-Z, 0-9).

Blanks and the symbols ! # $ % & ( ) * + , - . / : ; < = > ? @ [ ] ^ _ | { } as well as umlauts and ß must not be used.

Cause: You entered illegal restricted characters or umlauts when entering the password.

Action: The characters permitted for the password are only the numbers **0 to 9**, the letters **a to z** and **A to Z**.

Blanks and the symbols **! # $ % & ( ) * + , - . / : ; < = > ? @ [ ] ^ _ | { }** must not be used.

[ausmgt-16]

Message: Invalid filename entered!

Cause: You have entered an invalid filename for user import or export.

Action: Check your input.

[ausmgt-17]

Message: No filename entered!

Cause: No filename has been entered for user import or export.

Action: Enter a filename.

[ausmgt-18]

Message: The new user to be created has not been assigned to a user group!

 or

Message: The user has been assigned to no user group!

 or

Message: The selected user has not been assigned to a user group!

Cause: Each user must be assigned to a user group.

Action: Assign the user(s) (button "user groups").

[ausmgt-19]

Message: File *<filename>* does not exist!!

Cause: You entered a file which does not exist for the user import.

Action: Check the path and the file name entered and repeat the action.

[ausmgt-20]

Message: The user name entered is invalid.

                 User names cannot contain the figures **;** **@** and **,**.

Cause: You have entered unallowed figures for the user name.

Action: Correct the user name accordingly.

[ausmgt-21]

Message: Some tasks could not be processed as the user list has been changed by another user.

                 The user list has been updated.

Cause: Another ADONIS administrator has changed the user list and therefore your tasks could not be performed.

Action: The user list has been updated accordingly. Please repeat the intended tasks.

[ausmgt-22]

Message: The character '*<Invalid character>*' is not allowed in user names.

                 Allowed characters include the alphanumerics a-z A-Z 0-9 as well as blanks and the characters '.' (full stop), '-' (hyphen) and '_' (underscore).

Cause: You entered a password with an invalid character.

Action: Enter a password using the following permitted characters only:

                 Alphanumerics a-z, A-Z, and 0-9 as well as blanks and the characters '.' (full stop), '-' (hyphen) and '_' (underscore).

[avedbox-01]

Message: Figures chain "*<Text>*" not found!

Cause: The figures chain searched for does not appear in the text field.

Action: The search has possibly not been carried out from the beginning of the text. If necessary place the cursor at the beginning of the text field before starting the search.

[averdate-01]

Message: Internal error: the version handler has not been initialised.

Cause: An internal error has occurred.

Action: Contact your ADONIS administrator.

[averdate-02]

Message: A syntactical error has occurred when parsing the expression:

<LEO expression>

Cause: The value of the library attribute "versioning format" contains errors. The specified LEO expression contains syntactical errors. Pay attention to the appropriate [aleo] error message.

Action: Correct the value of the library attribute "versioning format" and pay attention to the specified [aleo] error message or contact your ADONIS administrator.

[averdate-04]

Message: The key word "VERSIONING" has not been found.

Cause: The first key word of the LEO expressions is not "VERSIONING".

Action: Correct the value of the library attribute "versioning format" by inserting the key word "VERSIONING" at the first position or contact your ADONIS administrator.

[averdate-05]

Message: An invalid key word has been used.

Cause: The LEO expression contains an invalid key word.

Action: Correct the value of the library attribute "versioning format" by deleting all invalid key words or contact your ADONIS administrator. Valid key words are "VERSIONING", "START_DATE", "TEXT_FIELD", "DAY_FIELD", "MONTH_FIELD" and "YEAR_FIELD".

[averdate-06]

Message: The start date can only be defined once.

Cause: The LEO expression contains the key word "START_DATE" more than once.

Action: Correct the value of the library attribute "versioning format" by deleting all unnecessary "START_DATE" key words or contact your ADONIS administrator.

[averdate-07]

Message: The day field can only be defined once.

Cause:	The LEO expression contains the key word "DAY_FIELD" more than once.
Action:	Correct the value of the library attribute "versioning format" by deleting all unnecessary "DAY_FIELD" key words or contact your ADONIS administrator.

[averdate-08]

Message:	The month field can only be defined once.
Cause:	The LEO expression contains the key word "MONTH_FIELD" more than once.
Action:	Correct the value of the library attribute "versioning format", by removing all unnecessary "MONTH_FIELD" key words or contact your ADONIS administrator.

[averdate-09]

Message:	The year field can only be defined once.
Cause:	The LEO expression contains the key word "YEAR_FIELD" more than once.
Action:	Correct the value of the library attribute "versioning format", by removing all unnecessary "YEAR_FIELD" key words or contact your ADONIS administrator.

[averdate-10]

Message:	The value of the day attribute in the start date is too small. It is only allowed within the area *<1>* to *<31>*.
Cause:	An error has occurred in the key word "START_DATE": The value of the attribute "day" is too small.
Action:	Correct the value of the library attribute "versioning format", by specifying the attribute "day" of the key word "START_DATE" with a valid value between 1 and 31 or contact your ADONIS administrator.

[averdate-11]

Message:	The value of the day attribute in the start date is too high. It is only allowed within the area *<1>* to *<31>*.
Cause:	An error has occurred in the key word "START_DATE": The value of the attribute "day" is too high.
Action:	Correct the value of the library attribute "versioning format", by specifying the attribute "day" of the key word "START_DATE" with a valid value between 1 and 31 or contact your ADONIS administrator.

[averdate-12]

Message:	The value of the month attribute in the start date is too small. It is only allowed within the area *<1>* to *<12>*.

Cause: An error has occurred in the key word "START_DATE": the value of the attribute "month" is too small.

Action: Correct the value of the library attribute "versioning format" by specifying the attribute "month" of the key word "START_DATE" with a valid value between 1 and 12 or contact your ADONIS administrator.

[averdate-13]

Message: The value of the month attribute in the start date is too high.

It is only allowed within the area <*1*> to <*12*>.

Cause: An error has occurred in the key word "START_DATE": the value of the attribute "month" is too high

Action: Correct the value of the library attribute "versioning format", by specifying the attribute "month" of the key word "START_DATE" with a valid value between 1 and 12 or contact your ADONIS administrator.

[averdate-14]

Message: The value of the year attribute in the start date is too small.

It is only allowed within the area <*0*> to <*9999*>.

Cause: An error has occurred in the key word "START_DATE": the value of the attribute "year" is too small.

Action: Correct the value of the library attribute "versioning format", by specifying the attribute "year" of the key word "START_DATE" with a valid value between 0 and 9999 or contact your ADONIS administrator.

[averdate-15]

Message: The value of the year attribute in the start date is too high.

It is only allowed within the area <*0*> to <*9999*>.

Cause: An error has occurred in the key word "START_DATE": the value of the attribute "year" is too high.

Action: Correct the value of the library attribute "versioning format", by specifying the attribute "year" of the key word "START_DATE" with a valid value between 0 and 9999 or contact your ADONIS administrator.

[averdate-16]

Message: The minimum value of the day field is greater than the maximum value.

Cause: An error has occurred in the key word "DAY_FIELD": the value of the attribute "minimum" is greater than the value of the attribute "maximum".

Action: Correct the value of the library attribute "versioning format" by specifying in key word "DAY_FIELD" the value of the attribute "minimum" with a value, which is smaller than the value of the attribute "maximum" or contact your ADONIS administrator.

[averdate-17]

Message: The minimum value of the month field is greater than the maximum value.

Cause: An error has occurred in the key word "MONTH_FIELD": the value of the attribute "minimum" is greater than the value of the attribute "maximum".

Action: Correct the value of the library attribute "versioning format", by specifying in key word "MONTH_FIELD" the value of the attribute "minimum" with a value, which is smaller than the value of the attribute "maximum" or contact your ADONIS administrator.

[averdate-18]

Message: The minimum value of the year field is greater than the maximum value.

Cause: An error has occurred in the key word "YEAR_FIELD": the value of the attribute "minimum" is greater than the value of the attribute "maximum".

Action: Correct the value of the library attribute "versioning format", by specifying in key word "YEAR_FIELD" the value of the attribute "minimum" with a value, which is smaller than the value of the attribute "maximum" or contact your ADONIS administrator.

[averdate-19]

Message: The default value of the day field is too small.

It is only allowed within the area *<1>* to *<31>*.

Cause: An error has occurred in the key word "DAY_FIELD": the value of the attribute "default" is either not in the area 1 to 31 or the value is smaller than the value of the attribute "minimum".

Action: Correct the value of the library attribute "versioning format" by specifying the value of the attribute "default" in the key word "DAY_FIELD" with a value, which is both between the values of the attributes "minimum" and "maximum" and between 1 and 31 or contact your ADONIS administrator.

[averdate-20]

Message: The default value of the day field is too high.

It is only allowed within the area *<1>* to *<31>*.

Cause: An error has occurred in the key word "DAY_FIELD": the value of the attribute "default" is either not in the area 1 to 31 or the value is higher than the value of the attribute "maximum".

Action: Correct the value of the library attribute "versioning format" by specifying the value of the attribute "default" in the key word "DAY_FIELD" with a value, which is both between the values of the attributes "minimum" and "maximum" and between 1 and 31 or contact your ADONIS administrator.

[averdate-21]

Message: The minimum value of the day field is too small.

It is only allowed within the area *<1>* to *<31>*.

Cause: An error has occurred in the key word "DAY_FIELD": the value of the attribute "minimum" is too small.

Action: Correct the value of the library attribute "versioning format" by specifying the value of the attribute "minimum" in the key word "DAY_FIELD" with a value between 1 and 31 or contact your ADONIS administrator.

[averdate-22]

Message: The minimum value of the day field is too high.

It is only allowed within the area *<1>* to *<31>*.

Cause: An error has occurred in the key word "DAY_FIELD": the value of the attribute "minimum" is too high.

Action: Correct the value of the library attribute "versioning format" by specifying the value of the attribute "minimum" in the key word "DAY_FIELD" with a value between 1 and 31 or contact your ADONIS administrator.

[averdate-23]

Message: The maximum value of the day field is too small.

It is only allowed within the area *<1>* to *<31>*.

Cause: An error has occurred in the key word "DAY_FIELD": the value of the attribute "maximum" is too small.

Action: Correct the value of the library attribute "versioning format" by specifying the value of the attribute "maximum" in the key word "DAY_FIELD" with a value between 1 and 31 or contact your ADONIS administrator.

[averdate-24]

Message: The maximum value of the day field is too high.

It is only allowed within the area *<1>* to *<31>*.

Cause: An error has occurred in the key word "DAY_FIELD": the value of the attribute "maximum" is too high.

Action: Correct the value of the library attribute "versioning format" by specifying the value of the attribute "maximum" in the key word "DAY_FIELD" with a value between 1 and 31 or contact your ADONIS administrator.

[averdate-25]

Message: The default value of the month field is too small.

It is only allowed within the area *<1>* to *<12>*.

Cause: An error has occurred in the key word "MONTH_FIELD": the value of the attribute "default" is either not in the area 1 to 12 or the value is smaller than the value of the attribute "minimum".

Action: Correct the value of the library attribute "versioning format" by specifying the value of the attribute "default" in the key word "MONTH_FIELD" with a value, which is both between the values of the attributes "minimum" and "maximum" and between 1 and 12 or contact your ADONIS administrator.

[averdate-26]

Message: The default value of the month field is too high.

It is only allowed within the area *<1>* to *<12>*.

Cause: An error has occurred in the key word "MONTH_FIELD": the value of the attribute "default" is either not in the area 1 to 12 or the value is greater than the value of the attribute "maximum".

Action: Correct the value of the library attribute "versioning format" by specifying the value of the attribute "default" in the key word "MONTH_FIELD" with a value, which is both between the values of the attributes "minimum" and "maximum" and between 1 and 12 or contact your ADONIS administrator.

[averdate-27]

Message: The minimum value of the month field is too small.

It is only allowed within the area *<1>* to *<12>*.

Cause: An error has occurred in the key word "MONTH_FIELD": the value of the attribute "minimum" is too small.

Action: Correct the value of the library attribute "versioning format", by specifying in the key word "MONTH_FIELD" the value of the attribute "minimum" with a value between 1 and 12 or contact your ADONIS administrator.

[averdate-28]

Message: The minimum value of the month field is too high.

It is only allowed within the area *<1>* to *<12>*.

Cause: An error has occurred in the key word "MONTH_FIELD": the value of the attribute "minimum" is too high.

Action: Correct the value of the library attribute "versioning format", by specifying in the key word "MONTH_FIELD" the value of the attribute "minimum" with a value between 1 and 12 or contact your ADONIS administrator.

[averdate-29]

Message: The maximum value of the month field is too small.

It is only allowed within the area *<1>* to *<12>*.

Cause: An error has occurred in the key word "MONTH_FIELD": the value of the attribute "maximum" is too small.

Action: Correct the value of the library attribute "versioning format", by specifying in the key word "MONTH_FIELD" the value of the attribute "maximum" with a value between 1 and 12 or contact your ADONIS administrator.

[averdate-30]

Message: The maximum value of the month field is too high.

It is only allowed within the area *<1>* to *<12>*.

Cause: An error has occurred in the key word "MONTH_FIELD": the value of the attribute "maximum" is too high.

Action: Correct the value of the library attribute "versioning format", by specifying in the key word "MONTH_FIELD" the value of the attribute "maximum" with a value between 1 and 12 or contact your ADONIS administrator.

[averdate-31]

Message: The default value of the year field is too small.

It is only allowed within the area *<0>* to *<9999>*.

Cause: An error has occurred in the key word "YEAR_FIELD": the value of the attribute "default" is either not within the area 0 to 9999 or the value is smaller than the value of the attribute "minimum".

Action: Correct the value of the library attribute "versioning format", by specifying in the key word "YEAR_FIELD" the value of the attribute "default" with a value which is both between the values of the attributes "minimum" and "maximum" and between 0 and 9999 or contact your ADONIS administrator.

[averdate-32]

Message: The default value of the year field is too high.

It is only allowed within the area *<0>* to *<9999>*.

Cause: An error has occurred in the key word "YEAR_FIELD": the value of the attribute "default" is either not within the area 0 to 9999 or the value is greater than the value of the attribute "maximum".

Action: Correct the value of the library attribute "versioning format", by specifying in the key word "YEAR_FIELD" the value of the attribute "default" with a value which is both between the values of the attributes "minimum" and "maximum" and between 0 and 9999 or contact your ADONIS administrator.

[averdate-33]

Message: The minimum value of the year field is too small.

It is only allowed within the area *<0>* to *<9999>*.

Cause: An error has occurred in the key word "YEAR_FIELD": the value of the attribute "minimum" is too small.

Action: Correct the value of the library attribute "versioning format", by specifying in the key word "YEAR_FIELD" the value of the attribute "minimum" with a value between 0 and 9999 or contact your ADONIS administrator.

[averdate-34]

Message: The minimum value of the year field is too high.

It is only allowed within the area *<0>* to *<9999>*.

Cause: An error has occurred in the key word "YEAR_FIELD": the value of the attribute "minimum" is too high.

Action: Correct the value of the library attribute "versioning format", by specifying in the key word "YEAR_FIELD" the value of the attribute "minimum" with a value between 0 and 9999 or contact your ADONIS administrator.

[averdate-35]

Message: The maximum value of the year field is too small.

It is only allowed within the area *<0>* to *<9999>*.

Cause: An error has occurred in the key word "YEAR_FIELD": the value of the attribute "maximum" is too small.

Action: Correct the value of the library attribute "versioning format", by specifying in the key word "YEAR_FIELD" the value of the attribute "maximum" with a value between 0 and 9999 or contact your ADONIS administrator.

[averdate-36]

Message: The maximum value of the year field is too high.

It is only allowed within the area *<0>* to *<9999>*.

Cause: An error has occurred in the key word "YEAR_FIELD": the value of the attribute "maximum" is too high.

Action: Correct the value of the library attribute "versioning format", by specifying in the key word "YEAR_FIELD" the value of the attribute "maximum" with a value between 0 and 9999 or contact your ADONIS administrator.

[averdate-37]

Message: The value of the day field is too small.

It is only allowed within the area *<1>* to *<31>*.

Cause: The day of the specified date is either smaller than the value of the attribute "minimum" of the key word "DAY_FIELD" or less than 1.

Action: Change the date, so that the day is between Minimum and Maximum or between 1 and 31.

[averdate-38]

Message: The value of the day field is too high.

It is only allowed within the area *<1>* to *<31>*.

Cause: The day of the specified date is either greater than the value of the attribute "maximum" of the key word "DAY_FIELD" or greater than 1.

Action: Change the date, so that the day is between Minimum and Maximum or between 1 and 31.

[averdate-39]

Message: The value of the month field is too small.

It is only allowed within the area *<1>* to *<12>*.

Cause: The day of the specified date is either smaller than the value of the attribute "minimum" of the key word "MONTH_FIELD" or smaller than 1.

Action: Change the date, so that the month is between Minimum and Maximum or between 1 and 12.

[averdate-40]

Message: The value of the month field is too high.

It is only allowed within the area *<1>* to *<12>*.

Cause: The day of the specified date is either greater than the value of the attribute "maximum" of the key word "MONTH_FIELD" or greater than 1.

Action: Change the date, so that the month is between Minimum and Maximum or between 1 and 12.

[averdate-41]

Message: The value of the year field is too small.

It is only allowed within the area *<0>* to *<9999>*.

Cause: The day of the specified date is either smaller than the value of the attribute "minimum" of the key word "YEAR_FIELD" or smaller than 0.

Action: Change the date, so that the year is between Minimum and Maximum or between 0 and 9999.

[averdate-42]

Message: The value of the year field is too high.

It is only allowed within the area *<0>* to *<9999>*.

Cause: The day of the specified date is either greater than the value of the attribute "maximum" of the key word "YEAR_FIELD" or greater than 9999.

Action: Change the date, so that the year is between Minimum and Maximum or between 0 and 9999.

[averdate-43]

Message: The internal display of the date is not correct.

Cause: An internal error has occurred: the format of the date is invalid.

Action: Contact your ADONIS administrator.

[averdate-44]

Message: The internal display of the date contains a day although no day field is defined.

Cause: An internal error has occurred: the format of the date is invalid.

Action: Contact your ADONIS administrator.

[averdate-45]

Message: The internal display of the date contains a month although no month field is defined.

Cause: An internal error has occurred: the format of the date is invalid.

Action: Contact your ADONIS administrator.

[averdate-46]

Message: The internal display of the date contains a year although no year field is defined.

Cause: An internal error has occurred: the format of the date is invalid.

Action: Contact your ADONIS administrator.

[averdate-47]

Message: The specified date cannot be increased.

Cause: The specified date is the last possible date.

Action: Use an earlier date since this is the last possible date or contact your ADONIS administrator.

[averdate-48]

Message: Error when parsing the string:

The content of the text field does not correspond with the defined content.

Cause: A string has been found which does not correspond to the one in the library attribute "versioning format".

Action: Adapt the string that it corresponds with the format defined in the library attribute "versioning format". Pay attention to the order! If necessary, contact your ADONIS administrator.

[averdate-49]

Message: Error when parsing the string:

The month name is not valid.

Cause: The name of a month is expected but could not be found.

Action: Possible month names are "January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November" and "December".

[averdate-50]

Message: Error when parsing the string:

The sting contains additional characters.

Cause: The string contains excess characters.

Action: Adapt the string so that it corresponds with the format defined in the library attribute "versioning format" or contact your ADONIS administrator.

[averdate-51]

Message: Error when parsing the string:

The combination of day and month is invalid.

Cause: You have attempted to use an invalid combination of day and month.

Action: Change the string so that it only contains valid combinations of day and month, invalid combinations include "29" and "2" or "31" and "4".

[averdate-52]

Message: Error when parsing the string:

No number has been found at the current place.

Cause: A number is expected in the string but could not be found.

Action: Adapt the string so that it corresponds with the format defined in the library attribute "versioning format" or contact your ADONIS administrator.

[averdate-53]

Message: Error when parsing the string:

The number found is too small.

Cause: A number found in the string is too small.

Action: Adapt the string so that it corresponds with the format defined in the library attribute "versioning format" or contact your ADONIS administrator.

[averdate-54]

Message: Error when parsing the string:

The number found is too high.

Cause: A number found in the string is too high.

Action: Adapt the string so that it corresponds with the format defined in the library attribute "versioning format" or contact your ADONIS administrator.

[averdate-55]

Message: Error while parsing string:

Between two dynamic values at least one TEXT field must exist.

Cause: When defining the library attribute "Versioning format" two dynamic fields (for day, month or year) have been entered consecutively.

Action: Change the definition by inserting a text field between the two dynamic fields or contact your ADONIS administrator.

[awrjpeg-01]

Message: *<Error-specific Message>*

Cause: When generating the graphic file, an unexpected error was encountered.

Action: Retry.

[awrjpeg-02]

Message: Too little space!

Cause: Not enough main memory is available.

Action: Close a number of applications and try again.

[awrjpeg-03]

Message: Error on opening file "*filename*".

Cause: The file specified could not be opened with write-access.

Action: Ensure that enough space exists in the directory to which you are writing and that you have access to update that directory.

[awrjpeg-04]

Message: Error while closing file.

Cause: An error occurred while closing the graphics file.

Action: Ensure you have enough space in the directory to which you are writing. If this directory is on the network, ensure that you are still connected. Try again.

[svxwins-01]

Message: No file name has been specified!

Cause: A file name must be specified for the selected action.

Action: Enter a valid file name.

[svxwins-02]

Message: The file does not exist!

Cause: You have attempted to open a file to read which does not exist.

Action: Enter a valid file name.

[svxwins-03]

Message: An error has occurred during the storage!

Cause: An error has occurred during the storage of the file.

Action: There may be not enough space on the data medium or the specified path is not valid. Check the data medium or the path.

Part IV
Appendix

1. Starting ADONIS

The login to ADONIS depends on the settings of the ADONIS database:

● exclusively as ADONIS user (see chap. 1.1, p. 528) or

● alternatively as ADONIS user or as a system user with Single-Sign-on functionality (see chap. 1.2, p. 529).

1.1 Start ADONIS

Start the Administration Toolkit by clicking on the program icon on your desktop or by selecting the program from the "Start" menu.

Figure 352: ADONIS login

Enter your user name, your password, the name of the ADONIS database with which you intend to work and click on the button "Login".

> **Hint:** Depending on the settings of your ADONIS installation it might be possible to select the name of the database from a list of possible values in a drop-down list. The availability of such a list is indicated by the symbol ⌄ at the end of the input field.

After a successful login the Administration Toolkit will be started.

Additionally the ADONIS application can be started from a **command line window** . In this case you will first have to enter the directory where ADONIS has been installed to.

By entering "aadma" for the ADONIS Administration Toolkit the ADONIS application will be started.

The following parameters can be used:

-u<*User name*>   replace <*User name*> with the name of the ADONIS/system user.

-p<*Password*>   replace <*Password*> with the password of the ADONIS/system user.

-d<*DB-Name*>   replace <*DB name*> with the name of the ADONIS database.

-f<*font type name>*	replace <*font type name>* with the name of the font to be used in ADONIS.
	The default font for Windows is "MS Sans Serif".
-h<*font size>*	replace <*font size>* with a number between 4 and 48 for the size of the font in ADONIS.
	The default for the font size is 12pt.
-x	causes that the ADONIS login screen to be shown despite the fact that parameters for the user name (**-u**), the password (**-p**) and the database name (**-d**) are shown.
-winlogin	determines the login as a system user.
	Note: To use the parameter -winlogin the Single-Sign-on functionality must be installed.
-no_login_screen	determines that in the case of an unsuccessful login the login screen will not be displayed after the error message has been shown.
	Note: An unsuccessful login means that the parameter for the user name (**-u**), the password (**-p**) and the database name (**-d**) have been stated but at least one of the parameters was incorrect.
-no_printer_warning	causes that the error message "No default printer installed" <u>will not be</u> displayed in the case of no default printer being installed.

Hint: If you pass on the user name (-u), the password (-p) and the database name (-d) correctly as parameters the ADONIS login screen will not be shown.

ATTENTION: Should the user name, password, database name, or font type names contain spaces the entire parameter has to be put within quotation marks (e.g. "-uUser 1" for the user "User 1").

ATTENTION: The user name and password are case sensitive!

Examples:

```
-uAdmin -ppassword -dadonisdb
```

logs in the user "Admin" with the password "password" to the ADONIS database "adonisdb".

```
-uAdmin -ppassword -dadonisdb -x
```

shows the ADONIS login screen (see fig. 352) where the field "User name" contains the value "Admin", the field "Password" the encrypted string "password" and the field "Database name" the value "adonisdb".

```
"-uUser 1" "-pUser 1" -ddatabase -h14
```

logs in the user "User 1" with the password "User 1" to the ADONIS database "database" using a font size of 14 pt in the ADONIS application.

1.2   Start ADONIS (Single-Sign-on)

Hint: The functionality for Single-Sign-on, i.e. the login to ADONIS as a system user without providing your username and password is only possible with Windows XP/2000/NT 4.0 and has to be enabled for the respective ADONIS database. For further information please contact your ADONIS consultant.

Start the Administration Toolkit by clicking on the program icon on your desktop or by selecting the program from the "Start" menu.

Figure 353: ADONIS login (Single-Sign-on)

To log into ADONIS you can choose from the following options:

1. as the **current system user**,

2. as a **different system user** or,

3. as a **ADONIS user**.

As a **current system user** enter the name of the ADONIS database you want to work with and select the option "as system user" and click on the button "Login" (it is not necessary to provide a user name or password).

As a **different system user** enter your system user name, your system password and the name of the ADONIS database you want to work with. Then select the option "as system user" and click on the button "Login".

> **Hint:** The log-in as a different system user is only possible within the current domain.

> **Hint:** To log-in as a system user you will have to arrange for your system user information to be imported into the ADONIS user management. Please contact your ADONIS administrator if necessary.

As **ADONIS user** enter your ADONIS user name, your ADONIS password and the name of the ADONIS database you want to work with and select the options "as ADONIS user" and click on the button "Login".

After a successful login the Administration Toolkit will be started.

Additionally the ADONIS application can be started from a **command line window** . Therefore change to the directory where ADONIS has been installed to.

By entering "aadma" for the ADONIS Administration Toolkit the ADONIS application will be started.

The following parameters can be used:

-u<*User name*>   replace <*User name*> with the name of the ADONIS/system user.

-p<*Password*>   replace <*Password*> with the password of the ADONIS/system user.

-d<*DB-Name*>	replace <*DB name*> with the name of the ADONIS database.
-f<*font type name*>	replace <*font type name*> with the name of the font to be used in ADONIS.
	The default font for Windows is "MS Sans Serif".
-h<*font size*>	replace <*font size*> with a number between 4 and 48 for the size of the font in ADONIS.
	The default for the font size is 12pt.
-x	causes the ADONIS login screen to be shown despite the fact that parameters for the user name (**-u**), the password (**-p**) and the database name (**-d**) are shown.
-winlogin	determines the login with as a system user.
	Note: To use the parameter -winlogin the Single-Sign-on functionality must be installed.
-no_login_screen	determines that in the case of an unsuccessful login the login screen will not be displayed after the error message has been shown.
	Note: An unsuccessful login means that the parameter for the user name (**-u**), the password (**-p**) and the database name (**-d**) have been stated but at least one of the parameters was incorrect.
-no_printer_warning	causes that the error message "No default printer installed" <u>will not be</u> displayed in the case of no default printer being installed.

Hint: If you pass on the user name (-u), the password (-p) and the database name (-d) correctly as parameters the ADONIS login screen will not be shown.

ATTENTION: Should the user name, password, database name, or font type names contain spaces the entire parameter has to be put within quotation marks (e.g. "-uUser 1" for the user "User 1").

ATTENTION: The user name and password are case sensitive!

Examples:

```
-dadonisdb -winlogin
```

logs in the current system user to the ADONIS database "adonisdb".

```
-uAdministrator -ppassword -dadonisdb -winlogin
```

logs in a different system user "Administrator" with the system password "password" to the ADONIS-database "adonisdb".

```
-uAdmin -ppassword -dadonisdb
```

logs in the user "Admin" with the password "password" to the ADONIS database "adonisdb".

```
-uAdmin -ppassword -dadonisdb -x
```

shows the ADONIS login screen (see fig. 352) where the field "User name" contains the value "Admin", the field "Password" the encrypted string "password" and the field "Database name" the value "adonisdb".

```
"-uUser 1" "-pUser 1" -ddatabase -h14
```

logs in the user "User 1" with the password "User 1" to the ADONIS database "database" using a font size of 14 pt in the ADONIS application.

2.   Closing ADONIS

To close the Administration Toolkit, double click on the "Exit" button, (to the left on the window bar) or click on the .

Within each component of the ADONIS Administration Toolkit, it is also possible to quit ADONIS by selecting the menu item "Exit ADONIS" from the menu list on the left.

3.   Status Information

Select the menu item "Status" from the "Help" menu in the Administration Toolkit in order to check the current status of ADONIS. This will cause the window "ADONIS: status information" to be displayed (see fig. 354).
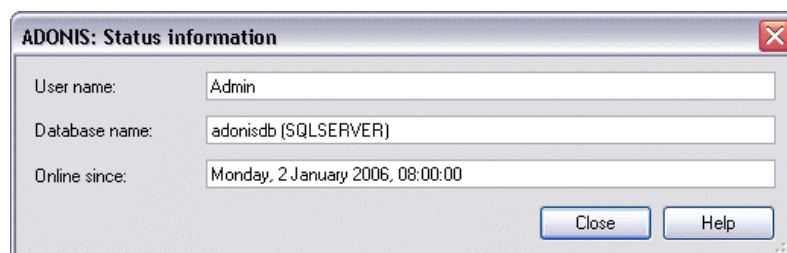
Figure 354: ADONIS status information in the Administration Toolkit

The user by which you are currently connected to ADONIS is shown in the field "User name". The database name is shown in the "Database name" field. Finally, the field "Online since" shows the date and time at which you first logged on.

4. Regular Expressions

A regular expression is a specific search text, in which special strings are used in order to make it applicable to text patterns in a file. A regular expression consists of one or more <u>non-empty</u> *branches* (=strings). These branches are connected or separated by the character "|".

Example: *branch* | *branch* | *branch*;

If a string within the text searched meets the criteria defined by a branch, an appropriate result is displayed (see fig. 355).
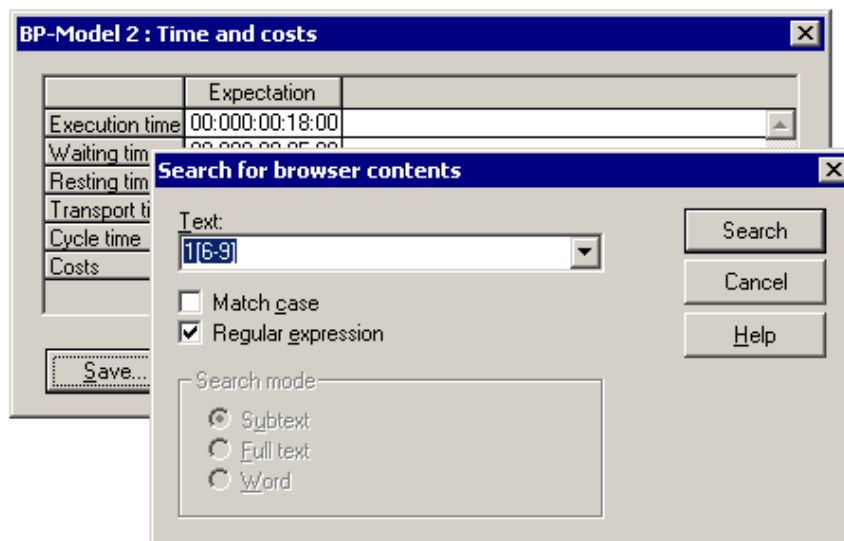
Figure 355: Search using regular expressions

Regular expression - meta-sign	Description
* ? . | () [] + \ and string(s) to be searched for	Search for all characters (letters, figures, other) except *** ? . | () [] + ** (=meta-signs or operators); the special function of the above signs is cancelled by putting a **** in front. **Example:** Search with *a*ctivity* for a*ctivity.
.	Instead of the **.** any character may occur in the result. **Example:** *w.rd* finds *ward, w4rd, wgrd, word,* etc.
[]	One of the characters within the brackets must be contained in the result, but the characters within the brackets may not be combined in the result. Please note that the order of the characters searched is of no importance when using square brackets (in contrast to round brackets). **Example:** *w[aeiou]rd* finds *ward, werd, wird, word, wurd* but, for example, not *waerd*.
[^]	The "caret - ^" within square brackets excludes the following string from the search result. **Example:** *wo[^0-9]rd* cannot find a result like *wo3rd*.
^	The result contains those lines starting with the character (combination) following the caret. **Example:** *^word* finds all lines starting with *word*.
$	The result shows all lines ending with the character (combination) in front of the "$". **Example:** *less$* finds all lines ending with *less*. If a line ends with *less.$*, the full stop must be cancelled by *less \\.$*.

Regular expression - meta-sign	Description

The three following operators always refer to the character immediately preceding the operator:

?	The character preceding the "?" occurs either once or not at all (? represents the quantity 0). **Example:** *wo?rd* finds the result *wrd* or *word*.
*	The character preceding the "*" occurs either not at all (* represents the quantity 0) or an indefinite number of times. The search result for *wo*rd* is either *wrd* or *word, woord, wooord* etc.
+	The character preceding the "+" occurs either once or an indefinite number of times. The search result for *wo+rd* is either *word* or *woord, wooord, woooord, ...*
()	Round brackets serve the grouping of character (sequences). The search result refers to the expression within the brackets. **Example:** *(wo)+rd* finds *wowoword* or *wwwword*, but not *wooord*!
|	You can combine search criteria by describing several regular expressions. These *branches* are connected or separated by the character |. **Example:** *(w.rd | less | .)s*; a string is displayed if it meets the criteria of at least one *branch*, e.g. *words, wards, less.s*

5. ADONIS Attribute Types

ADONIS offers you the following types of attributes:

- Attribute Profile Reference (ATTRPROFREF) (see chap. 5.1, p. 536)
- Calendar (see chap. 5.9, p. 538)
- Date (see chap. 5.5, p. 537)
- Date and Time (DATETIME) (see chap. 5.6, p. 537)
- Double (see chap. 5.8, p. 538)
- Enumeration (see chap. 5.2, p. 536)
- Enumeration List (see chap. 5.3, p. 537)
- Expression (see chap. 5.4, p. 537)
- Integer (see chap. 5.7, p. 538)
- Interref (see chap. 5.12, p. 539)
- Long String (see chap. 5.10, p. 538)
- Program Call (see chap. 5.11, p. 539)
- Record (see chap. 5.13, p. 539)
- String (see chap. 5.14, p. 539)
- Time (see chap. 5.15, p. 539)

Hint: The attribute types are specified during the definition of the application library.

5.1 Attribute Profile Reference (ATTRPROFREF)

An attribute of the type "Attribute profile reference" (ATTRPROFREF) is defined for references to attribute profiles (see chap. 8., p. 77). An attribute profile reference attribute is made of the attributes defined in the referenced attribute profile.

The standard value for attributes of this type depends on the attribute types defined in the referenced attribute profile.

Hint: An attribute profile can contain attributes of all ADONIS attribute types except the type "attribute profile reference".

The value of an attribute of type "Attribute profile reference" will be aligned left by default in the ADONIS browser (see chap. 5., p. 44).

5.2 Enumeration (ENUMERATION)

An attribute of the type "Enumeration" is characterised by a defined set of values. An "Enumeration" attribute has exactly one value of this set.

The standard value of this type is specified in the library definition.

> **Hint:** The set of values of an "Enumeration" attribute can be extended in the library configuration (Library Management in the Administration Toolkit).

The value of an attribute of type "Enumeration" will be aligned left by default in the ADONIS browser (see chap. 5., p. 44).

5.3 Enumeration List (ENUMERATIONLIST)

An attribute of the type "Enumeration list" is characterised by a defined set of values. An "ENUMERATIONLIST" attribute has either no value, a single or several values of this set.

The standard value of this type is specified in the library definition.

> **Hint:** The set of values of an "Enumeration List" attribute can be extended in the library configuration (Library Management in the Administration Toolkit).

The value of an attribute of type "Enumeration list" will be aligned left by default in the ADONIS browser (see chap. 5., p. 44).

5.4 Expression (EXPRESSION)

An attribute of type "Expression" is characterised by the possibility to determine its value via a flexible calculation rule.

The standard value for the attribute of this type is defined in the library definition.

> **Hint:** You can define the possibility of the value entry of an expresion attribute in the library configuration (library administration in Administration Toolkit).

5.5 Date (DATE)

An attribute of the type "date" is defined for the entry of a date in format "YYYY:MM:DD" (year:month:day).

The standard value for attributes of this type is "2002:01:01" or a value defined in the application library.

The value of an attribute of type "Date" will be aligned right by default in the ADONIS browser (see chap. 5., p. 44).

5.6 Date and Time (DATETIME)

An attribute of type "Date time" is defined for the entry of a date in format "YYYY:MM:DD hh:mm:ss" (year:month:day hour:minute:second).

The standard value for attributes of this type is "2002:01:01 00:00:00" or a value defined in the application library.

The value of an attribute of type "Date time" will be aligned right by default in the ADONIS browser (see chap. 5., p. 44).

5.7 Integer (INTEGER)

An attribute of the type "Integer" is defined as an integer from -1,999,999,999 to 1,999,999,999.

The standard value of attributes of this type is "0" or a value defined in the application library.

Hint: The range of values of an "Integer" attribute can be limited during the library definition. An appropriate error message is displayed when a value outside the range is entered.

The value of an attribute of type "Integer" will be aligned right by default in the ADONIS browser (see chap. 5., p. 44).

5.8 Floating Point (DOUBLE)

An attribute of the type "Double" is defined for a float within +/-999,999,999,999,999 for an integer (without decimal places) or +/-999,999,999.999999 for figures with 6 decimals. The corresponding attribute value is displayed to 6 decimal places.

The standard value of attributes of this type is "0.000000" or a value defined in the application library.

Hint: The range of values of a "Double" attribute can be limited during the library definition. An appropriate error message is displayed when a value outside the range is entered.

Hint: The number of figures after the dot can be set in the library definition.

The value of an attribute of type "Double" will be aligned right by default in the ADONIS browser (see chap. 5., p. 44).

5.9 Calendar

An attribute of the type "Calendar" is defined for a performer calendar (=time at which the performer is available at work), for process calendar (=occurrence frequency and probability of business processes) and for agent calendars (=evaluation period of agents).

In the case of the performer calendar, the standard values for attributes of this type are defined from Monday to Friday from 09:00 - 12:00 and from 12:30 to 16:30 (excluding: 1.1., 1.5., 15.8., 26.10., 25.12. and 26.12.). In case of the process calendar it is set up similarly that is from Monday to Friday, between 09:00 - 12:00 and from 12:30 to 16:30 (excluding: 1.1., 1.5., 15.8., 26.10., 25.12. and 26.12.). However the probability with which the process will begin between these times is also specified. The default value for this is the uniform distribution between 15 and 30 minutes. In case of agents, the standard values for attribute of this type are defined from Monday till Saturday, from 00:00 to 24:00 o'clock.

5.10 Long String (LONGSTRING)

An attribute of type "Longstring" is defined for texts up to 32000 characters of any type.

The standard value of attributes of this type is "" (no entry) or a value defined in the application library.

The value of an attribute of type "Longstring" will be aligned left by default in the ADONIS browser (see chap. 5., p. 44).

5.11   Program Call (PROGRAMCALL)

An attribute of the type "Program Call" is characterised by a defined value set of programs to be called or by AdoScripts (see chap. 16., p. 609). A program call has either no value or just one value from this set.

The standard value of attributes of this type is specified in the library definition.

>   **Hint:** The value set of a "Program Call" attribute can be extended (see chap. 2.1.2, p. 184) in the library configuration (Library Management in the Administration Toolkit).

The value of an attribute of type "Program call" will be aligned left by default in the ADONIS browser (see chap. 5., p. 44).

5.12   Inter-model Reference (INTERREF)

An attribute of the type "Interref" is defined for inter-model references on models or objects.

The standard value of attributes of this type is "" (no entry).

The value of an attribute of type "Interref" will be aligned left by default in the ADONIS browser (see chap. 5., p. 44).

5.13   Record (RECORD)

An attribute of the type "Record" (RECORD) is defined by a flexible list/record administration of assembled attribute types.

The standard value for the attributes of this type depends on the attribute type defined in the record class.

>   **Hint:** A record can contain attributes of all ADONIS attribute types except the type "Record" and "Attribute profile reference" .

The value of an attribute of type "Record" will be aligned left by default in the ADONIS browser (see chap. 5., p. 44).

5.14   String (STRING)

An attribute of the type "String" is defined for texts up to 3700 characters of any type.

>   **Hint:** The maximum number of characters for name is 250.

The standard value of attributes of this type is "" (no entry) or a value defined in the application library.

The value of an attribute of type "String" will be aligned left by default in the ADONIS browser (see chap. 5., p. 44).

5.15   Time (TIME)

An attribute of the type "Time" is defined for the entry of a time period in the ADONIS time format YY:DDD:HH:MM:SS (years:days:hours:minutes:seconds).

The standard value of attributes of this type is "00:000:00:00:00" or a value defined in the application library.

The value of an attribute of type "Time" will be aligned right by default in the ADONIS browser (see chap. 5., p. 44).

6. Language Codes

In the following tables, all Windows languages are listed. Every table contains the languages of one codepage.

> **Hint:** In ADONIS, both the ISO 639 codes and the Windows Language IDs can be used.
>
> If more than one Windows Language IDs exists for an ISO 639 code (e.g. "German (Germany)" and "German (Austria)", they are combined into one Windows Language ID. These combinations are displayed in bold letters in the tables (German - for example - with the ISO 639 code "de" is always available in ADONIS with the Windows Language ID "1031").

> **ATTENTION:** The ADONIS installation must be available in the specific languages for being able to use the language codes.

Language	ISO 639 Code	Windows Language ID
Thai	th	1054

Table 5: Languages within the codepage 874

Language	ISO 639 Code	Windows Language ID
Japanese	ja	1041

Table 6: Languages within the codepage 932

Language	ISO 639 Code	Windows Language ID
Chinese (Singapore)	zh	4100
Chinese (People's Republic of China)	**zh**	**2052**

Table 7: Languages within the codepage 936

Language	ISO 639 Code	Windows Language ID
Korean	ko	1042

Table 8: Languages within the codepage 949

Language	ISO 639 Code	Windows Language ID
Chinese (Hongkong S.A.R.)	zh	3076
Chinese (Macao S.A.R.)	zh	5124
Chinese (Taiwan)	zh	1028

Table 9: Languages within the codepage 950

Language	ISO 639 Code	Windows Language ID
Albanian	sq	1052
Bosnian (Bosnia and Herzegovina)	bs	5146
Croatian (Croatia)	**hr**	**1050**

Language	ISO 639 Code	Windows Language ID
Croatian (Bosnia and Herzegovina)	hr	4122
Czech	cs	1029
Hungarian	hu	1038
Polish	pl	1045
Romanian	ro	1048
Serbian (Latin)	**sr**	**2074**
Serbian (Latin, Bosnia and Herzegovina)	sr	6170
Slovak	sk	1051
Slovenian	sl	1060

Table 10: Languages within the codepage 1250

Language	ISO 639 Code	Windows Language ID
Azeri (Cyrillic)	az	2092
Belarussian	be	1059
Bulgarian	bg	1026
Macedonian (FYR)	mk	1071
Kazakh	kk	1087
Kyrgyz	ky	1088
Mongolian	mn	1104
Russian	ru	1049
Serbian (Cyrillic)	sr	3098
Serbian (Cyrillic, Bosnia and Herzegovina)	sr	7194
Tatar	tt	1092
Ukrainian	uk	1058
Uzbek (Cyrillic)	uz	2115

Table 11: Languages within the codepage 1251

Language	ISO 639 Code	Windows Language ID
Afrikaans	af	1078
Basque	eu	1069
Catalan	ca	1027
Danish	da	1030
Dutch (Belgium)	nl	2067
Dutch (Netherlands)	**nl**	**1043**
English (Australia)	en	3081
English (Belize)	en	10249
English (Canada)	en	4105

Language	ISO 639 Code	Windows Language ID
English (Caribbean)	en	9225
English (Ireland)	en	6153
English (Jamaica)	en	8201
English (New Zealand)	en	5129
English (Republic of the Philippines)	en	13321
English (South Africa)	en	7177
English (Trinidad and Tobago)	en	11273
English (United Kingdom)	**en**	**2057**
English (United States)	en	1033
English (Zimbabwe)	en	12297
Faroese	fo	1080
Finnish	fi	1035
French (Belgium)	fr	2060
French (Canada)	fr	3084
French (France)	**fr**	**1036**
French (Luxembourg)	fr	5132
French (Principality of Monaco)	fr	6156
French (Switzerland)	fr	4108
Galician	gl	1110
German (Austria)	de	3079
German (Germany)	**de**	**1031**
German (Liechtenstein)	de	5127
German (Luxembourg)	de	4103
German (Switzerland)	de	2055
Indonesian	id	1057
Italian (Italy)	**it**	**1040**
Italian (Switzerland)	it	2064
Icelandic	is	1039
Malay (Brunei Darussalam)	ms	2110
Malay (Malaysia)	ms	1086
Northern Sotho	ns	1132
Norwegian (Bokmal)	nb	1044
Norwegian (Nynorsk)	nn	2068
Portuguese (Brazil)	pt	1046
Portuguese (Portugal)	pt	2070
Quechua (Bolivia)	qu	1131

Language	ISO 639 Code	Windows Language ID
Quechua (Ecuador)	qu	2155
Quechua (Peru)	qu	3179
Sami (Inari, Finland)	se	9275
Sami (Lule, Norway)	se	4155
Sami (Lule, Sweden)	se	5179
Sami (Northern, Finland)	se	3131
Sami (Northern, Norway)	se	1083
Sami (Northern, Sweden)	se	2107
Sami (Skolt, Finland)	se	8251
Sami (Southern, Norway)	se	6203
Sami (Southern, Sweden)	**se**	**7227**
Swedish	**sv**	**1053**
Swedish (Finland)	sv	2077
Spanish (Argentinia)	es	11274
Spanish (Bolivia)	es	16394
Spanish (Chile)	es	13322
Spanish (Costa Rica)	es	5130
Spanish (Dominican Republic)	es	7178
Spanish (Ecuador)	es	12298
Spanish (El Salvador)	es	17418
Spanish (Guatemala)	es	4106
Spanish (Honduras)	es	18442
Spanish (Colombia)	es	9226
Spanish (Mexico)	es	2058
Spanish (Nicaragua)	es	19466
Spanish (Panama)	es	6154
Spanish (Paraguay)	es	15370
Spanish (Peru)	es	10250
Spanish (Puerto Rico)	es	20490
Spanish (Spain)	**es**	**3082**
Spanish (Spain)	es	1034
Spanish (Uruguay)	es	14346
Spanish (Venezuela)	es	8202
Swahili	sw	1089
Tswana	tn	1074
Welsh	cy	1106

Language	ISO 639 Code	Windows Language ID
Xhosa	xh	1076
Zulu	zu	1077

Table 12: Languages within the codepage 1252

Language	ISO 639 Code	Windows Language ID
Greek	el	1032

Table 13: Languages within the codepage 1253

Language	ISO 639 Code	Windows Language ID
Azeri (Latin)	az	1068
Turkish	tr	1055
Uzbek (Latin)	**uz**	**1091**

Table 14: Languages within the codepage 1254

Language	ISO 639 Code	Windows Language ID
Hebrew	he	1037

Table 15: Languages within the codepage 1255

Language	ISO 639 Code	Windows Language ID
Arabic (Algeria)	ar	5121
Arabic (Bahrain)	ar	15361
Arabic (Egypt)	ar	3073
Arabic (Iraq)	ar	2049
Arabic (Jordan)	ar	11265
Arabic (Kuwait)	ar	13313
Arabic (Lebanon)	ar	12289
Arabic (Libya)	ar	4097
Arabic (Morocco)	ar	6145
Arabic (Oman)	ar	8193
Arabic (Qatar)	ar	16385
Arabic (Saudi Arabia)	**ar**	**1025**
Arabic (Syria)	ar	10241
Arabic (Tunisia)	ar	7169
Arabic (U.A.E.)	ar	14337
Arabic (Yemen)	ar	9217

Language	ISO 639 Code	Windows Language ID
Farsi	fa	1065
Urdu	ur	1056

Table 16: Languages within the codepage 1256

Language	ISO 639 Code	Windows Language ID
Estonian	et	1061
Latvian	lv	1062
Lithuanian	lt	1063

Table 17: Languages within the codepage 1257

Language	ISO 639 Code	Windows Language ID
Vietnamese	vi	1066

Table 18: Languages within the codepage 1258

7. Change Password

You can use ADONIS after you have received a user name and a password from your ADONIS administrator.

When you start ADONIS for the first time, you should change your password. In addition, it is recommended that you change your password frequently for security reasons.

You can change your password in any active component by selecting the option "Change password" from the "Extras" menu. The window "Change password (User: <*user name*>)" (see fig. 356) appears.

Figure 356: Change user password

Enter your old password once and the new one twice and continue by clicking on the OK button or by pressing the enter key. If all entries are correct, the password will be changed.

> **ATTENTION:** The password you enter is case sensitive, i.e. it distinguishes between capital and lower case characters. The password must consist of at least 3 characters. Valid characters for defining a password are the **figures 0 to 9**, the **letters a to z** and **A to Z**, **blanks** and the **characters ! # $ % & ( ) * + , - . / : ; < = > ? @ [ ] ^ _ { | } ~**.

> **Hint:** The passwords are not displayed on the screen. For every character entered an asterisk is displayed.

8.   Messages

The ADONIS email function enables you to send (see chap. 8.1, p. 548) messages to other ADONIS users created on the same ADONIS database, to receive (see chap. 8.4, p. 552) messages from other users and to reply (see chap. 8.2, p. 550) to received messages and/or to forward them (see chap. 8.3, p. 551).

> **Hint:** The addresses of your messages will not be logged in ADONIS when the message is sent. Additionally, you can only send instant messages (see chap. 8.8, p. 555) to ADONIS users who are currently logged in.

The user can change settings (see chap. 8.9, p. 555) concerning messages in order to adapt message functionality to the user needs.

8.1   Send Messages

Send a message to one or more ADONIS users by selecting in the menu "Extras" the menu item "Messages" and the submenu "Send new message".

The window "Messages - Send message" (see fig. 357) will be displayed, in which you can enter the addressee, the subject and the text of the message.
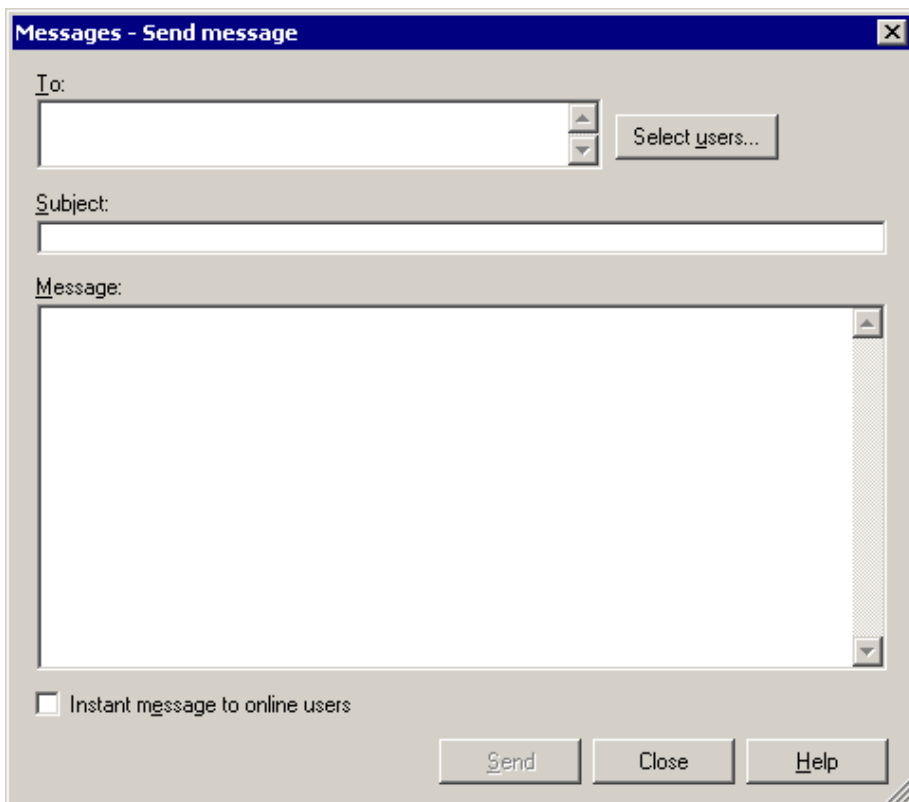
Figure 357: Send message

Enter in the field **"To"** the name(s) of the addressee(s)(= ADONIS user names).

> **Hint:** When entering several ADONIS user names, separate them using a semicolon (";").

Hint: Pay attention to the correct spelling (particularly upper and lower case letters) of the addresses. Clicking on the button "User selection" (see chap. 8.1.1, p. 549) will provide you with support for the entry of the ADONIS user names.

In the field **"Subject"**, you can enter the subject of your message.

In the field **"Message"**, enter the text of your message.

Once you have successfully entered the information, click on the button "Send", to send your message. All addressees logged in will receive information (see fig. 361) about the new message. The addressees not logged in when the message is send will receive the information at their next login.

Select the option **"Instant message to online users"** to send relevant messages to logged in ADONIS users. Instant messages will be directly delivered, i.e. the addressee receives the whole message in a pop-up window (see fig. 364) on her/his screen.

8.1.1 Select Receiver

The support window for the selection of the email receiver(s) shows all ADONIS users created in the ADONIS database (see fig. 358).
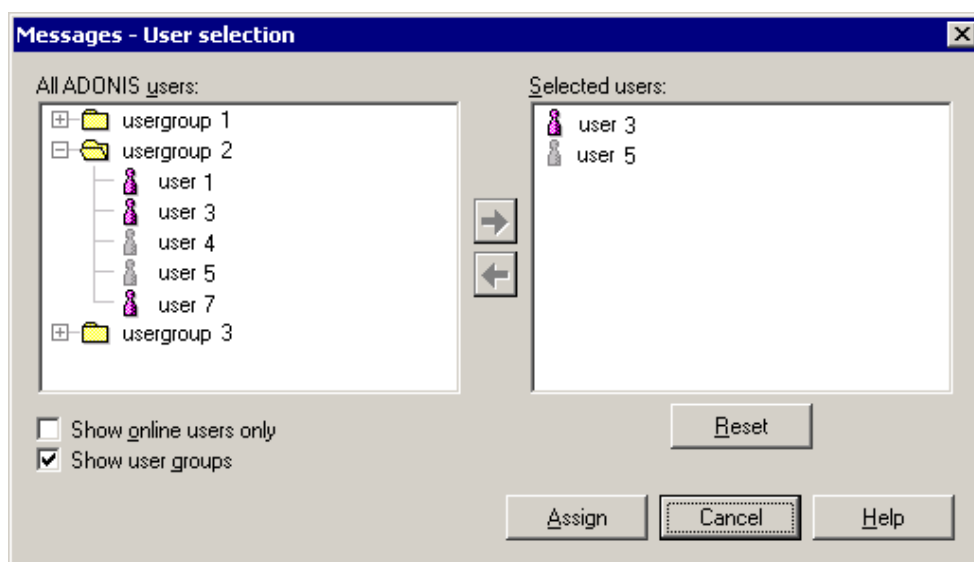
Figure 358: Select user as message receiver

All users are displayed in the **list "All ADONIS users"**. The icon before the user name indicates whether the user is already logged into the ADONIS database 👤 or not 👤 . Depending on the selected options, the display changes as follows:

Show online users only	If this option is activated, only the users currently logged in to the ADONIS database will be listed.
Show user groups	If this option is activated, the users will be shown in the structure of the user group they are assigned to. If this option is deactivated, you will see an alphabetically sorted list of all ADONIS users.

Select the receiver of the message in the list "All ADONIS users" and click on the arrow button ➡, whereby the selected user will be transferred to the list **"Selected users"**.

When a user group is selected and the arrow button ➡ is clicked, all users, which are assigned to the user group, will be transferred to the list "Selected user".

You can remove single users from the list "Selected users", by selecting them and clicking on the arrow button ←. Clicking on the button **"Reset"** deletes all selected users.

Once you have successfully selected the message receivers, click on the button "Assign". As a result, the selected user will be transferred to the field "To" in the window "Messages - Send message" (see fig. 357).

8.2   Reply Message

Reply to messages of an ADONIS user, by selecting them in the window "Messages - received messages" (see fig. 362) and clicking on the "reply" button.

Hint: You can reply on an instant message directly from the message window (see fig. 364).

The window "Messages - Reply to messages" (see fig. 359), in which you enter the reply, will be displayed. The receiver and the subject field will be set automatically.
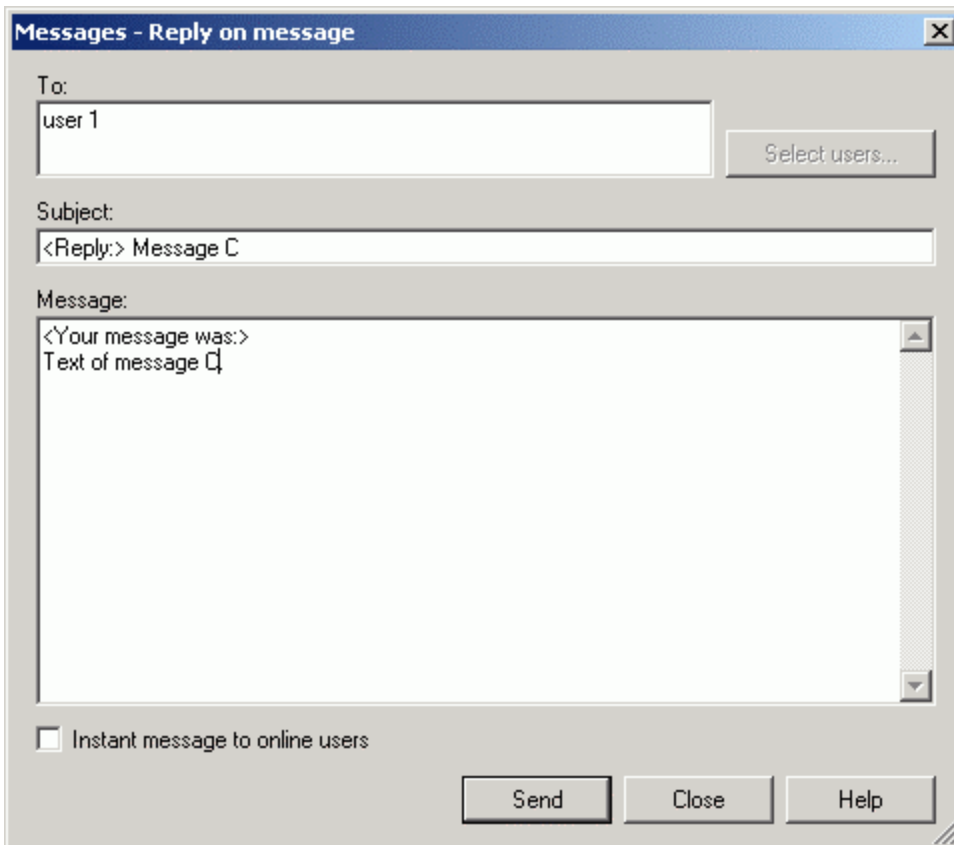
Figure 359: Reply to messages

The name of the receiver (=sender of the previously selected message) has been transferred to the field **"To"**.

The title of the previously selected message has been transferred to the **field "Subject"**, the title is preceded by "**<Reply:>**".

The text of the previously selected message has been transferred to the field **"Message"**, whereby the message is preceded by "**<Your message was:>**". Enter the text of your reply.

Once you have successfully entered the message, click on the button "Send", to send your message.

Select the option **"Instant message to online users"** to send the reply as an instant message to the ADONIS users currently online.

8.3 Forward Message

Select in the window "Messages - received messages" (see fig. 362), the messages you want to forward to one or more ADONIS users and click on the button "Forward".

The window "Messages - forward messages" (see fig. 360), in which you can, if necessary, enter the receiver and a comment, which will be displayed. The subject and the text will be set automatically.
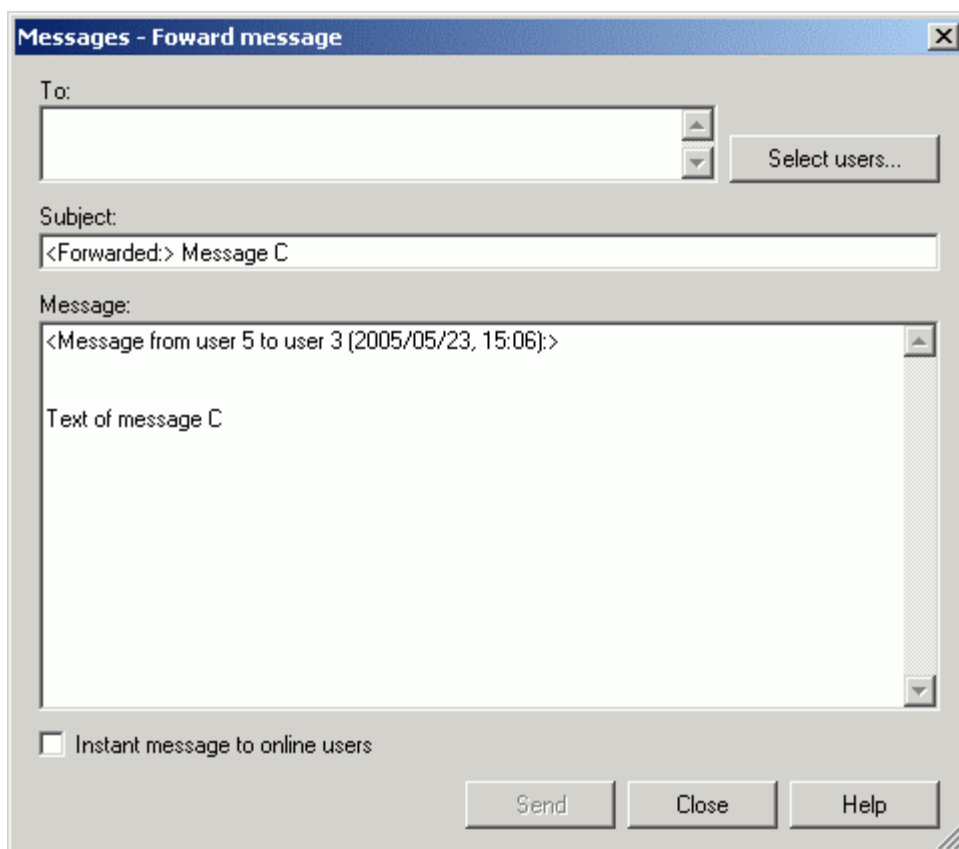
Figure 360: Forward message

Enter in the field **"To"** the name(s) of the addressee(s) (= ADONIS user names).

> **Hint:** When entering several ADONIS user names, you have to separate them with semicolons (";") .

> **Hint:** Pay attention to the spelling (incl. capitalisation) of the addresses. Clicking on the button "User selection" (see chap. 8.1.1, p. 549) you will get a support window for the entry of the ADONIS user names.

The title of the previously selected message has been transferred to the field **"Subject"**, the title is preceded by "**<Forwarded:>**".

The title of the previously selected message has been transferred to the field **"Subject"**, whereby the text is preceded by "**<message from <user name> to <user name> (<date>, <time>):>**". you can also enter text in the message to be forwarded.

After the successful entry click on the button "Send" to forward the message.

Select the option **"Instant message to online users"** to forward the message as an instant message to the ADONIS users.

8.4   Receive New Message

New messages arriving will be announced by the information window (see fig. 361).

Figure 361: Indication of new message

Click on the OK button to close the window and to have the received message displayed (see chap. 8.5, p. 552) at a later time.

Clicking on the button "Open" will show the window "Messages - Received messages" (see fig. 362) with the new messages (see chap. 8.6, p. 552) in ADONIS.

> **Hint:** When the option "Info box on new messages" is deactivated in the settings (see chap. 8.9, p. 555), the arrival of new messages will be announced by a prompt sound (instead of the info box).

> **Hint:** By deactivating the option "Messaging system active" in the settings (see chap. 8.9, p. 555), the arrival of new messages will **not** be announced.

8.5   Display Message

To display new messages, select the menu item "Messages" and the submenu "Read messages" from the "Extras" menu.

The window "Messages - received messages" (see fig. 362) containing all received messages (see chap. 8.6, p. 552) will be opened in ADONIS.

8.6   Received Message

The window "Messages - received messages" (see fig. 362) lists the received messages in a table.
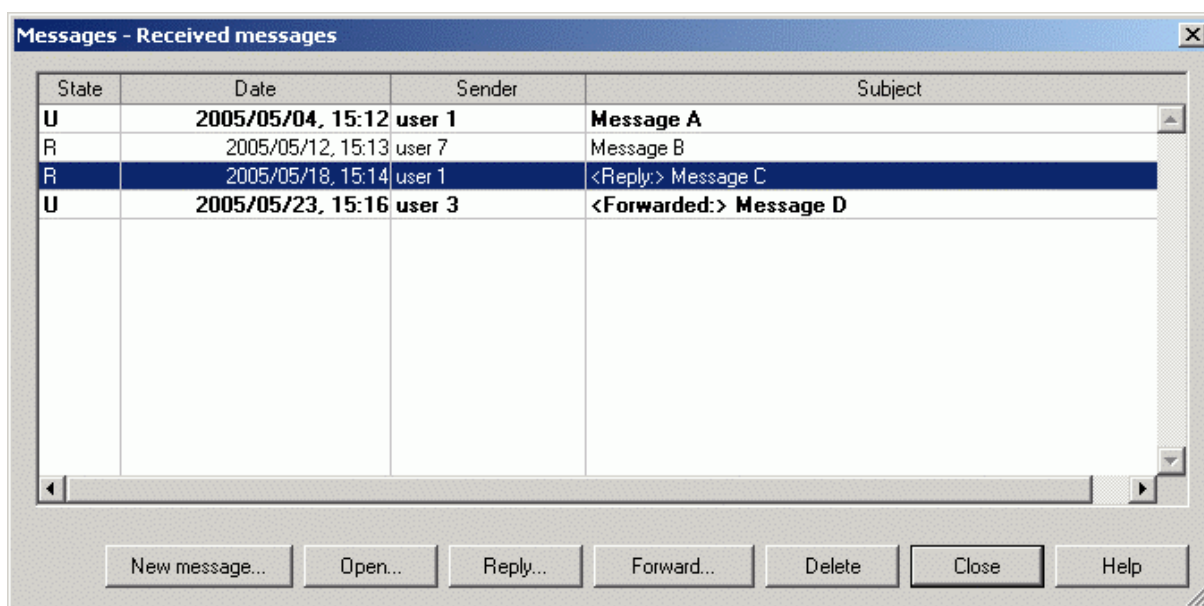
Figure 362: List of received messages

The list of received messages contains the following information:

● State

● Date

● Sender

● Subject

The column **"State"** can display the following entries:

N New messages, i.e. messages received just now or during the display of the window "Messages - received messages" (see fig. 362).

   Note: New messages will be shown as unread messages (U) when closing the window "Messages - Received messages".

U Unread messages, i.e. messages that have been received but not yet read.

R Read messages.

   Hint: New and unread messages will be brought out in bold.

The column **"Date"** contains the day and the time when the message was sent.

The column **"Sender"** contains the name of the ADONIS user who sent the message.

The column **"Subject"** will show the title of the message.

   Hint: The title of the message will be preceded by "**<Reply:>**" if it is a reply to a previously sent message.

   Hint: The title of the message will be preceded by "**<Forwarded:>**" if it is a forwarded message.

The following buttons are available under the list of received messages:

"New message" to create and send a new message (see chap. 8.1, p. 548).

"Open"	to open and read a selected message (see chap. 8.7, p. 554).
"Reply"	to reply to a selected message (see chap. 8.2, p. 550).
"Forward"	to forward a selected message (see chap. 8.3, p. 551).
"Delete"	to delete previously selected messages.

> **Hint:** You can also use the functions "Open", "Answer", "Forward" and "Delete" after having selected a message if you open the context menu (right mouse button).

> **Hint:** You can select several messages for the function "Delete".

8.7 Read Message

The text of a message will be displayed in a window (see fig. 363).
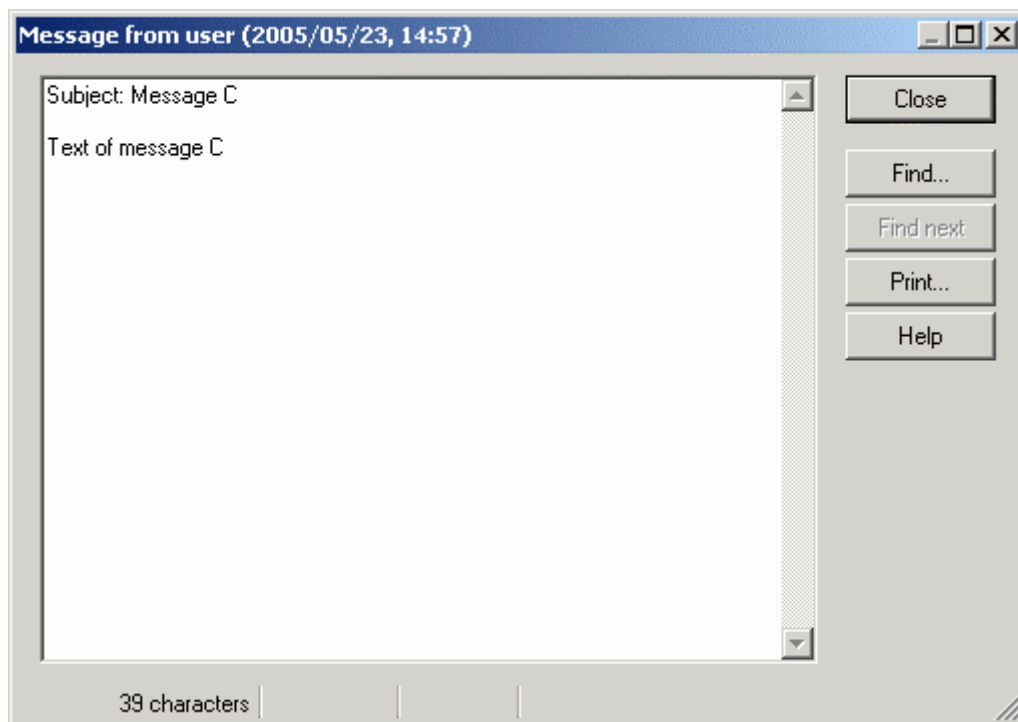
Figure 363: Display of received message

You can take the sender as well as the date and time of sending from the **title line** of the window.

The subject will be specified in the first line of the **text field** and the text of the message is displayed underneath.

> **Hint:** If replying to a message, the information "**<Reply:>**" in the subject line as well as "**<Your message was:>**" will be displayed before the original message.

> **Hint:** If forwarding a message, the information "**<Forwarded:>**" will be displayed in the subject line as well as "**<Message from *<user name>* to *<user name>* (*<date>*, *<time>*):>**" in the text field before the forwarded message.

By clicking on the **button "Press"** you can print the message, the **buttons "Find"** and **"Find more"** enable the finding of text within the message.

8.8 Receive Instant Message

When an instant message has been received, the whole message, i.e. sender, sending time, title and text will be displayed in a window (see fig. 364).
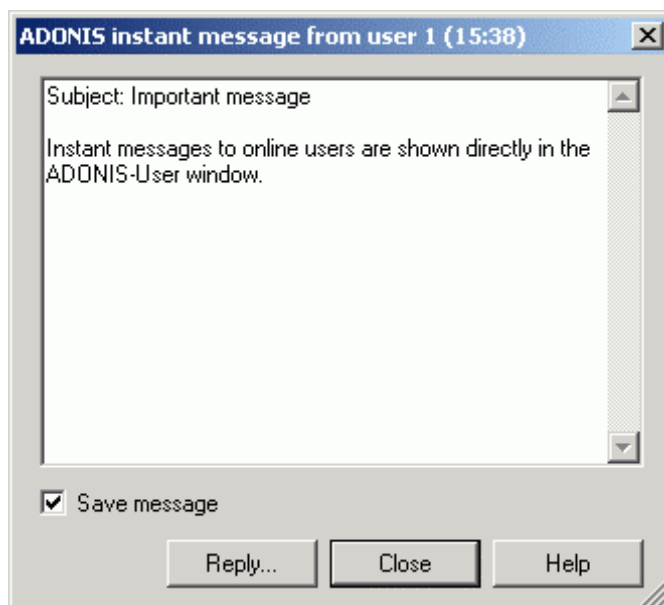
Figure 364: Instant message display

By clicking on the **button "reply"**, you can reply to the message (see chap. 8.2, p. 550).

Click on the button: **"Close"** to close the window. By activating the option **"Save message"**, the instant message will be sent and can be read again later.

> **Hint:** By deactivating the option "Show instant message immediately" in the settings (see chap. 8.9, p. 555), the appropriate window (see fig. 361) will be displayed instead of the instant message.

8.9 Settings

In the window "Messages - Settings" (see fig. 365), you can carry out the settings described as follows.

Figure 365: Message settings

Option "Messaging system active"

If this option is activated, new messages will be announced by an appropriate info box (see fig. 361).

If this option is deactivated, the arrival of new messages will not be announced.

Option "Show instant message immediately"

If this option is activated, instant messages received, i.e. the whole message (sender, sending time, title and text), will be immediately shown in a window (see fig. 364).

If this option is deactivated, the arrival of instant message will be announced by an info box (see fig. 361).

Option "Info box on new messages"

If this option is activated, new messages will be indicated by an appropriate info box (see fig. 361).

If this option is deactivated, the arrival of a message will be announced by a sound prompt.

"Time interval for quering for new message (minutes)"

Specify the frequency (in minutes) to check for new message arrivals in the ADONIS database.

Hint: We recommend to deactivate the message system during long-lasting processing in ADONIS (e.g. simulation runs, import/export of models), since the announcement of new messages can abort the process, which can only be continued once the message window has been closed.

9.   File Formats in ADONIS

The following file formats can be generated and are supported within ADONIS:

Hint: The extensions listed below are suggested by the program. They are optional i.e. other extensions may be used instead. Note however that when searching for files, the filters are preset in order to search for the appropriate extension as listed below. Also using the preset extension makes it easier to see which type of file you are dealing with.

9.1   ADONIS Binary Language (ABL)

ABL files are generated when exporting application libraries (see chap. 2.3.2, p. 284) from within the Administration Toolkit.

An application library which has been exported into an ABL file may be imported into the ADONIS Administration Toolkit (see chap. 3., p. 16) on another computer.

9.2   ADONIS Comparable Representation (ACR)

ACR files are generated after saving simulation and query results, when the results are to be later used in the "Comparison of results" option within the "Evaluation" component.

9.3 ADONIS Definition Language (ADL)

ADL files are generated when exporting models (and/or model groups). ADL files can then be imported and edited within the ADONIS Business Process Management Toolkit on other computers which may not be connected to a network or which are connected to different databases.

9.4 ADONIS Protocol Format (APF)

APF files can be created as protocols in short form while running the "Capacity analysis" or "Workload analysis" simulation algorithms.

These files can then be read by the "Offline animation" function, which allows you to play through the simulation again observing the times and optionally adding breakpoints at certain times.

9.5 Windows-Bitmap (BMP)

BMP files can be created when generating graphics of models, model sections or the graphical representation of the ADONIS browser results. In addition BMP files can be created when generating a documentation.

BMP files may be opened or inserted into graphics programs and may be edited further there or integrated into other applications such as word processor applications.

9.6 Comma Separated Value (CSV)

CSV files will be created when saving the contents of the ADONIS browser.

CSV files can be further edited in spreadsheet calculation programs (without loss).

9.7 Windows Enhanced Metafiles (EMF)

EMF files will be created when generating graphics from models or model parts and when copying the graphic result display of the ADONIS browser. In addition EMF files can be created when generating a documentation.

EMF files can be inserted into graphic programs and be further edited there or connected to any applications (e.g. spreadsheet calculation programs).

9.8 FlowMark Definition Language (FDL)

FDL files are generated when ADONIS models are transformed in the Import/Export component. FDL files can be imported into the Workflow Management system WebSphere MQ/MQSeries Workflow/FlowMark from IBM.

9.9   HyperText Markup Language (HTML)

HTML files can be created when saving the contents of the ADONIS browser and also when generating the HTML documentation from model contents within the "Documentation Component".

HTML files can be viewed in a Web Browser.

9.10   JPEG File Interchange-Format (JPG)

JPEG or JPG files can be created when generating graphics of models, model sections or the graphical representation of the ADONIS browser results. In addition JPG files can be created when generating a documentation.

JPG files may be opened or inserted into graphics programs and may be edited further there or integrated into other applications such as word processor applications.

9.11   ZSoft Paintbrush Graphic (PCX)

PCX files will be created when generating graphics of models, model sections or the graphical representation of the ADONIS browser results.

PCX files may be opened or inserted into graphics programs and may be edited further there or integrated into other applications such as word processor applications.

9.12   Portable Network Graphics (PNG)

PNG files will be created when generating graphics from models or model parts and when copying the graphic result display of the ADONIS browser. In addition PNG files can be created when generating a documentation.

PNG files can be inserted to graphic programs and then further edited or be connected to any application (e.g. spreadsheet programs).

9.13   Rich Text Format (RTF)

RTF files can be created when saving the contents of the ADONIS browser and also when using the HTML generation within the "Documentation Component"

RTF files can be further edited in spreadsheet calculation programs.

9.14   Scalable Vector Graphics (SVG)

SVG files are used when generating graphics from models or model parts and when copying the graphical result representation of the ADONIS browser. In addition SVG files can be created during documentation generation.

In general, SVG files are used in HTML browsers.

> **Hint:** SVG files can be displayed using Microsoft Internet Explorer (Version 5.0 and up, with SVG Viewer plugin) or Mozilla Firefox (Version 1.5 and up).

However, it is recommended to use Microsoft Internet Explorer with SVG Viewer Plugin, as Mozilla Firefox does not interpret text and some other SVG objects correctly.

9.15 ASCII-Text (TXT)

TXT files are generated when saving simulation and analysis results as well as the contents of an ADONIS Notebook. TXT files can be inserted in other applications and edited further.

9.16 User Definition Language (UDL)

UDL files are created by exporting (see chap. 1.7, p. 122) users (and/or user groups) in the User Management component in the Administration Toolkit. UDL files can then be imported (see chap. 1.6, p. 116) into the Administration Toolkit on a different computer. (This can be useful for example where the database system is changing).

9.17 Extensible Markup Language (XML)

XML files will be created when exporting models. The models which have been exported to XML files, can for instance be imported to other computers, where they can be edited.

Hint: In ADONIS 3.9 XML files will be created with encoding "UTF-8".

10. Expressions

Expressions are - rougly speaking - formulae for investigating or calculating something. This can be both a fixed calculation or the processing of attribute values. The syntax of the language for EXPRESSION attributes is based on the following grammar:

__ExprDefinition__: **EXPR type:** *ResultType* [**format:***FormatString*]
                    expr:[**fixed:**] *CoreExpression* .

__ResultType__: **double** | **integer** | **string** | **time** .

Each definition of the EXPRESSION attribute starts with the keyword **EXPR**. The result type will be fixed by means of the attribute **type:** and a standard fomula expr. Each time an instance is created (object, model or connector), the formula will be used to calculate the result value of the expression.

By setting the modifier as **fixed:** the expression will be unchangeable in the Business Process Management Toolkit. It means that the EXPRESSION attribute evaluates the same formula everywhere.

> **Hint:** The formula cannot be longer than 3.600 characters.

In the expressions where you can change the instances, each evaluation instance will be assigned to an appropriate expression. Then the specification of an expression in the "Expression-Definition" is optional and is of use for initial assignment of an expression to evaluation instances. For each evaluation instance, you can open an input window from the ADONIS Notebook, which enables the editing of the expression. The changeable expression will be saved to the evaluation instance under **expr**.

In ADONIS, three different types of expressions can be found:

- **LeoExpressions** (see chap. 17.3, p. 674)**:** LeoExpressions are a set of basic functions and operators, from which an expression can be aggregated. They are used to calculate values, manipulate strings and perform other similar basic tasks. LeoExpressions can be used with every LEO language. Wherever LEO strings are possible, LEO expressions can be used.

- **CoreExpressions:** A CoreExpression is an ADONIS-specific extension of a LeoExpression. When creating a CoreExpression, both functions and operators of LeoExpressions and core-specific functions designed specifically for ADONIS can be utilised. CoreExpressions can only be used in attributes of type **EXPRESSION** (see chap. 5.4, p. 537).

- **AdoScriptExpressions:** An AdoScriptExpression is an extension of a LeoExpression. Consequently, all functions and operators of the LeoExpressions can be used here as well. Furthermore, specifically designed AdoScript functions can be utilised. AdoScriptExpressions can only be used within **AdoScripts** (see chap. 16., p. 609) in the ADONIS Administration Toolkit.

The CoreExpressions functions:

> **Hint:** The word "this" always refers to the context where the expression is evaluated.

aval (*instID, attrName***)**

provides the value of the instance object specified by both parameters.

aval (*attrName***)**

provides the attribute value *attrName* of the instance.

aval (*instName, attrName***)**

provides in this model of this class the instance with an attribute value *attrName.*

aval (*className, instName, attrName***)**

provides the attribute value of the instance in this model which is specified by the three parameters.

avalf (*format,*** *instID,*** *attrName***)**

avalf (*format,*** *attrName***)**

avalf (*format,*** *instName,*** *attrName***)**

avalf (*format,*** *className,*** *instName,*** *attrName***)**

These functions correspond to the call of aval without the first parameter, whereby INTERREF attribute will be formatted. The string *format* will be called for each reference, whereby the following replacements will be carried out: **%o** by the object name of the target object, **%c** by the class name of the target object, **%m** by the model name of the target model, **%t** by the model type name of the target model. The results will be joined together separated by a line break, which is the result of the call.

maval (*attrName***)**

provides the value of a model attribute *attrName* of this model.

irtmodels (*attrName***)**

provides the models of this instance referenced by the INTERREF attribute *attrName*.

irtmodels (*instID,*** *attrName***)**

provides the models referenced in this model by the INTERREF attribute of this instance.

irtmodels (*instName,*** *attrName***)**

provides the models referenced in this model by the INTERREF attribute of the instance of this class.

irtobjs (*attrName***)**

provides the objects referenced by an INTERREF attribute of this instance.

irtobjs (*instID,*** *attrName***)**

provides the objects referenced in this model by the INTERREF attribute *attrName* of the instance *instID*.

irtobjs (*instName,*** *attrName***)**

provides the objects referenced in this model by the INTERREF attribute *attrName* of the instance *instName* of this class.

objirsobjs ()

provides the objects referencing this instance. The usage of this function is not recommended due to great preformance losses.

objirsobjs (*targetName***)**

provides the objects referencing the instance *targetName*. The usage of this function is not recommended due to great preformance losses.

objirsobjs (*targetID***)**

provides the objects referencing the instance with the ID *targetID*. The usage of this function is not recommended due to great preformance losses.

modelirsobjs ()

provides the objects referencing this model. The usage of this function is not recommended due to great preformance losses.

profile (*attrName***)**

provides the attribute profile referenced by the INTERREF attribute *attrName* of this instance.

paval (*profilerefAttrName, attrNameInProfile***)**

determines the attribute profile referenced by the ATTRPROFREF attribute *attrNameInProfile* of this instance and provides the value of an attribute it contains *attrNameInProfile*.

pavalf (*format, profilerefAttrName, attrNameInProfile***)**

determines the attribute profile referenced by the ATTRPROFREF attribute *profilerefAttrName* of this instance and provides the value of an INTERREF attribute *attrNameInProfile* it contains, whereby the format is the same as in **paval**.

ctobj()

provides the target object of this connector.

cfobj()

provides the start object of this connector.

ctobjs (*relnName***)**

provides the objects connected with this object in an outgoing direction via the specified relation *relnName*.

ctobjs (*instid, relnName***)**

provides the objects connected with the specified object in an outgoing direction via the specified relation *relnName*.

cfobjs (*relnName***)**

provides the objects connected with this object in an incoming direction via the specified relation *relnName*.

cfobjs (*instid, relnName***)**

provides the objects connected with the specified object *instid* in incoming direction via the specified relation *relnName*.

conn (*relnName, fromObjs, toObjs***)**

provides the connectors of the class *relnName* between the specifed objects. The objects per parameter can be seen as one ID, or many IDs containing strings.

objofrow (*rowID***)**

provides the instance, which contains the specified RECORD row *rowID*.

rcount (*recordAttrName***)**

provides the number of rows of a RECORD attribute *recordAttrName* of this instance.

row (*recordAttrName, rowIndex***)**

provides a row *rowIndex* of a RECORD attribute *recordAttrName* of this instance. The numbering of rows starts with **1** and finishes with **rcount()**.

rasum (*recordAttrName, columnName***)**

provides the sum of a record column *columnName* of the attribute *recordAttrName* of this instance.

rasum (*idStr, recordAttrName, columnName***)**

determines the sum of the specified RECORD columns *columnName* for each specified instance and provides the sum of all the results.

prasum (*profileAttrName, recordAttrName, columnName***)**

determines the sum of a RECORD column *columnName* in the attribute profile referenced by an ATTRPROFREF attribute of this instance.

> **ATTENTION:** Please note that all the summed values with **rasum()** or **prasum()** must be of the same type. The accepted types are integers and floating point numbers; invalid values are ignored. EXPRESSION attributes, which provide values in both formats, can also be summed up.

isloaded (`modelid`**)**

returns '1' if the model *modelID* is loaded, otherwise '0'.

allobjs (`modelID, className`**)**

provides all objects of the class *className* in the specified model *modelID*.

aql (`aqlExpr`**)**

evaluates the AQL expression and provides the result objects in form of an ID string.

prevsl()

provides the predecessors of this swimlane.

nextsl()

provides the successors of this swimlane.

asum (`idStr, attrName`**)**

provides the sum of all attribute values of the specified attribute *attrName* for the specified instances *idStr*.

amax (`idStr, attrName`**)**

provides the maximum number of all attribute values of the specified attribute *attrName* for the specified instances *idStr*.

awsum (`instIDStr, objAttrName, connIDStr, connAttrName`**)**

provides the weighted sum for all the specified instances *instIDStr* of all attribute values of the given attribute *objAttrName*. This sum is weighted by attribute values *connAttrName* of the connectors *connIDStr*.

pmf (`className, profileAttrName, recAttrName, orgunitColName, quantityColName, factorAttrName`**)**

Human resource management function.

pmf (`className, recAttrName, orgunitColName, quantityColName, factorAttrName`**)**

Human resource management function (old version).

class (`instID`**)**

provides the classes ID of an instance *instID*.

class (`className`**)**

provides the ID of the class having the name *className*.

mtype ()

provides the model type of this model.

mtype (`modelID`**)**

provides the model type of the specified model *modelID*.

mtclasses (`modelTypeName`**)**

provides the classes allowed in the model type **modelTypeName**.

mtrelns (`modelTypeName`**)**

provides the relation classes, which are allowed in the model type **modelTypeName**.

allcattrs (*classID***)**

provides all class attributes of the specified class *classID*.

alliattrs (*classID***)**

provides all instance attributes of the class *classID*.

allrattrs (*relationID***)**

provides all attributes of the specified relation *relationID*.

attrname (*attrID***)**

provides the name of the specified attribute *attrID*.

atttype (*attrID***)**

provides the type name ("INTEGER", "DOUBLE", "STRING", "DISTRIBUTION", "TIME", "DATETIME", "ENUMERATION", "ENUMERATIONLIST", "LONGSTRING". "PROGRAMCALL", "INTERREF", "EXPRESSION", "RECORD" or "ATTRPROFREF") of the specified attribute *attrID*.

Additionally, the following variables are preset in every EXPRESSSION attribute:

- **attrid** with the ID of this attribute
- **objid** with the ID of this object/connector
- **classid** with the ID of this class/relation class
- **modelid** with the ID of this model

For every evaluation instance the result of its expression is calculated and saved to the attribute values of the EXPRESSION attribute under **val**. A new calculation is triggered every time the result could possibly have changed. This is the case when the expression was changed or when an attribute value necessary for the expression has been changed in the core component.

11. HOMER Scenarios

The HOMER scenario describes the settings of an acquisition table for a specific use (e.g. acquisition of execution times, acquisition of the documents used).

In HOMER the user can create different scenarios for the purpose of acquisition.

You can manage the scenarios (see chap. 11.1, p. 566) using the scenario manager.

The definition of the table structures (see chap. 11.2, p. 571), i.e. the definition of the settings of the acquisition tables is carried out over the scenario settings.

11.1 HOMER Scenarios Administration

Select in the menu "Options" the menu item "Scenario Manager", to open the window "HOMER: Scenario Manager" (see fig. 366).
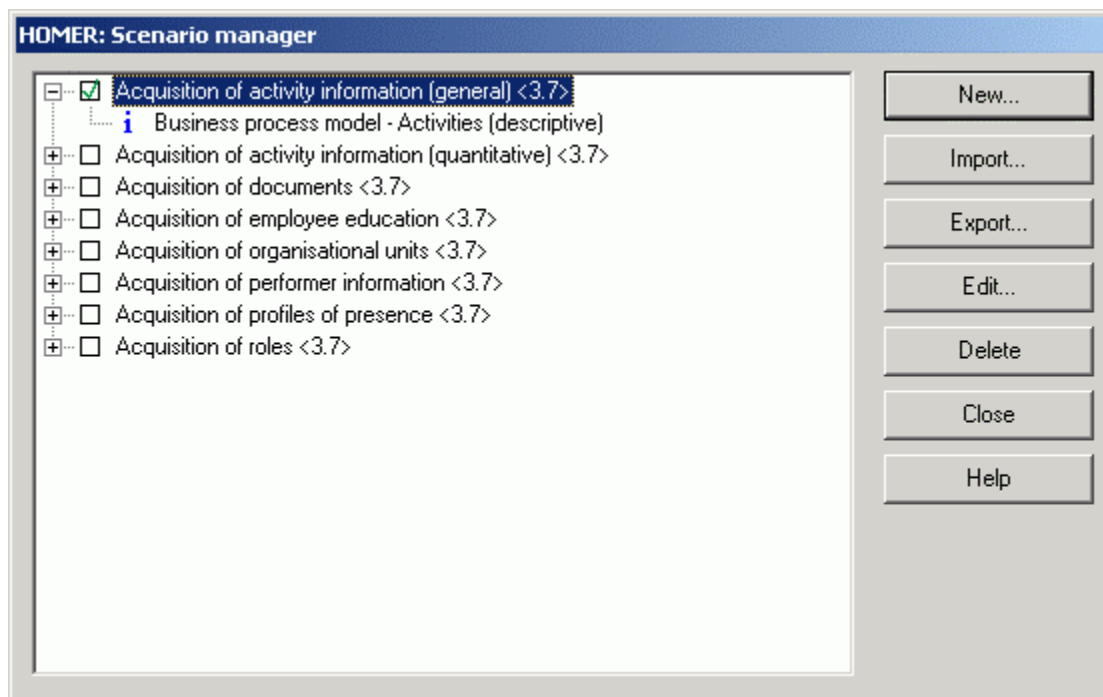
Figure 366: HOMER: Scenario Manager

Click on the respective button to do the following:

Create	to create a new scenario (see chap. 11.1.1, p. 567),
Import	to import scenarios (as well as INI files of earlier HOMER versions) (see chap. 11.1.2, p. 567),
Export	to export scenarios (see chap. 11.1.3, p. 570),
Edit	to edit the name, the verison and the keywords of the selected scenario (see chap. 11.1.4, p. 571),
Delete	to delete the selected scenario (see chap. 11.1.5, p. 571),

Exit   to exit the scenario manager.

The **selection of a scenario** is done when you activate the checkbox of the appropriate scenarios by clicking on it. An activated scenario is represented by the symbol ☑ and appears in the status bar of HOMER.

11.1.1   Creating HOMER Scenario

To create a scenario, click in the "HOMER: Scenario manager" window (see fig. 366) on the button "New". Enter the name, a version number as well as a keyword for the new scenario in the window "HOMER: Create new scenario" (see fig. 367).

Figure 367: Create HOMER scenario

Hint: The fields "Name" and "Version" have to be filled.

Then click on the button "Continue", the scenario will be created and then you can define the structure of the acquisition table (see chap. 11.2, p. 571).

11.1.2   Importing HOMER Scenario

By importing the HOMER scenarios, you can

- import (see chap. 11.1.2.1, p. 568) scenarios created and exported with HOMER 3.6 or higher (XLS files) or,

- convert (see chap. 11.1.2.2, p. 569) scenarios (INI files) created in earlier HOMER versions to the format of HOMER 3.7.

  ATTENTION: The conversion of existing INI files cannot be aborted. Depending on the size of the scenario this might take a few minutes.

  Hint: You should transfer all existing acquisition tables to ADL first with the old HOMER version, since columns will be moved during the conversion. Once you have successfully converted the INI files, you can **automatically** transfer the acquisition tables converted in ADL into HOMER.

  ATTENTION: Scenarios from XLS files as well as scenarios from converted INI files can only be used with HOMER version 3.6 and higher.

To import scenarios or existing INI files, click in the window "HOMER: Scenario manager" (see fig. 366) on the button "Import".

The window "Select HOMER scenario" will appear (see fig. 368).

  ATTENTION: INI files converted for HOMER 3.7 can only be used with HOMER 3.7.
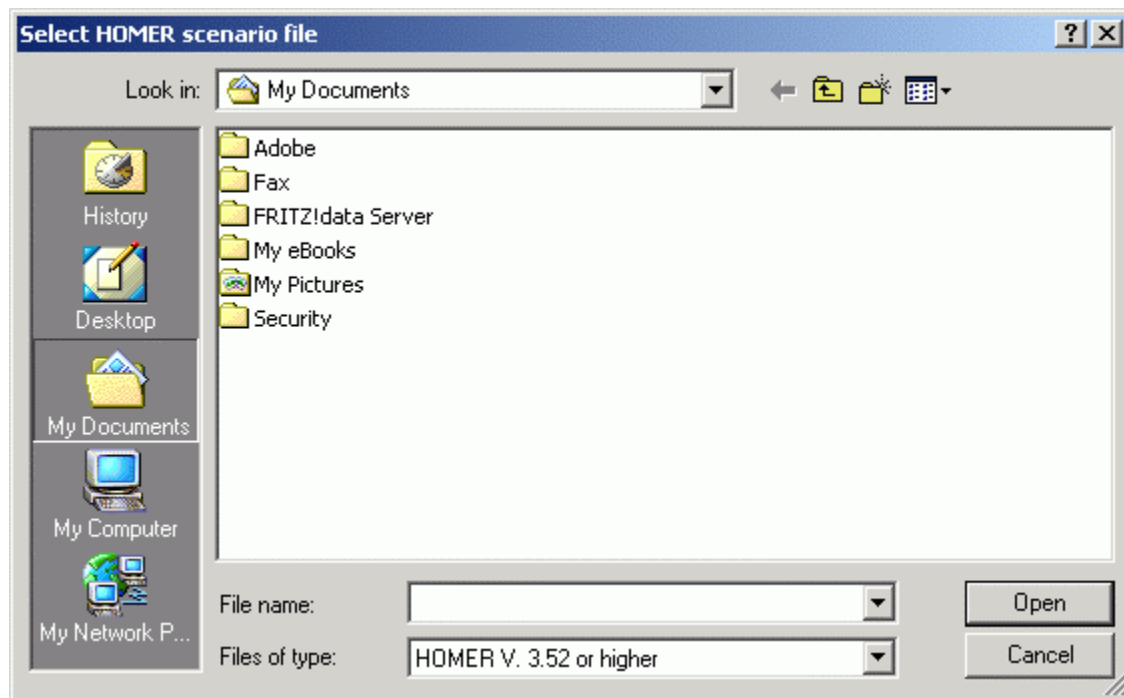
Figure 368: Import HOMER scenario - Select INI file

Select from the **list "Type of file"** the entry "HOMER V. 3.52 or higher", to import (see chap. 11.1.2.1, p. 568) a HOMER scenario, or "HOMER V. 2.0 to V. 3.5", to convert (see chap. 11.1.2.2, p. 569) an INI file.

11.1.2.1 Importing HOMER Scenario

ATTENTION: Scenarios from XLS files as well as scenarios from converted INI files can only be used with HOMER version 3.6 and higher.

Select the XLS file to import and then click on the button "Open". The window "HOMER: Select scenario for the scenario import" (see fig. 369).
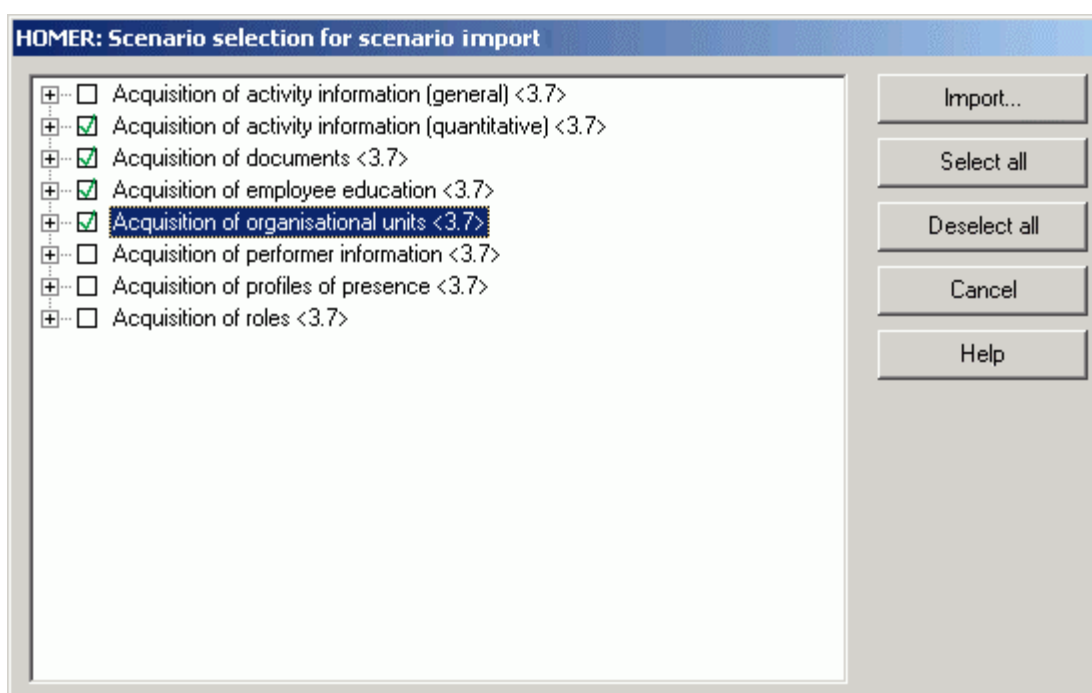
Figure 369: HOMER scenario import - Scenario selection

Select the scenario to be imported (multiple selection is possible) and then click on the button "Import", to start the import.

> **Hint:** When you import an existing scenario, the error message [hacini-34] will be displayed. Close this message and then change the name and/or the version number of the scenario you want to import in the window "HOMER: edit scenario" (see chap. 11.1.4, p. 571).

The successful import will be indicated by a specific message.
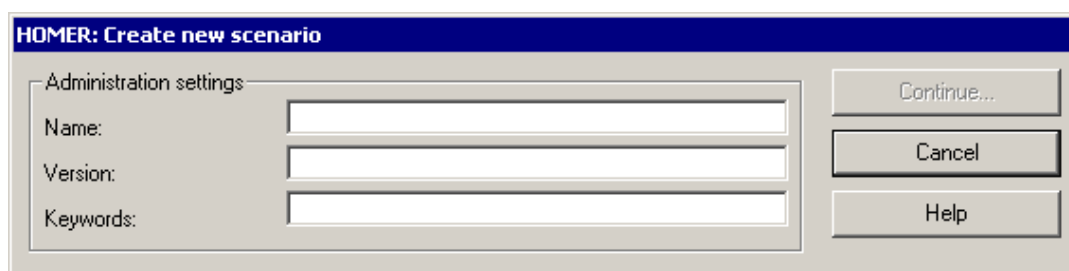
> **Hint:** If the file HOMER.xls (in the HOMER program directory) is write-protected no new scenarios are created in the window "HOMER: scenario manager" even when receiving the message that the import has been successful. Remove the write-protection to successfully import the scenarios.

11.1.2.2   Converting HOMER INI File

> **ATTENTION:**   The conversion of existing INI files cannot be aborted. Depending on the size of the scenario this might take a few minutes.

> **Hint:** You should transfer all existing acquisition tables to ADL first with the old HOMER, since columns will be moved during the conversion. Once you have successfully converted the INI files, you can **automatically** transfer the acquisition tables converted in ADL again into HOMER.

Select the INI file you want to import and then click on the button "Open". The window "HOMER: Create new scenario" (see fig. 370) will appear.

Figure 370: Import HOMER scenario - create scenario

Enter the name and the version number of the scenario you want to create and, if necessary, some keywords and then click on the button "Next".

> **Hint:** The fields "Name" and "Version" must be filled.

After the conversion, the window "HOMER: scenario settings" with the defined table structure (see chap. 11.2, p. 571) will appear.
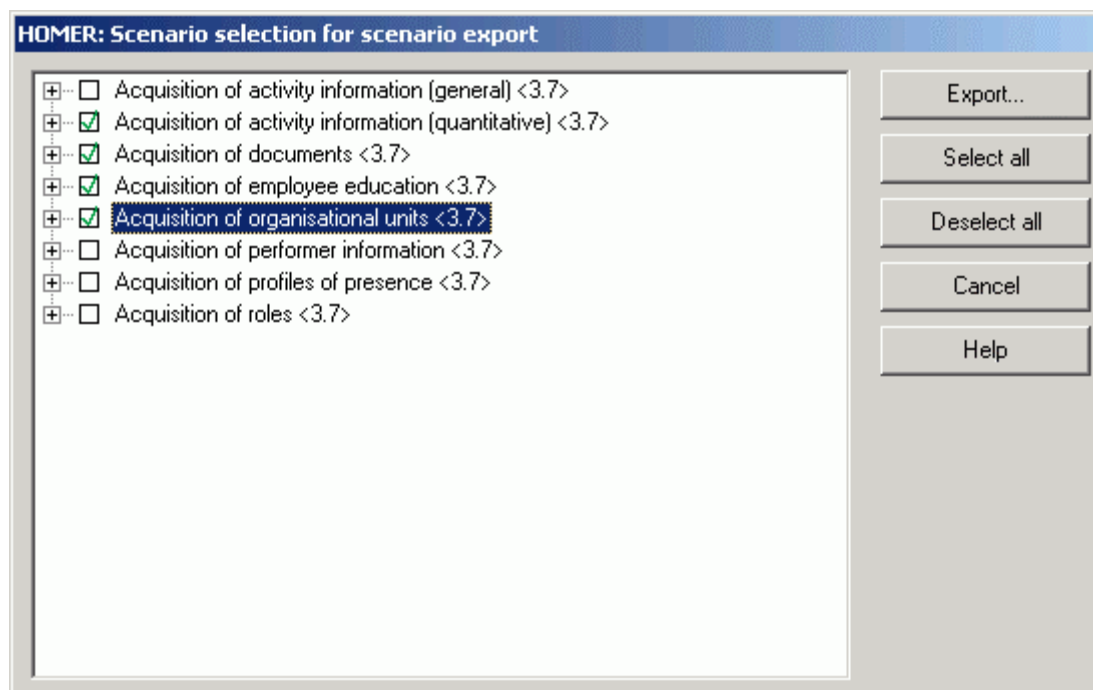
Check the settings (especially the version number of your current ADONIS version as well as the name of the basis library). To finish the conversion, click on the button "Close".

11.1.3   Exporting HOMER Scenario

When exporting scenarios, you can export an existing scenario to an XLS file, for instance to save this scenario or to transmit it to other HOMER users. The XLS file is protected by a password and can only be edited using HOMER.

To import scenarios or to import existing INI files, click in the window "HOMER: Scenario manager" (see fig. 366) on the button "Export".

The window "HOMER: Select scenario for the scenario export" (see fig. 371), with all previously defined scenarios will be opened.

Figure 371: HOMER-export scenarios - scenario selection

Select the scenario you want to export and then click on the button "Export".

In the window "Export scenario", enter the file name as well as the path for the XLS file, to which the selected scenarios should be exported, and then click on the button "Save".

The successful export will be indicated by an appropriate message.

11.1.4   Editing HOMER Scenario

To edit a scenario, select in the window "HOMER: Scenario manager" (see fig. 366) the appropriate scenario and then click on the button "Edit". In the window "HOMER: edit scenario" (see fig. 372), you can change the name, the version number as well as the keyword.

Figure 372: Edit HOMER scenario

Hint: The fields "Name" and "Version" have to be filled.

Then click on the OK button, to adopt the changes.

11.1.5   Deleting HOMER Scenario

To remove **one** scenario from HOMER, select this scenario in the window "HOMER: Scenario manager" (see fig. 366) and then click on the button "Delete".

Hint: Should there be only one scenario available, the button "Delete" is not available.

Before deleting the selected scenario, an appropriate security query will appear.

11.2   Defining the Structure of Acquisition Tables

The definition of the structure of the HOMER acquisition tables consists of fixing the parameter and creating the required attributes.

The **model parameters** (see chap. 11.2.1, p. 572) refer to the basic model information for instance the name of the library used, the name of the relation to create etc. This information will be used for the settings of HOMER.

The **object settings** (see chap. 11.2.2, p. 574) determine the features (attributes) of your activity. Some examples are for instance (specialised) descriptions, type of performer or also quantitative attributes such as performance and waiting times.

11.2.1 General Model Parameters

To adjust, the general model parameter of HOMER, select in the menu "Options" the menu item "Scenario settings". The window "HOMER: Scenario settings" (see fig. 373) will appear, in which you will carry out the following described settings and then save it by clicking on the "OK" button.

Register **"General configuration"**:

Figure 373: HOMER: scenario settings - general configuration

Enter in the field **"Basis library name"** the name of the BP or WE library, in which the modelling objects that you want to generate or to read, are defined.

Enter in the field **"Application library name"** the name of the application library, in which the modelling objects that you want to generate or to read, are defined.

> **Hint:** In the field "Name of the application library", it is only possible/required to have one entry, in case you want to administrate attribute profiles.

Enter in the field **"Relation name"** the name of the relation by which you want to connect the single objects to each other.

> **Hint:** If you don't want to generate a relation, enter the value "None".

Register **"Model settings"**:
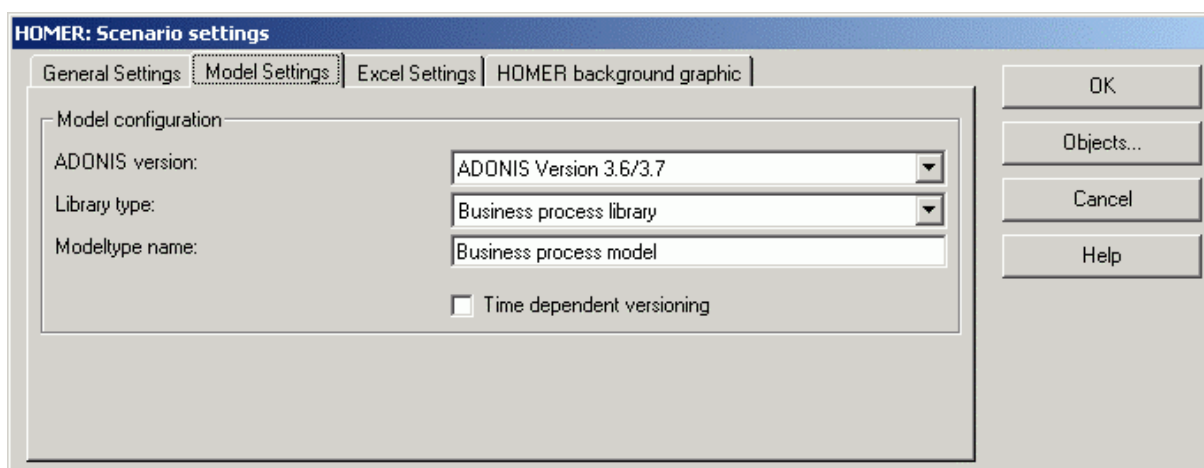
Figure 374: HOMER: scenario settings - model settings

Select from the list **"ADONIS version"** the version of ADONIS for which you want to create or read ADL files.

Select from the list **"library type"** the type from which you want to create or read ADL files.

Enter in the field **"model type name"** the name of the model type you want to generate or read.

> **Hint:** Model types can only be generated for ADONIS from version 3.0.

Click on the option **"time-related versioning"**, if you use an application library with activated time-related versioning (see chap. 6.1.2, p. 69).

> **Hint:** The time-related versioning can only be realised with ADONIS from version 3.5, this is why this field is inactive, when you have selected the option "ADONIS version 2.x" or "ADONIS version 3.x" in the list "ADONIS version".

Register **"Excel settings"**:
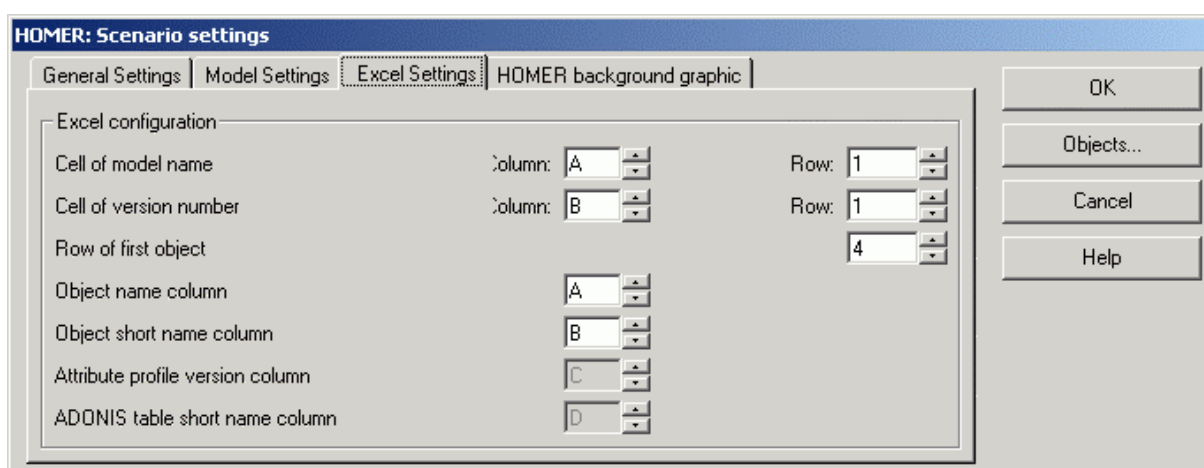
Figure 375: HOMER: scenario- settings - Excel settings

Please define the Excel fields, in which information from the generated Excel fields and specific for model and attribute profiles are found:

Cell of model name   **Column** and **row** containing the model name

Cell of version number	**Column** and **row** containing the model version
Row of first object	Row containing the first modelling objects
Object name column	Column containing the name of the modelling objects
Object short name column	Column containing the class short name of the modelling objects
Attribute profile version column	Column containing the version information for attribute profiles
ADONIS record short name column	Column containing the short names for ADONIS records

> **Hint:** The column of the attribute profile version can only be specified, if you have previously created an attribute profile. The column of the ADONIS table short name can only be specified, if you have previously created an ADONIS table .

Register **"HOMER background graphic"**:
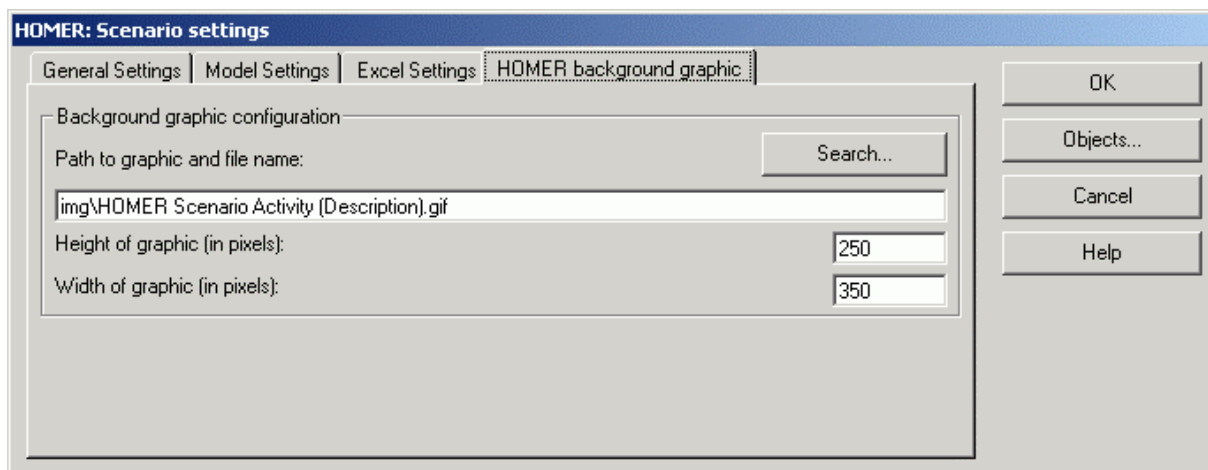
Figure 376: HOMER: scenario-settings - HOMER background graphic

Enter in the field **"Path to graphic and file name"** the path and the name of the graphic file you want to have displayed as background picture for the current scenario in the main HOMER window. Clicking on the button "Search" will provide you support for this.

> **ATTENTION:** Always use the absolute (full) path.

The following graphic formats are available as background graphic:

● Windows Bitmap (BMP)

● Windows Meta File (WMF)

Enter in the fields **"Height of the graphic (in pixels)"** and **" Width of the graphic (in pixels)"** the desired height and width of the selected graphic.

> **Hint:** The maximum height for a background graphic is 550 Pixels, the maximum width is 705 Pixels. Please note that graphics bigger than these limits will be cut.

11.2.2   Object Settings

To adapt the object settings of HOMER, select in the menu "Options" the menu item "Scenario settings". The window "HOMER: Scenario settings" (see fig. 373) will appear. Click on the button "Objects" of this window. The window "HOMER: Object settings" (see fig. 377) will appear. Alternatively

you can start from the window "HOMER: Scenario settings" (see fig. 373) and click on the button "Objects" to the window "HOMER: Object settings".
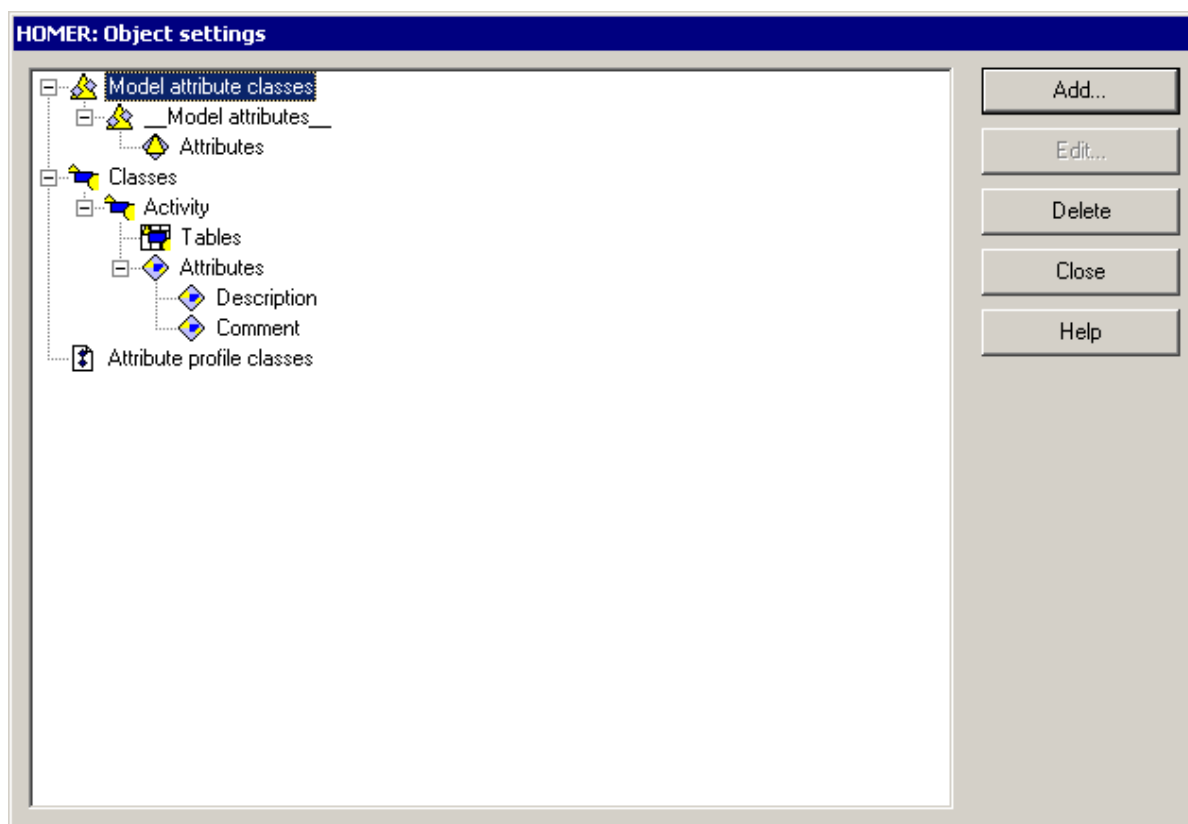
Figure 377: HOMER: classes

Starting from the window "HOMER: Object settings", you can perform the following operations:

● **Classes**

-   add (see chap. 11.2.2.1, p. 576),

-   edit (see chap. 11.2.2.2, p. 577) or

-   delete (see chap. 11.2.2.3, p. 577),

● **Attribute profile classes**

-   add (see chap. 11.2.2.4, p. 577),

-   edit (see chap. 11.2.2.5, p. 578) or

-   delete (see chap. 11.2.2.6, p. 578),

● **Records**

-   add (see chap. 11.2.2.7, p. 579),

-   edit (see chap. 11.2.2.8, p. 579) or

-   delete (see chap. 11.2.2.9, p. 580),

● **Attributes**

-   add (see chap. 11.2.2.10, p. 580),

-   edit (see chap. 11.2.2.13, p. 586) or

-   delete (see chap. 11.2.2.14, p. 588).

- **Model attribute classes**
  - add (see chap. 11.2.2.15, p. 588),
  - edit (see chap. 11.2.2.16, p. 589) or
  - delete (see chap. 11.2.2.17, p. 590),
- **Model attributes**
  - add (see chap. 11.2.2.18, p. 590),
  - edit (see chap. 11.2.2.19, p. 591) or
  - delete (see chap. 11.2.2.20, p. 591).

11.2.2.1 Adding Classes

Hint: The definition of a class can be found at "Terms and definitions" (see chap. 1., p. 28).

To create a class, click on the object "Classes" in the object list of the window "HOMER: Object settings" (see fig. 377) and then on the button "add". The window "HOMER: Add object" (see fig. 378) will appear.
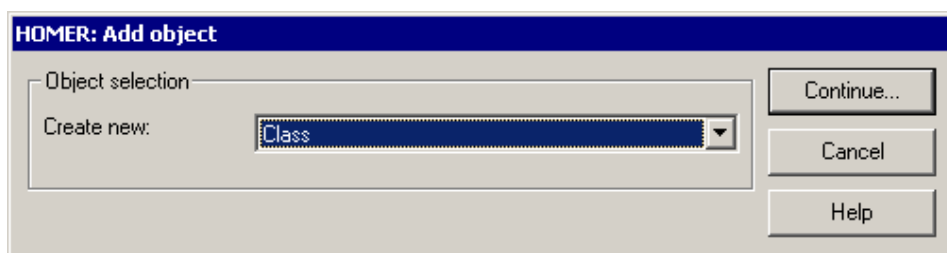
Figure 378: HOMER: add object

Select from the list **"Create new"** the object "class" and then click on the button "Continue". The window "HOMER: Create class" (see fig. 379) will appear.
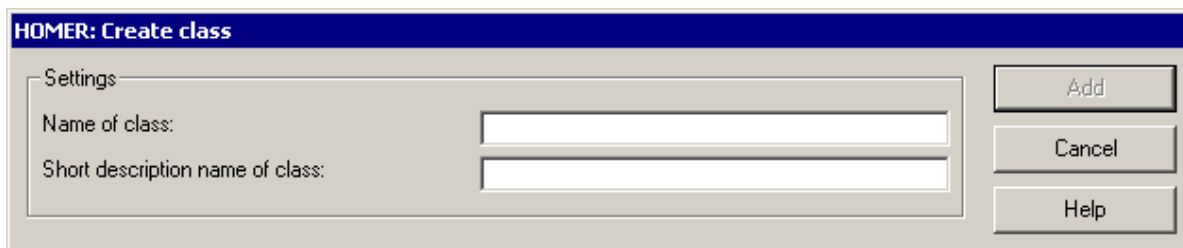
Figure 379: HOMER: add class

Enter in the field **"Name of class"** the name of the class of which objects you want to administrate in HOMER.

Enter in the field **"Short name of class"** any short name for the class of which objects you would like to administrate in HOMER.

ATTENTION: The name of the short name must be clear.

Inside the class, tables (see chap. 11.2.2.7, p. 579) and attributes (see chap. 11.2.2.10, p. 580) can be now created.

11.2.2.2 Editing Class Configuration

To change the configuration of a class, select the class to change from the object selection of the window "HOMER: Object settings" (see fig. 377) and then click on the button "Edit". The window "HOMER: Edit class" (see fig. 380) will appear.

Figure 380: HOMER: edit class

Enter in the field **"Name of class"** the name of the class of which objects you want to administrate in HOMER.

Enter in the field **"Short name of class"** the short name of the class of which objects you want to manage in HOMER.

ATTENTION: The name of the short term must be explicit.

11.2.2.3 Deleting Classes

To remove **one** class, first select it in the window "HOMER: Object settings" (see fig. 377) and then click on the button "delete".

To remove **all** classes, select in the window "HOMER: object settings" (see fig. 377) the object "Classes" and then click on the button "delete".

An appropriate security query will appear before the deletion of the selected classes.

11.2.2.4 Adding Attribute Profile Classes

Hint: The definition of the attribute profile can be found in the "Attribute profile" (see chap. 8., p. 77) section.

To create an attribute profile class, click on the object "Attribute profile classes" in the object list of the window "HOMER: object settings" (see fig. 377) and then on the button "Add". The window "HOMER: Add object" (see fig. 381) will appear.

Figure 381: HOMER: add object

Select the object "Attribute profile class" from the list **"Create new"** and then click on the button "Continue". The window "HOMER: Create attribute profile class" (see fig. 382) will appear.

Figure 382: HOMER: create attribute profile class

In the field **"Name of class"** enter the name of the attribute profile class of the objects you want to administrate in HOMER.

Enter in the field **"Short name of class"** any short name for the attribute profile class of which objects you would like to administrate in HOMER.

> **ATTENTION:** The name of the short description must be clear.

11.2.2.5 Editing Attribute Profile Classes Configuration

To change the configuration of an attribute profile class, select the attribute profile class to be changed in the object selection of the window "HOMER: object settings" (see fig. 377) and then click on the button "Edit". in the window "HOMER: Edit attribute profile class" (see fig. 383).

Figure 383: HOMER: edit attribute profile class

In the field **"Name of class"** enter the name of the attribute profile class of which you want to administrate in HOMER.

In the field **"Short name of class"** enter the short name for the attribute profile class of which objects you want to administrate in HOMER.

> **ATTENTION:** The name of the short description must be clear.

11.2.2.6 Deleting Attribute Profile Classes

In order to delete **one** attribute profile class, first select it in the window "HOMER: object settings" (see fig. 377) and then click on the button "delete".

To remove **all** attribute profile classes, select the object "Attribute profile classes" in the window "HOMER: object settings" (see fig. 377) and then click on the button "delete".

An appropriate security query will be displayed before the deletion of the selected attribute profile classes.

11.2.2.7 Adding Records

> **Hint:** You can find the definition of a record, or of a record attribute in the "Record attribute" (see chap. 9., p. 79).

To create a record (for ADONIS attributes of the type "RECORD"), select in the window "HOMER: object settings" (see fig. 377) a record object of a class ▦ or an attribute profile class ▣, for which you want to add this record and then click on the button "Add". The window "HOMER: add object" (see fig. 384) will appear.
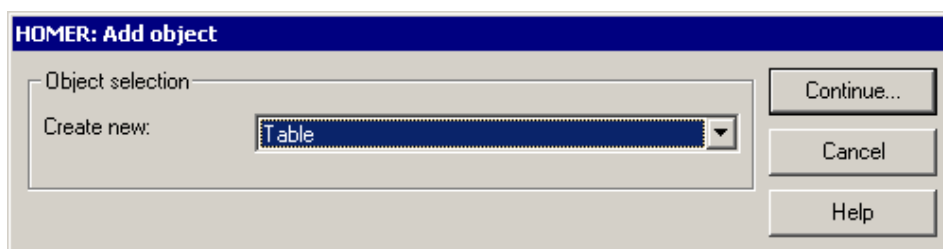
Figure 384: HOMER: add object

Select the object "Record" in the list **"Create new"** and then click on the button "Continue". The window "HOMER: create ADONIS record" (see fig. 385) will appear.
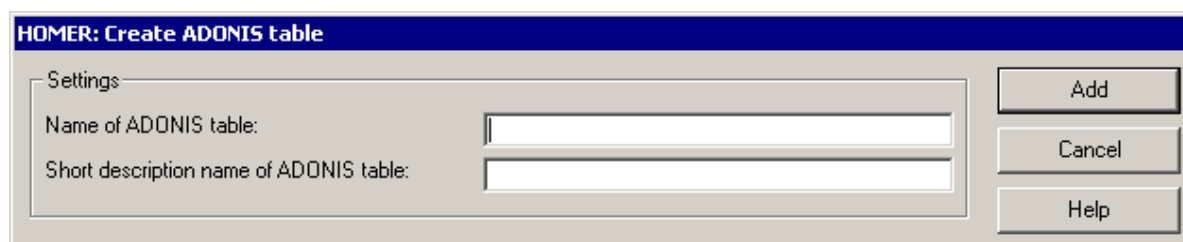
Figure 385: HOMER: add record

Enter in the field **"Name of ADONIS record"** the name of the record of which attributes you would like to administrate in HOMER.

> **Hint:** The name of the attribute of the class, which contains the reference to the record class must be specified.

Enter in the field **"Short name of the ADONIS record"**, the short name for this record.

11.2.2.8 Editing Records

To change the configuration of a record (for ADONIS attribute of type "RECORD"), select in the window "HOMER: object settings" (see fig. 377) the records to change and then click on the button "Edit". The window "HOMER: edit ADONIS record" (see fig. 386) will appear.
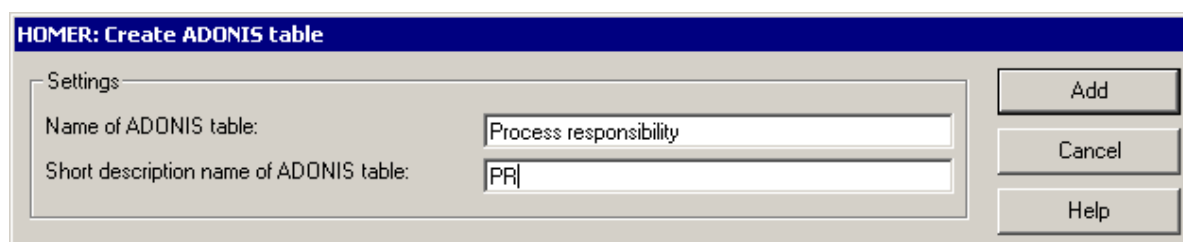
Figure 386: HOMER: edit record

Enter in the field **"Name of ADONIS record"** the name of the selected record.

> **Hint:** The name of the attribute of the class, which contains the reference to the record class, must be specified.

Enter in the field **"Short name of ADONIS record"** the short name for this record.

11.2.2.9 Deleting Records

To remove **one** record, first select it in the window "HOMER: object settings" (see fig. 377) and then click on the button "Delete".

To remove **all** records of a class or attribute profile classes, first select the object "Records" of the appropriate class or attribute profile class in the window "HOMER: object settings" (see fig. 377) and then click on the button "Delete".

An appropriate security query will be displayed before the deletion of the selected records.

11.2.2.10 Adding Attributes

> **Hint:** You can find the definition of an (object) attribute in the section "Terms and connections" (see chap. 1., p. 28).

To create an attribute or a model attribute, select a model attribute ◆, an attribute of a class ◆, an attribute profile class ⬧ or a record ⬧ and then click on the button "Add" in the window "HOMER: object settings" (see fig. 377). The window "HOMER: add object" (see fig. 387) will appear.
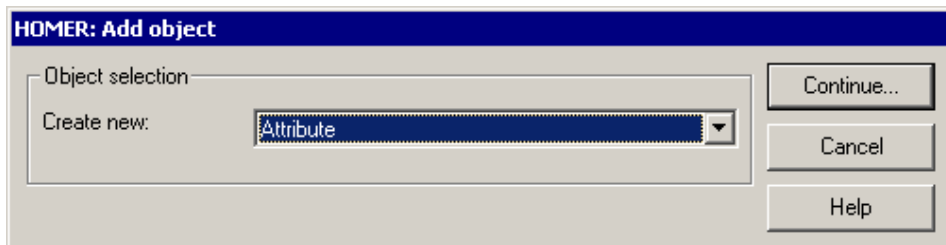
Figure 387: HOMER: configurate attributes

Select in the list **"Create new"** the object "Attribute" and then click on the button "Continue". The window "HOMER: create attribute " (see fig. 388) will be displayed.

Figure 388: HOMER: create attribute

Enter in the field **"Name of attribute"** the name of the attribute of what data you want to acquire in HOMER.

Select the type of the current attribute (see chap. 11.2.2.11, p. 583) in the list **"Type of attribute"**.

> **Hint:** The following fields and pull-down menus appear depending on the selected attribute type and task format. The figure below illustrates the membership of the attributes and their task possibilities

> **Hint:** In the following fields, the abbreviation "ref." stands for "referenced".

Attribute type	Attribute format	Attribute value column	Row breaking point	Separate sign	Column of referenced model(s)	Column of referenced model	Column of referenced object name	Column of referenced object	Column of ref. attribute profile	Default name of called program	Column of called file	Column of the form name	Form column
String	One line	X											
	Multiple lines	X	X										
Number	Integer	X											
	Integer	X											
Time	ADONIS (yy:dd:hh:mm:ss)	X											
	Year (Floating point number)	X											
	Day (Floating point number)	X											
	Hour (Floating point number)	X											
	Minute (Floating point number)	X											
	Second (Floating point number)	X											
	mm:ss	X											
	[m]:ss	X											
	[h]:mm	X											
	[h]:mm:ss	X											
Enumeration		X											
Enumeration list		X		X									
Model reference					X	X							
Object reference					X	X	X	X					
Attribute profile reference									X				
Programm call	(fixed program name)									X		X	
	(variable program name)										X	X	
Discrete random generator												X	X
Expression	One line	X											
	Multiple lines	X	X										
Date	ADONIS (yy:mm:dd)	X											
Date and time	ADONIS (yy:dd:hh:mm:ss)	X											

Figure 389: Membership of the attributes and their task possibilities

<u>**Explanation:**</u>

When choosing attribute type "string", the user has the possibility to choose between input formats "single row" and "many rows" as well as the ability to enter columns of attribute values and line break indicators.

> **Hint:** The field **"Next free column"** contains the name of the next free column (i.e. not assigned) - starting from the end of the record.

> **Hint:** The field **"Next free column"** is shrunk when creating model attributes, since they are not entered in columns but in cells.

> **ATTENTION:** All free columns between two assigned columns will be ignored.

Select in the field **"Typing format of attribute"** the format of the current attribute (see chap. 11.2.2.12, p. 584).

Enter in the field **"Column of attribute"** the name of the column, in which the current attribute should be.

> **ATTENTION:** If the entered column is not available, you have the possibility to move all attributes concerned to the right. Make sure for attributes with multiple columns, that you can separate the neighbouring columns of the attribute.

Enter in the field **"Separator"** a separator sign to separate possible enumeration values of the attribute.

Enter in the field **"Column of ref. model type name"** the name of the column in which the model type name of the referenced model/object should be.

Enter in the field **"Column of ref. model"** the name of the column in which the model name of the referenced model/object should be.

Enter in the field **"Column of ref. object type name"** the name of the column in which the name of the class of the object reference should be.

Enter in the field **"Column of ref. object name"** the name of the column in which the name of the object of the object reference should be.

Enter in the field **"Column of ref. attribute profile"** the name of the column in which the name of the referenced attribute profile should be.

Enter in the field **"Standard name of called program"** the name of the program that should be used by default for the called file.

> **Hint:** If you have selected the type "program call" in the list "type of attribute" and this field is not active, you must activate the radio button in front of the field name.

Enter in the field **"Column of called program"** the name of the column in which the program, that should be used with the called file, should be.

> **Hint:** If you have selected the type "program call" in the list "type of attribute" and this field is not active, you must activate the radio button in front of the field name.

Enter in the field **"Column of called file"** the name of the column in which the called file should be.This field is only relevant for the attribute type "program call".

> **Hint:** The file to call has to be entered with its path.

Enter in the field **"Column of domain name"**, the name of the column in which the name of each variable domain should be. This field is only relevant for the attribute type "Discrete variable setting".

Enter in the field **"Column of domain value"** the name of the column in which the variable domain should be.

11.2.2.11   Attribute Types in HOMER

The following attribute types will be directly supported by HOMER:

String   means that you can enter any text of one or several-lines in this attribute of ADONIS.

Number   means that you can enter whole numbers or floating points in this attribute of ADONIS.

Time   means that you can enter times in the ADONIS time format (YY:DDD:HH:MM:SS) in this attribute.

Enumeration   means that by using this attribute of ADONIS you can select **one** value from a list, click on **one** check box or mark **one** value over a radio button.

Enumeration list   means that you can select **several** values from a list over this attribute of ADONIS.

Model reference   means that you can reference one or more models over this attribute of ADONIS.

Object reference   means that you can reference one or more objects over this attribute of ADONIS.

Attribute profile reference   means that you can reference one or more attribute profiles over this attribute of ADONIS.

    ATTENTION: Attribute profile classes will be supported by ADONIS from version 3.5. If you have **not** selected the option "ADONIS version 3.5x" in the pull-down menu "ADONIS Version" in the general model parameters (see chap. 11.2.1, p. 572), then you cannot create attributes of type "Attribute profile reference".

Program call   means that you can reference one or more files and their programs over this attribute in ADONIS.

Discrete random generator   means that you can produce the **discrete** random generator for a branching of the process path over this attribute in ADONIS.

Expression   means that in ADONIS you can enter values with flexible formulas within this attribute.

Date   means that in ADONIS you can enter date values using the ADONIS date format (YY:DDD:HH:MM:SS) within this attribute.

Date and time   means that in ADONIS you can enter time values using the ADONIS date and time format (YYYY:MM:DD hh:mm:ss) within this attribute.

    Hint: If you require an attribute type not specified here, it is possible to define it as a string (text) and to enter the required ADL syntax manually in the acquisition table. Using an Excel macro (which has to be created), you can change the format of the attribute fields to the appropriate value, in order to create the partially large-scale ADL syntax in a proper way.

    Hint: Further information for the HOMER attribute types can be found in chapter "ADONIS attribute types" (see chap. 5., p. 536)

11.2.2.12   Attribute Formats in HOMER

The following formats are available depending on the selected attribute type:

1. *string*

    - **Single line**

        Set this format for any single line text.

        All symbols will be edited except line breaks.

        A maximum of 255 symbols are allowed in HOMER

    - **Multi line**

        Set this format for any multi line text.

        All symbols will be edited.

        A maximum of 255 symbols are allowed per cell in HOMER. If there are more symbols used, then the below-standing cell will be used.

    ATTENTION: **Line** breaks will be changed to **cell** breaks, i. e. the next deeper cell will be used.

    Note: The new line indicator is useful to differentiate, whether a new cell has started, because there is a line break in the original text or because the maximum number of symbols allowed per cell has been exceeded. If the new line indicator has the value "1", then a new line has started, because there is a line break in the original source attribute at this position. If the value is "0", a new cell has started, because the maximum number of symbols has already been reached.

2. *Number*

    - **Floating point number**

        Set this format for floating point numbers (e. g.: 0.9 or 1.0).

        Only numbers and one dot will be accepted.

        A maximum of 255 symbols are allowed in HOMER.

    - **Integer (whole numbers)**

        Set this format for floating point number (e. g.: 4 or 17).

        Only numbers will be edited.

        A maximum of 255 symbols are allowed in HOMER.

3. *Time*

    - **ADONIS (yy:ddd:hh:mm:ss)**

        Set this format for times you want to enter directly in the ADONIS time format.

        Enter the time value in the format "yy:ddd:hh:mm:ss" (Year:Days:Hours:Minutes:Seconds). Three days, five hours and ten seconds must be entered as follows: 00:003:05:00:10.

        A maximum of 15 symbols are allowed in HOMER.

    - **Year (floating point number)**

        Set this format, if you want to enter the time calculated in years as a floating point number. Three days, five hours and ten seconds must be entered as follows: 0.0087902714358193810

        Only numbers and one dot will be accepted.

        A maximum of 255 symbols are allowed in HOMER.

- **Day (floating point number)**

  Set this format, if you want to enter the time calculated in days as a floating point number. Three days, five hours and ten seconds must be entered as follows: 3,208449074

  Only numbers and one dot will be accepted.

  A maximum of 255 symbols are allowed in HOMER.

- **Hours (floating point number)**

  Set this format, if you want to enter the time calculated in hours as a floating point number. Three days, five hours and ten seconds must be entered as follows: 77,0027778

  Only numbers and one dot will be accepted.

  A maximum of 255 symbols are allowed in HOMER.

- **Minutes (floating point number)**

  Set this format, if you want to enter the time calculated in minutes as a floating point number. Three days, five hours and ten seconds must be entered as follows: 4620,16667

  Only numbers and one dot will be accepted.

  A maximum of 255 symbols are allowed in HOMER.

- **Seconds (floating point number)**

  Set this format, if you want to enter the time calculated in seconds as a floating point number. Three days, five hours and ten seconds must be entered as follows: 277210

  Only numbers and one dot will be accepted.

  A maximum of 255 symbols are allowed in HOMER.

- **mm:ss**

  Set this format, if all the times for this attribute are smaller than one hour and you want to enter minutes and seconds as whole numbers.

  Enter the minutes (two-digit) followed by a colon and the seconds (two-digit). Five minutes and ten seconds must be entered as follows: 05:10.

  A maximum of 5 symbols is allowed in HOMER.

- **[m]:ss**

  Set this format, if all the times for this attribute are **not** smaller than one second and you want to enter minutes and seconds as whole numbers.

  Enter the minutes followed by a colon and the seconds (two-digit). Three days, five hours and ten minutes must be entered as follows: 4630:13

ATTENTION: Note that for this format all times greater than 60 minutes must be calculated in minutes, i.e. three days corresponds for instance to 4620 minutes.

  A maximum of 255 characters are allowed in HOMER.

- **[h]:mm**

  Set this format, if all the times for this attribute are **not** smaller than one minute and you want to enter hours and minutes as whole numbers.

  Enter the hours followed by a colon and the minutes (two-digit). Three days, five hours and ten minutes must be entered as follows: 77:10

ATTENTION: Note that for this format all times greater than 24 hours must be calculated in hours, i.e. three days corresponds for instance to 72 hours.

  A maximum of 255 characters are allowed in HOMER.

- **[h]:mm:ss**

   Set this format, if you want to enter hours, minutes and seconds as whole numbers.

   Enter the hours followed by a colon and the minutes (two-digit), followed by a colon and the seconds (two-digit). Three days, five hours, ten minutes and thirteen seconds must be entered as follows: 77:10:13

ATTENTION: Note that for this format all times greater than 24 hours must be calculated in hours, i. e. three days correspond for instance to 72 hours.

   A maximum of 255 characters are allowed in HOMER.

Hint: If you require an attribute type not specified here, it is possible to define it as a string (text) and to enter the required ADL syntax manually in the acquisition table. Using an Excel macro (which has to be created), you can change the format of attribute fields to the appropriate value, in order to create the large ADL syntax in the correct way.

11.2.2.13   Modifying Attributes

To change the configuration of an attribute, select the attribute to be changed in the window "HOMER: object settings" (see fig. 377) and then click on the button "Edit". The window "HOMER: edit attribute" (see fig. 390) will appear.

Figure 390: HOMER: edit attributes

Enter in the field **"Name of attribute"** the name of the attribute of which data you want to acquire in HOMER.

Select from the list **"Type of attribute"** the type of the current attribute (see chap. 11.2.2.11, p. 583).

Hint: The following fields and pull-down menus appear depending on the selected attribute type and task format. The figure below illustrates the membership of the attributes and their task possibilities

Hint: In the following fields, the abbreviation "ref." stands for "referenced".

Attribute type	Attribute format	Attribute value column	Row breaking point	Separate sign	Column of referenced model ty	Column of referenced model	Column of referenced object name	Column of referenced object	Column of ref. attribute profile	Default name of called program	Column of called file	Column of the form name	Form column
String	One line	X											
	Multiple lines	X	X										
Number	Integer	X											
	Integer	X											
Time	ADONIS (yy:dd:hh:mm:ss)	X											
	Year (Floating point number)	X											
	Day (Floating point number)	X											
	Hour (Floating point number)	X											
	Minute (Floating point number)	X											
	Second (Floating point number)	X											
	mm:ss	X											
	[m]:ss	X											
	[h]:mm	X											
	[h]:mm:ss	X											
Enumeration		X											
Enumeration list		X		X									
Model reference					X	X							
Object reference					X	X	X	X					
Attribute profile reference									X				
Programm call	(fixed program name)									X	X		
	(variable program name)									X	X		
Discrete random generator												X	X
Expression	One line	X											
	Multiple lines	X	X										
Date	ADONIS (yy:mm:dd)	X											
Date and time	ADONIS (yy:dd:hh:mm:ss)	X											

Figure 391: Membership of the attributes and their task possibilities

Explanation:

When selecting the attribute type "String" you have the choice of two input formats "Single line" and "Multi line" as well as the option to define the columns for the attribute value and the line-break indicator.

> **Hint:** The field **"Next free column"** contains the name of the next available (i.e. not already used) column.

> **Hint:** The field **"Next free column"** is not available when creating model attributes, as these are not maintained in columns but in cells.

> **ATTENTION:**   Free columns between two already used columns will be ignored.

Select in the field **"Format of attribute"** the format of the current attribute (see chap. 11.2.2.12, p. 584).

Enter in the field **"Column of attribute"** the name of the column, in which the current attribute should be.

> **ATTENTION:**   If the entered column is not available, you have the possibility to move all attributes concerned to the right. Insure that for attributes with multiple columns, you can separate the neighbouring columns of the attribute.

Enter in the field **"Separator"** the character used to separate the possible enumeration values of the attribute.

Enter in the field **"Column of ref. model type name"** the name of the column in which the model type name of the referenced model/object should be.

Enter in the field **"Column of ref. model"** the name of the column in which the model name of the referenced model/object should be.

Enter in the field **"Column of ref. object type name"** the name of the column in which the name of the class of the object reference should be. This field is only relevant for the attribute type "object reference".

Enter in the field **"Column of ref. object name"** the name of the column in which the name of the object reference should be.

Enter in the field **"Column of ref. attribute profile"** the name of the column in which the name of the referenced attribute profile should be.

Enter in the field **"Standard name of called program"** the name of the program, which should be used by default for the called file.

> **Hint:** If you have selected the type "program call" in the list "type of attribute" and this field is not active, you must activate the radio button in front of the field name.

Enter in the field **"Column of called program"** the name of the column in which the program, that should be used with the called file, should be.

> **Hint:** If you have selected the type "program call" in the list "type of attribute" and this field is not active, you must activate the radio button in front of the field name.

Enter in the field **"Column of called file"** the name of the column in which the called file should be.

> **Hint:** The file to call has to be entered with its path.

Enter in the field **"Column of domain name"**, the name of the column in which the name of each variable domain should be.

Enter in the field **"Column of domain value"** the name of the column in which the variable domain should be.

11.2.2.14   Deleting Attributes

To remove **one** attribute from HOMER, first select it in the window "HOMER: Object settings" (see fig. 377) and then click on the button "Delete".

To remove **all** attributes from HOMER, select in the window "HOMER: object settings" (see fig. 377) the object "Attributes" of any object and then click on the button "Delete".

Before deleting the selected scenario, an appropriate security query will appear.

11.2.2.15   Adding Model Attribute Classes

To create a model attribute class, click on the object "Model attribute class" in the object list of the window "HOMER: object settings" (see fig. 377) and then on the button "Add". The window "HOMER: add object" (see fig. 378) will appear.
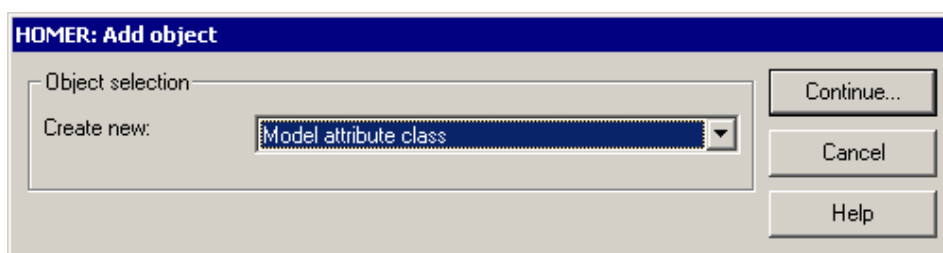
Figure 392: HOMER: add object

Select from the list **"Create new"** the object "model attribute class" and then click on the button "Continue". The window: "HOMER: create model attribute class" (see fig. 393) will appear.
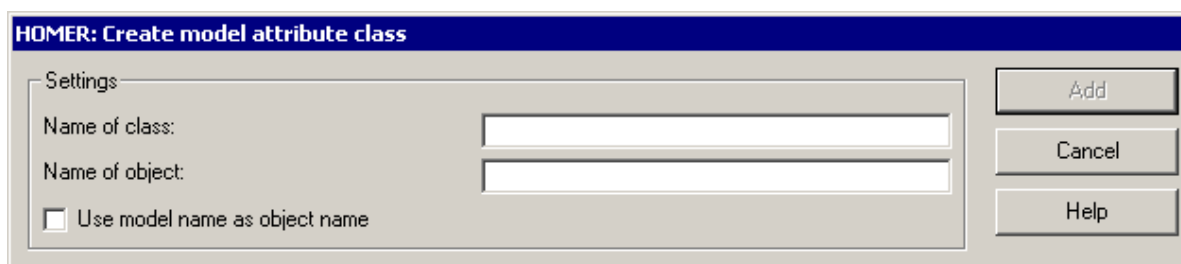
Figure 393: HOMER: create model attribute classes

Enter in the field **"Name of class"** the name of the class of the objects you would like to administrate in HOMER.

Enter in the field **"Name of object"** the name of the object to create which should contain the value for the model attribute.

In case you want to use, for the object name, the name of the model where the current object is, click on the option **"Use model name as object name"**.

11.2.2.16   Editing Model Attribute Class Configuration

To change the configuration of a model attribute class, select the model attribute class you want to change in the object selection of the window "HOMER: object settings" (see fig. 377) and then click on the button "Edit" in the window "HOMER: edit model attribute class" (see fig. 394) will appear.
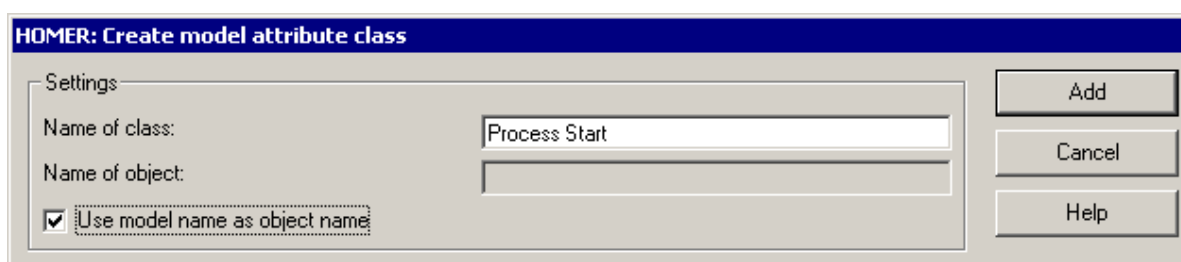
Figure 394: HOMER: edit model attribute class

Enter in the field **"Name of class"** the name of the model attribute class of which objects you would like to administrate in HOMER.

Change in the field **"Name of object"** the name for the object you want to create, which should contain the value for the model attribute .

In case you want to use the name of the model where the current object is as the object name, click the option **"Use model name as object name"**.

11.2.2.17   Deleting Model Attribute Classes

To remove **one** model attribute class from HOMER, first select them in the window "HOMER: Object settings" (see fig. 377) and then click on the button "Delete".

To remove **all** model attribute classes from HOMER, select in the window "HOMER: object settings" (see fig. 377) the object "Attributes" of any object and then click on the button "Delete".

Before deleting the selected scenario, an appropriate security query will appear.

11.2.2.18   Adding Model Attributes

> **Hint:** The definition of the model attributes can be found in "the terms and context" (see chap. 1., p. 28) chapter.

To create a model attribute, select a model attribute and click on the button "Add" in the window "HOMER: object settings" (see fig. 377). The window "HOMER: add object": (see fig. 395) will appear.
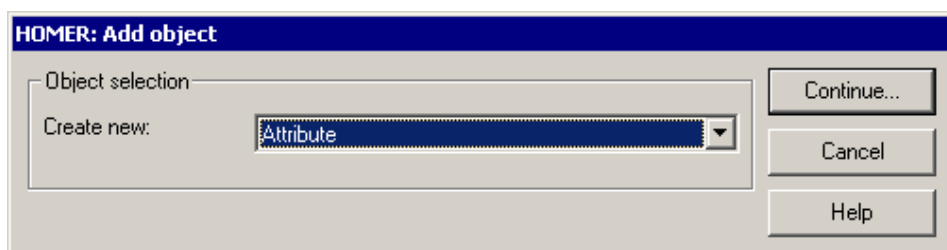
Figure 395: HOMER: configurate attributes

Select from the list **"Create new"** the object "Attribute" and then click on the button "Continue". The window: "HOMER: create attribute" (see fig. 393) will appear.
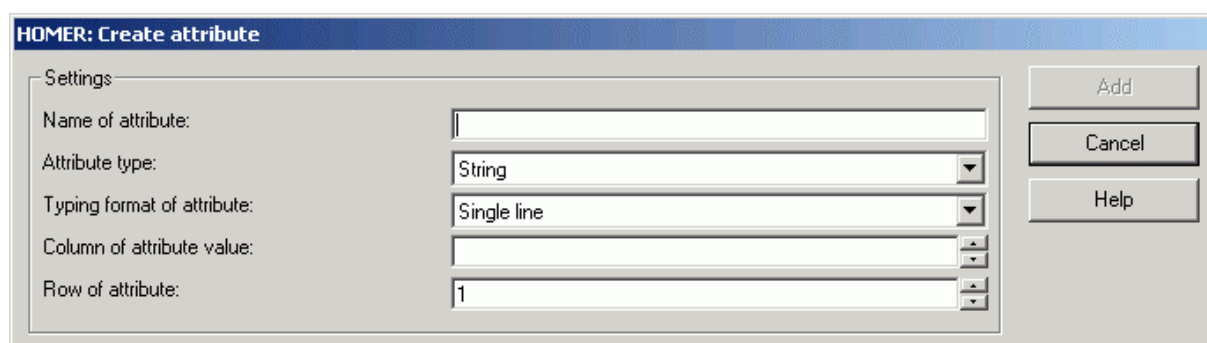
Figure 396: HOMER: create model attribute

Enter in the field **"Name of attribute"** the name of the model attribute you want to acquire in HOMER.

Enter in the list **"Type of attribute"** the type of the current attribute (see chap. 11.2.2.11, p. 583).

> **Hint:** The following fields and pull-down menus appear depending on the selected attribute type and will be listed in the order of the attribute type dependence, i.e. first the field of the attribute type "String" in the order of its appearance and then those of the attribues of type "number" etc.

Select in the field **"Typing format of attribute"** the format of the current attribute (see chap. 11.2.2.12, p. 584). This field is only relevant for attributes of types "String", "Number" and "Time".

Enter in the field **"Column of attribute"** the name of the column for the current attribute.

Enter in the field **"Row of attribute"** the number of the row for the current attribute.

11.2.2.19 Changing Model Attributes

To change the configuration of a model attribute, first select it in the window "HOMER: object settings" (see fig. 377) and then click on the button "Edit". the window "HOMER: edit attribute" (see fig. 397) will appear.

Figure 397: HOMER: edit model attributes

Enter in the field **"Name of attribute"** the name of the model attribute value you want to acquire in HOMER.

Select from the list **"Type of attribute"** the type of the current attribute (see chap. 11.2.2.11, p. 583).

> **Hint:** The following fields and lists appear depending on the selected attribute type and will be listed in the order of the attribute type dependence, i.e. first the field of the attribute type "String" in the order of its appearance and then those of the attribute type "number" etc.

Select in the field **"Typing format of attribute"** the format of the current attribute (see chap. 11.2.2.12, p. 584). This field is only relevant for attributes of type "String", "Number" and "Time".

Enter in the field **"Column of attribute value"** the name of the column, in which the current attribute should be.

Enter in the field **"Row of attribute"** the number of the row, in which the current model attribute should be.

11.2.2.20 Deleting Model Attributes

To remove **one** model attribute from HOMER, first select it in the window "HOMER: object settings" (see fig. 377) and then click on the button "Delete".

To remove **all** model attributes from HOMER, select in the window "HOMER: object settings" (see fig. 377) the object "Attribute" for any object and then click on the button "Delete".

Before deleting the selected scenario, an appropriate security message will appear.

12. ADONIS Query Language (AQL)

The ADONIS query language AQL (**A**DONIS **Q**uery **L**anguage) enables you to run queries on Business Process Models or Working Environment models. The result of an AQL expression is a set of objects or connectors which meet the search criteria you specified.

AQL is used in the following places in the ADONIS Administration Toolkit:

● when editing the library attributes of BP and WE libraries in order to define predefined queries

● when editing the library attributes of BP and WE libraries in order to define predefined evaluation queries.

● when entering selection criteria for model search according to model attributes in the Model Management component (see chap. 2., p. 125) for the purpose of deleting models.

(See examples 1, 2, 8, 12, 13, 14 and 15 for **BP Models** or examples 3, 4, 5, 6, 7, 9, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27 and 28 for **WE Models**.)

The query language AQL contains static elements which can be evaluated at any time in the Analysis Component. Static elements are allowed to be assigned in the simulation only if one or more performers are removed from the used Working Environment model. Dynamic elements are evaluated during the capacity and workload analysis via the performer assignment. Additionally the dynamic elements allow the access to variable values or performers of previous activities during the simulation.

> **Hint:** All explanations and descriptions refer to models based on the **ADONIS standard application library** (see chap. 20., p. 693).

12.1 Syntax and Semantics of AQL

Extended Backus Naur Form (EBNF) Notation

The EBNF notation specifies by so-called (production) rules which expressions meet the requirements of the AQL syntax.

Non-terminal symbols are included in <...> and serve to formulate the rules (see chap. 12.2, p. 595). Each rule starts with a non-terminal symbol, followed by "::=" and the symbol's definition. The non-terminal symbols occurring in the definitions are defined in other rules.

Terminal symbols (= symbols which cannot be split up further) are included by inverted commas and are interpreted directly.

The symbols {...}, [...] and | serve to formulate rules in a more compact form.

Summary of the EBNF notation:

<...>	non-terminal symbol
'...'	terminal symbol
{...}	arbitrary number of iterations (even 0-times)
[...]	optional (0- or 1-time)
|	alternative

Terminal symbols and key words

The syntax of AQL requires that names of classes, objects, relations, attributes, variables and alpha-numerical constants are denoted by inverted commas. White-spaces of any kind (blank, tabulator or return) are allowed in any place where they make sense.

<Class> ::=	The name of a class or relation. The classes and relations such as "Process start", "Activity", "Decision", "Subsequent" and so on are available to the user via the modelling panel in the model editor, so that he can design business process models. The model, that contains a class or relation, arises from the context of the analysed or simulated models. If another model is desired (i.e. because a class or relation should be referenced in particular model), it has to be entered explicitly (through names and model types):
	<Class>':'<Model name>':'<Model type>
<Object> ::=	The name of an object within a concrete business process or working environment model. "Write letter", for example, could be the name of an object of the class "Activity". Should objects be ambiguous within a model, the name of the class has to be appended to the object's name as in <Object>':'<Class>.The model, that contains a class or relation, arises from the context of the analysed or simulated models. If another model is desired (i.e. because a class or relation should be referenced in particular model), so it has to be entered explicitly (through names and model types):
	<Object>':'<Model name>':'<Model type>, or
	<Object>':'<Class>':'<Model name>':'<Model type>
<Relation> ::=	The name of a relation, e.g. "Subsequent" in business process models or "Has role" in working environment models.
<Attribute> ::=	The name of an attribute, e.g. the class "Activity" has - among others - the attribute "Execution time".
<Operator> ::=	'>' | '>=' | '=>' | '=' | '<=' | '=<' | '<' |'!='| 'like' | 'unlike' are comparison operators. 'like' and 'unlike' are used for alphanumerical signs and allow the use of wild cards (? and/or *) in symbol chains. The other operators are used for comparing numerical values.
	The operator 'contains' is available where complex attribute values (e.g. inter-model-references) can be interrogated. The operator represents on the one hand, in the sense of a set operation, something comparable with a set inclusion (subset feature), on the other hand the adherence will be expressed this way in the sense of a subtext. However this operator is not an AQL key word!
<Value> ::=	<Constant> | '!' <Variable> | '?' <Attribute>
<Constant> ::=	A constant expression such as 1000, Yes or others.

Logical Operators serve to link two (or more) AQL expressions:

<Logical Operator> ::=   'AND' \| 'OR' \| 'DIFF'

- 'AND' links two AQL expressions. The result is the intersection set of the two expressions, i.e. the result consists of all those objects which are both the result of the first and the result of the second expression.

- 'OR' links two AQL expressions. The result is the union set of the two expressions, i.e. the result consists of all objects which are either the result of the first or the result of the second expression or both.

- 'DIFF' links two AQL expressions. The result is the difference of the two expressions, i.e. the result consists of all those objects which are the result of the first expression, but not the result of the second expression.

'AND' links more strongly than 'OR' and 'OR' more strongly than 'DIFF', i.e. the expression "a DIFF b OR c AND d" is evaluated correctly like this: a DIFF (b OR (c AND d)).

12.2   Rules for formulating AQL expressions

The following rules are used while formulating AQL expressions (Analysis Component and static performer assignment of the Simulation Component) or dynamic performer assignment (Simulation Component):

Rule 1:   **AQL expression> ::= <AQL expression> {'AND' | 'OR' | 'DIFF' <AQL expression>}**

Each expression can be linked to one or more expressions by logical operators.

See examples 16, 20, 29, 30, 31, 32 and 34.

Rule 2:   **<AQL expression> ::= '(' <AQL expression> ')'**

Expressions can have parentheses.

See examples 12, 16, 20, 22, 29, 30, 31, 32 and 34.

Rule 3:   **<AQL expression> ::= '{' <object> [',' <object>] '}'**

The result is a set of objects. These are not necessarily derived from a class.

See examples 1, 2. As far as the allocation of resources to activities is concerned, see example 36.

Rule 4:   **<AQL expression> ::= '<' <class> '>'**

The result is all objects of the class specified.

See examples 3, 4, 5, 17, 18, 19, 20, 21, 23, 27, 28, 31, 33, 39 and 40.

Rule 5:   **<AQL expression> ::= <AQL expression> '->' <Class>**

The result is all objects which are of the specified class.

See example 11.

Rule 6:   **<AQL expression> ::= <AQL expression> '->' <Relation>**

The result is all objects which are linked as direct targets of the given relation with at least one object from the AQL expression.

See example 6.

Rule 7:   **<AQL expression> ::= <AQL expression> '<-' <Relation>**

The result is all objects which are linked as a direct start object of the relation specified with at least one object from the AQL expression.

See examples 7 and 8.

Rule 8:   **<AQL expression> ::= <AQL expression> '->>' <Relation>**

The result is all objects which are linked transitively as a target object of the relation specified with at least one object from the AQL expression.

See example 12.

Rule 9:   **<AQL expression> ::= <AQL expression> '<<-' <Relation>**

The result is all objects which are linked transitively as a start object of the relation specified with at least one object from the AQL expression.

See examples 9, 10 and 11.

Rule 10: **<AQL expression> ::= <AQL expression> '->' '<' <Relation> '>'**

The results are all connectors of the specified relation which have one of the specified objects as a start object.

See example 13.

Rule 11: **<AQL expression> ::= <AQL expression> '<-' '<' <Relation> '>'**

The results are all connectors of the specified relation which have one of the specified objects as an end point.

See example 14.

Rule 12: **<AQL expression> ::= <AQL expression> '[' <Value> <Operator> <Value> ']'**

The result is all objects of the start query, which attributes fulfil the defined criteria. Constants (numbers, string constants) can only be at the right of the operator. At the left of the operator, there are only attributes or variable references.

See examples 17, 18, 19, 20, 31 and 33.

Note: Queries with variable references as dynamic components in the performer assignment are only allowed in the simulation. See example 16.

Rule 13: **<AQL expression> ::= <AQL expression> '[' <Value> ']' '[' <Value> <Operator> <Value> ']'**

The result is all objects of the start query where their record attribute profile attribute fulfils the defined criteria. The first value is an attribute reference and specifies the name of the record or attribute profile attribute. In the second expression, constants (numbers, string constants) can only be at the right side of the operator. At the left side of the operator, there are only attributes or variable references.

Note: In case of a record attribute, the criteria is always fulfilled, if at least a table row of the record attribute meets the defined criteria.

See example 21 for record attributes and 41 for attribute profile attributes.

Rule 14: **<AQL expression> ::= 'done by' <Activity>**

Performer assignment. The result is the performer who has carried out the activity during a simulation run. This way, you will ensure, that an activity will be carried out by the same performer as defined in the performer assignment for this activity. The dynamic performer assignment is useful only for the simulation and is only possible within a Business Process Model.

See example 15.

Rule 15: **<AQL expression> ::= 'done by' <Variable>**

Performer assignment. This dynamic performer assignment can be used in an inter-model assignment . The result is the performer who has carried out the activity referenced through the variable during a simulation run. To use this construct it is necessary to enter the name of the variable in the attribute "done by" of the referenced activity.

See example 35.

Rule 16: **<AQL expression> ::= 'current performer' '->' <Resources relation>**

Resources assignment. This way you will ensure that the performer who carries out the current activity uses the resources given to him (resources assignment to performer).

See example 37.

Rule 17: **<AQL expression> ::= <AQL expression> '-->' <Attribute>**

The result is all objects which are referenced in the specified attribute of the object.

See example 39.

Rule 18: **\<AQL expression\> ::= \<AQL expression\> '--\>\>' \<Attribute\>**

The result is all objects which are transitively referenced in the specified attribute of the class.

Rule 19: **\<AQL expression\> ::=\<AQL expression\> '\<--'**

The result is all objects which refer the specified object.

See example 40.

12.3 AQL Examples

Hint: The AQL expressions in the examples listed below contain line-breaks. These are not necessary, they are just used to improve the readability of the examples.

Example 1

List all the objects "Request accepted?", "Work on request" and "Take holiday request" of a model:

AQL expression: `{"Request accepted?","Work on request",`
`"Take holiday request"}`

Example 2

List all the objects "Request accepted?", "Work on request" of the model "Holiday request":

AQL expression: `{"Request accepted?":"Holiday request"`
`:"Business Process Model","Work on request"`
`:"Holiday request":"Business Process Model"}`

Example 3

List all the activities of a Business Process Model:

AQL expression: `<"Activity">`

Example 4

List all the activities of the model "Holiday request":

AQL expression: `<"Activity":"Holiday request":"Business Process Model">`

Example 5

List all the roles of a Working Environment model:

AQL expression: `<"Role">`

Example 6

List all the roles of the performer "Maier":

AQL expression: `{"Maier"}->"Has role"`

Example 7

List all the performers who belong to the organisational unit "Distribution":

AQL expression: `{"Distibution"}<-"Belongs to"`

Example 8

List all the performers who have the role "Clerk":

AQL expression: `{"Clerk"}<-"Has role"`

Example 9

Display the whole organisation (i.e. all the organisational units), if it is hierarchically structured and if the organisational unit "Management" is in the upper hierarchy level:

AQL expression: `{"Management"}<<-"is subordinated"`

Example 10

List all the objects within a Business Process Model which are connected with the relation "Subsequent":

AQL expression: `<"End"><<-"Subsequent"`

Example 11

List all the objects of the class Activity within a Business Process Model which are connected with the relation "Subsequent":

AQL expression: `<"End"><<-"Subsequent" >"Activity"<`

Example 12

List all the performers who belong to the organisational unit "Distribution" or an organisational unit subordinated to the distribution:

AQL expression: `({"Distribution"}->>"Is subordinated") <- "Belongs to"`

Example 13

List all the connectors of the class "subsequent" within a Business Process Model which start from objects of the class "Decision":

AQL expression: `<"Decision"> -><"Subsequent">`

Example 14

List all the connectors of the class "Has role" of a Working Environment model which lead to the role "Clerk":

AQL expression: `{"Clerk"} <- <"has role">`

Example 15

During the simulation, the activity "Write letter" should be carried out by the same performer, which has executed the activity "Fill form". Enter the following in the attribute "Performer" of the activity "Write letter":

AQL expression (Performer assignment):

```
Has done "Fill form"
```

Example 16

Example of a variable reference

During the simulation, the activity "Control" should be done by different performers depending on the variables "Credit level": 1. for a credit level above 500.000 by the manager of the organisational unit "Credit department", 2. for a credit level under or equal to 500.000 by performers with the role "Clerk".

Enter the following AQL expression for Activity attribute "Performer" of the Activity "Control":

AQL expression (Performer assignment):

```
(({"Clerk"} <- "Has role")
[!"Credit level" <= 500000]) OR
(({"Credit department"} <- "Is manager")
[!"Credit level" > 500000])
```

Example 17

List all the activities with execution time of five minutes:

AQL expression: `<"Activity">[?"execution time" like "00:000:00:05:00"]`

Example 18

List all the activities for which the attribute "Description" contains no entry:

AQL expression: `<"Activity">[?"Description" like ""]`

Example 19

List all the variables that contain the variable type "Float":

AQL expression: `<"Variable">[?"Variable type" like "Float"]`

or: `<"Variable">[?"Variable type" like "G*"]`

(Since the attribute variable type can only be assigned with "Float" or "Enumeration", the first letter is enough for the entry.)

Example 20

List all the activities which are carried out by clerks and for which the activity costs are at least 10:

AQL expression: `(<"Activity">[?"Performer" like "*Clerk*"]) AND`
`(<"Activity">[?"Costs" >= 10])`

(Only the activities which are carried out by performers who contain the text "Clerk" in exactly this spelling in the role description. "*clerk*" will not work for this query!)

Example 21

List all the objects of the class "Process start", which have entered the value "Major responsible" in any table row of the record attribute "Process ownership" in the attribute "Classification" :

AQL expression: `<"Process start">[?"Process ownership"]`
`[?"Classification" = "Major responsible"]`

Example 22

List all the performers with the role "Clerk" who do not belong to the organisational unit "Distribution":

AQL expression: `({"Clerk"}<-"Has role") DIFF`
`({"Distribution"}<-"Belongs to")`

Example 23

List all the performers of a Working Environment model:

AQL expression: `<"Performer">`

Example 24

List all the roles of the performers "Maier", "Sommer" and "Winter":

AQL expression: `{"Maier", "Sommer", "Winter"} -> "Has role"`

Example 25

List all the oganisational units which are managed by Mr. Müller:

AQL expression: `{"Müller"} -> "Is manager"`

Example 26

List all the organisational units, for which the organisational unit "Department" is subordinated:

AQL expression: `{"Department"} -> "Is subordinated"`

Example 27

List all the managers of the organisational unit "Department":

AQL expression: `<"Department"> <- "Is manager"`

Example 28

List all the employees who have a role:

AQL expression: `<"Role"> <- "Has role"`

Example 29

List all the performers who have the role "Clerk" and belong to the organisational unit "Department":

AQL expression: `({"Clerk"} <- "Has role") AND`
`({"Department"} <- "Belongs to")`

Example 30

List all the performers who have the role "Clerk" and belong to the organisational unit "Org" or who are managers of the organisational unit "Org":

AQL expression: `(({"Clerk"} <- "Has role") AND`
`({"Org"} <- "Belongs to")) OR`
`({"Org"} <- "Is manager")`

Example 31

List of all the performers whose name starts with "M" and who's hourly wages are not higher than 20:

AQL expression: `(<"Performer">[?"Name" like "M*"]) AND`
`(<"Performer">[?"Hourly wages" <= 20])`

Example 32

List all the organisational units and roles which have the entry "Test" in the attribute "Description":

AQL expression: `(<"Organizational unit"> OR`
`<"Role">)[?"Description" = "Test"]`

Example 33

List all the performers whose name contains six letters and the fifth letter is an "e":

AQL expression: `<"Performer">[?"Name" like "????e?"]`

Example 34

List all the performers who have the role "Clerk" and belong to the organisational unit "Org" or who are manager of the organisational unit "Org", less the performers who have the role "Secretary" or who are manager of this organisational unit:

AQL expression: `(((({"Performer"} <- "Has role") AND`
`({"Org"} <- "Belongs to")) OR`
`({"Org"} <- "Is manager")) DIFF`
`((({"Secretary"} <- "Has role") OR`
`({"Org"} <- "Is manager"))`

Example 35

During the simulation, the Activity "Write letter" must be carried out by the performer who has executed the Activity "Fill form". In the attribute "Done by" of the Activity "Fill form", define e.g. the variable "Maier". In the Activity "Write letter", enter the following expression in the performer assignment:

AQL expression (Performer assignment):

`Done by "Maier"`

Example 36

To carry out the current activity, the resource defined in the Working Environment, e.g. a colour printer, must be used. For this Activity enter the following AQL expression in the resource assignment:

AQL expression (resource assignment):

`{"colour printer"}`

Example 37

The performer who carries out the current activity must use the reources that are assigned to him, e.g. his PC. Enter the following AQL expression in the resource assignement of this Activity:

AQL expression (resource assignement):

`Current performer -> "Uses resource"`

Example 38

All the Business Process Models which have been changed up to the 1st January 2001.

AQL expression: `<"Business Process Model">`
`[?"Last change on" like "01.01.2001*"]`

Example 39

All objects of the class "Organizational unit" which are refered to in aBusiness Process Model.

AQL expression: `<"Activity"> --> "ORG unit"`

Example 40

All the objects which refer to an object of the class "Role".

AQL expression: `<"Role"> <--`

Example 41

All the performers who are working at least 3 days a week. (The attribute "Days per week" is defined in the attribute profile attribute "Presence" .)

AQL expression: `<"Perfomer"> [?"Presence"][?"Days per week" >= 3]`

13. Update Catalogue Statistics of Database

Whenever application libraries or a large number of models have been imported, the DB catalogue statistics should be updated. We also recommend that this is carried out at regular intervals after a large number of models have been created or deleted.

The catalogue statistics contain static data about the distribution of tabular and index values which are applied by the database system to optimise queries. Updated or refreshed catalogue statistics can lead to more efficient execution plans and therefore to distinctly shorter response times in ADONIS applications.

In particular, after libraries or models have been imported into an ADONIS database for the first time, the recalculation (update) of the statistics can distinctly improve the system's performance.

If you wish to refresh the DB catalogue statistics, select the option "Update catalogue statistics of database" in the "Extras" menu.

> **ATTENTION:** No other ADONIS user should be logged into the database while the catalogue statistics are being updated.

> **Hint:** Updating the DB catalogue statistics may take some time.

14. Create a Database-selective List

Using a database selection drop-down list you can view all the ADONIS databases available in the ADONIS database Server as an INI file. The advantage is that during the login to ADONIS, a list of all the available databases will be provided and the user can select a database from this list.

The availability of such a list is indicated by the symbol at the end of the input field "Database name".

Create the database selection list by creating a file with the name `adblist.ini` in the ADONIS installation directory and include the names of the databases for the list as follows:

```
<Database name1>;
<Database name2>;
...
```

> **Hint:** Each database name has to be entered into a separate line and the line has to end with a ";" (semi-colon). It is recommended to avoid empty lines.

15. Save External Files to the ADONIS Database

By saving external files to the ADONIS database, you can save all files, which are required for the library-specific functions (e.g. documentation, page layout, model type symbol) in the ADONIS database. This way these files are available independently of the access possibilities and access paths in your file system.

Select in the menu "Extras" the menu item "File management" to show the window "*<Data base file name>* - Database File Management" (see fig. 398) .
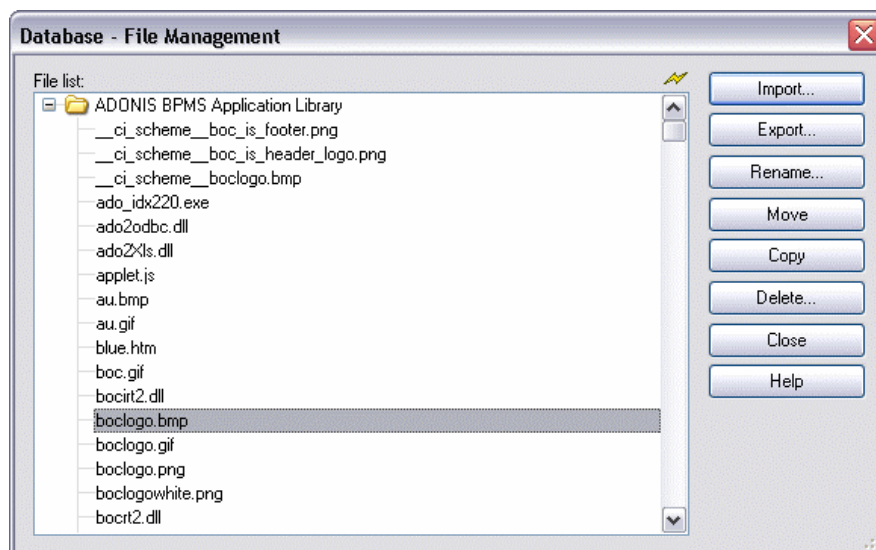
Figure 398: File management

In the **file list**, the files stored in the database will be listed according to their utilisation in the specific application libraries (i.e. in the appropriate order).

> **Hint:** The files may not be assigned to any application libraries, if for instance an application library from ADONIS 3.81 is used. These files will be shown in the file list on the same level as the folders of the application library.

Clicking on the button:

"Import"   enables to save (see chap. 15.1, p. 607) a file to the database,

"Export"   enables to export (see chap. 15.2, p. 607) a file saved from the database,

"Rename" enables to assign a new name (see chap. 15.3, p. 607) to a file stored in the database,

"Move"   enables to move a saved file to another application library (see chap. 15.4, p. 608),

"Copy"   enables to copy a saved file to another application library (see chap. 15.5, p. 608),

"Delete"   enables to delete a saved file from the database.

> **Hint:** When you delete or rename files, the regular utilisation of the library-specific functions is no longer guaranteed.

The **integration** of external files saved in the database is done using the following syntax:

```
db:\<file name>
```

If a file stored in the database is called by the library specific functions, the search is done firstly in the folder of the appropriate application library and then in the other not assigned files.

> **Hint:** In most of the cases, when integrating external files, the file name (incl. path) has to be set between quotation marks. The back slashes "\" in the path specification of the file must in this case be masked by '\' (e.g. "db:\\logo.gif" for the file "logo.gif" stored in the database).

15.1   Import Files

By clicking on the window "*<Data base name>* - file management" (see fig. 398) and then on the button "Import" you can import (save) a file into ADONIS.

In the window "Open" select the appropriate file and click on the button "Open". As a result, the imported file will be shown on the list.

15.2   Export Files

In order to export a file from the ADONIS database, select the file to be exported in the window "*<Data base name>* - file administration" (see fig. 398) and then click on the button "Export".

In the window "Save as" select the path and the name for the file to be exported and then click on the button "Save". As a result, the exported file will be saved to the selected place in the file system.

15.3   Rename Files

Rename a file from the ADONIS database, by selecting the file to be changed in the window "*<Data base name>* - file management" (see fig. 398) and then clicking on the button "Rename".

The current file name will be displayed in the window "Rename file" (see fig. 399).
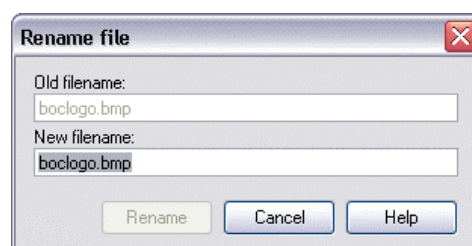
Figure 399: Rename file

Enter in the fields "New file name" the new name of the file and then click on the button "Rename". The new file name will be shown in the file list.

15.4　Move Files

In order to move a file to a folder belonging to another application library, open the window "*<Data base name>* - file management" (see fig. 398), select the file to be moved and click on the button

"Move". As soon as the mouse pointer is moved onto the detailed list, it is changed into   .

After that, click on the target folder.

15.5　Copy Files

In order to copy a file into a folder belonging to a different application library, open the window "*<Data base name>* - file management" (see fig. 398), select the file to be copied, and then click on the button

"Copy". As soon as the mouse pointer is moved onto the detailed list, it is changed into   .

After that, click on the target folder.

16. AdoScript

"AdoScript" script language allows to add additional functionality to the standard version of ADONIS. Apart from that, AdoScripts allow to call any external programs and functions in DLLs.

The AdoScript call is possible via the library attribute "External tool coupling" (see chap. 2.1.3.11, p. 197) and can be defined either as a library-specific menu point or via special "turn of events". All the acceptable "events" are described in the chapter "Event handler" (see chap. 16.19, p. 664).

Additionally it is possible to start AdoScripts in the ADONIS Notebook, via the attribute of type "Programcall" (see chap. 5.11, p. 539).

The syntax of AdoScript is based on the following grammar:

***StatementSeq* :**	`{ Statement } .`

***Statement* :**	`Execute │ Send │ CC │ System │ Start │ Call │`
	`Set │ SetL │ SetG │ Leo │ IfStatement │ WhileStatement │`
	`ForStatement │ BreakStatement │ ExitStatement │`
	`FunctionDefinition │ ProcedureDefinition │ ProcedureCall .`

***Execute* :**	`ExecuteFile │ ExecuteEx .`
***ExecuteFile* :**	`EXECUTE file: scriptText [ scope:ScopeSpec ] .`
***ExecuteEx* :**	`EXECUTE scriptText [ scope:ScopeSpec ] .`
***ScopeSpec* :**	`separate │ same │ child .`
***Send* :**	`SEND msgText to:msgPortName [ answer:varName ] .`
***CC* :**	`CC msgPortName [ debug ] [ raw ] anyLeoElement .`
***System* :**	`SYSTEM strExpr [ hide ] [ result:varName ] .`
***Start* :**	`START strExpr [ cmdshow:CmdShow ] .`
***CmdShow* :**	`showmaximized │ showminimized │` `showminnoactive │ shownormal .`
***Call* :**	`CALL dll:strExpr function:strExpr { InputParam }` `        [ result:varName ] [ freemem:strValue ] .`
***InputParam* :**	`varName:anyExpr .`
***Set* :**	`SET { VarAssignment } .`
***SetL* :**	`SETL { VarAssignment } .`
***SetG* :**	`SETG { VarAssignment } .`
***VarAssignment* :**	`varName:anyExpr .`
***Leo* :**	`LEO { LeoCmd } .`

***LeoCmd* :**	`parse:stringExpr │ get-elem-count:varName │`
	`set-cur-elem-index:intExpr │ get-keyword:varName │`
	`is-contained:varName [ :strExpr ] │`
	`get-str-value:varName [ :strExpr ] │`
	`get-int-value:varName [ :strExpr ] │`
	`get-real-value:varName [ :strExpr ] │`

	get-tmm-value:*varName* [**:***strExpr*] | **get-time-value:***varName* [**:***strExpr*] | **get-modifier:***varName* **:***strExpr* .
***IfStatement* :**	**IF** *booleanExpr* **{** *StatementSequence* **}** { **ELSIF** *booleanExpr* **{** *StatementSequence* **}** } [**ELSE** **{** *StatementSequence* **}**] .
***WhileStatement* :**	**WHILE** *booleanExpr* **{** *StatementSequence* **}** .
***ForStatement* :**	*ForNumStatement* | *ForTokenStatement* .
***ForNumStatement* :**	**FOR** *varName* **from:***numExpr* **to:***numExpr* [**by:***numExpr*] **{** *StatementSequence* **}** .
***ForTokenStatement* :**	**FOR** *varName* **in:***strExpr* [**sep:***strExpr*] **{** *StatementSequence* **}** .
***BreakStatement* :**	**BREAK** .
***NextStatement* :**	**NEXT** .
***ExitStatement* :**	**EXIT** .
***FunctionDefinition* :**	**FUNCTION** *functionName* [**:global**] { *FormalFuncParameter* } **return:***expression* .
***FormalFuncParameter* :**	[**reference**] *paramName* **:***TypeName* .
***ProcedureDefinition* :**	**PROCEDURE** [**global**] *ProcedureName* [*MainParameter*] { *FormalProcParameter* } **{** *StatementSequence* **}** .
***MainParameter* :**	*TypeName* **:***paramName* .
***FormalProcParameter* :**	[**reference**] *paramName***:***TypeNameOrReference* .
***TypeNameOrReference* :**	*TypeName* | **reference** .
***TypeName* :**	**string** | **integer** | **real** | **measure** | **time** | **array** | **expression** | **undefined** .
***ProcedureCall* :**	*anyLeoElement* .
***ProcedureName* :**	*keyword* .

16.1   EXECUTE

Execute ***:***   **EXECUTE** *Source* [**scope:***ScopeSpec*] [**result:***varName*] .

Source* :**   **file:filename* \| *scriptText* .

***ScopeSpec* :** **separate** \| **same** \| **child** \| **default** .

filename and *scriptText* are strings.

EXECUTE executes a new script dynamically which can be given in a run-time string variable.

Example:

The code

```
CC "AdoScript" EDITBOX title:"AdoScript"
#-> text
IF (endbutton = "cancel") {
```

```
    EXIT
}
EXECUTE (text)
```

lets the user enter any text, dynamically, which is interpreted as a new script and then executed.

Example:

The code

```
EXECUTE file:"d:\\test.asc"
```

gives the same effect as

```
CC "AdoScript" FREAD file:"d:\\test.asc"
EXECUTE (text)
```

However in the first situation, in case of error messages, the system will also display the file name and in the second case it will not.

Scopes:

Scopes are areas, in which determined variables are visible (accessible). As a result, for example, a variable created locally within a subprogram (procedure or instructions executed through **EXECUTE**) is not visible outside it. However, it is also possible to gain access to the variables of the subprogram which were created outside it. It is also possible that the scopes are nested, but the scope variables in the outstanding scopes are visible. The scopes are created at a procedure or **EXECUTE** call.

EXECUTE has three possibilities for the scoping behaviour.

scope:separate does the execution in a new scope, in which no access to another scope apart from global variables is possible.

scope:child creates a scope which is a subscope of the current scope. Therefore the visible variables of the **EXECUTE** call are also visible within the **EXECUTE** execution.

scope:same does the execution of the socpe within the **EXECUTE** call, as if the instructions of the **EXECUTE** execution are in the place as the **EXECUTE** call.

In the case of **scope:default** (or no **scope**-specification), the variable **xscope** will be evaluated: If the **xscope** has the value **"child"** (or **"same"**) , it means that **scope:child** (or **scope:same**) will be taken into account. In all other cases, the behaviour of **scope:separate** will be used.

16.2   SEND / CC

__Send :__   **SEND** *msgText* **to:***msgPortName* [**answer:***varName*] .

__CC :__   **CC** *msgPortName* [**debug**] [**raw**] *anyLeoElement* .

MessagePorts (see chap. 16.18, p. 621) are internal instances in ADONIS which are part of a component that receives strings (messages) which contain component-specific commands. A message is a LEO text.

When a message is received, the MessagePort decodes it and executes the desired command. Some MessagePort commands return a value which can be assigned to a run-time variable using **answer** in the **SEND** command. The return value is also generally a LEO text, which makes it possible for structured values to be returned.

The MessagePort (specified with **to** in the **SEND** command) can be one of the following:

"AdoScript"	some useful dialogues;
"Core"	interface to the core component
"CoreUI"	interface to the CORE UI modules dialogues (e.g. the ModelSelectBox)
"Application"	interface to the application and its main window
"Acquisition"	interface to the Acquisition Component
"Modelling"	interface to the Modelling Component
"Analysis"	interface to the Analysis Component
"Simulation"	interface to the Simulation Component
"Evaluation"	interface to the Evaluation Component
"ImportExport"	interface to the Import/Export Component
"Documentation"	interface to the Documentation Component
"UserMgt"	interface to the User Management

CC ("component call") provides an easier method to send messages to components and get the needed information out of the answers. The **CC** element behaves like a macro.

For example,

```
CC "AdoScript" INFOBOX "Primroses & elephants"
```

would mean:

```
SEND "INFOBOX \"Primroses & elephants\"" to:"AdoScript"
```

.

The LEO element after the **CC** element is called a *message element* and does not really belong to the AdoScript syntax, i.e. it would cause a syntax error without the leading **CC** element. The benefit of the **CC** form above is that the string does not have to be masked. If there are more parameters, maybe some of them as expressions, it is easier to write the "natural" LEO element than to construct a string which will be a LEO element. Expressions in the message element are evaluated before the message is sent. An example for that is given below.

The second feature is that **CC** assigns attributes encoded in an answer as a **RESULT** LEO element to run-time variables. If the answer of a message is, for example

```
RESULT ecode:0 val:"Berlin"
```

CC behaves like executing

```
SET ecode:0 val:"Berlin"
```

Example:

The fragment:

```
SEND ("GET_ATTR_VAL objid:" + STR objid + " attrid:" + STR attrid)
to:"Core" answer:text
LEO parse:(text) get-int-value:ecode:"ecode" get-str-value:val:"val"
IF (ecode > 0) ...
```

could be replaced by:

```
CC "Core" GET_ATTR_VAL objid:(objid) attrid:(attrid)
 IF (ecode > 0) ...
```

"Raw Text"

If **raw** in a **CC** command is quoted, the message will not be executed and sent to the component, i.e. without evaluating expressions and replacing them through result values. This is needed if the message which should be sent contains an AdoScript-Code.

"Debugging"

If **debug** is quoted in a **CC** command, the **RESULT** string will be displayed in an InfoBox. This can be useful to find several errors. If the variable **ccdebug** is set on **"always"** an Infobox will always be displayed at **CC** even if **debug** is not quoted. If the variable **ccdebug** is set as**"never"**, a Debug-Infobox will be displayed.

Example:

```
CC "Core" debug GET_ATTR_VAL objid:(objid) attrid:(attrid)
SET ccdebugfile:"d:\\ccdebug.log"
CC "AdoScript" debug FILE_COPY from:"ei1.txt" to:"ei2.txt"
SET ccdebugfile:""
```

16.3   SYSTEM

System :   **SYSTEM** *strExpr*
                [**hide**] [**result:***varName*] .

SYSTEM executes a command in an operating system environment. The call is synchronous i.e. the AdoScript process waits until the execution of the command is finished. The result of the executed command can be assigned through **result** to a variable. Through quoting **hide** the appearance of a command window can be prevented.

Example 1: Let the user edit a file

```
SYSTEM ("notepad " + filename)
```

If functions or commands of the command line shall be used, make sure to not use*Command* but instead**cmd /c** *Command*.

Example 2: Delete a file

```
SYSTEM (cmd /c del " + filename) result:rc
IF (rc != 0) ...
```

16.4   START

Start :   **START** *strExpr* [**cmdshow:***CmdShow*] .

CmdShow **:**   `showmaximized` | `showminimized` |
                   `showminnoactive` | `shownormal` .

START starts an application in the operating system environment. The call is synchronous i.e. the AdoScript process will be continued directly and the called application is executed in paralell. The best **cmdshow** parameter specificates the start condition of the application main window.

<u>**Example:**</u>

To start **notepad.exe** in the maximised window: (as maximised window)

```
START ("notepad " + filename) cmdshow:showmaximized
```

16.5   CALL

<u>*CallStatement :*</u> **CALL dll:***strExpr* **function:***strExpr*
                   { *InputParam* }
                   [**result:***varName*] [**freemem:***strValue*] .

InputParam **: varName:***anyExpr* .

CALL calls a function in a dynamic link library (DLL). The name of the DLL is specified with **dll**. The function is specified with **function** using the C function declaration syntax. A combination of a return value type and (zero, one ore more) parameter value types is called a *signature*.

Supported return value types of a called DLL function are: **void**, **long**, **double**, **char***. Supported parameter value types are: **long**, **double**, **char***, **long***, **double***, **char****, where the first three are input and the last three are output parameters. All possible signatures for these types are supported.

If the return value type is **char*** or a parameter value type is **char**** the DLL is allocating memory to hold the string value. The freeing of this memory has to be done within the DLL too, but the "freemem" call is done by AdoScript. By default AdoScript calls a function **freemem (char*)** of the DLL if the memory is no longer needed (i.e. the string has been copied to an AdoScript run-time variable). If another function is specified with **freemem** in the **CALL** statement that function is called instead.

All input paramerers are given as attributes within the **CALL** statement using the same names as the formal parameters in the **function** declaration. (So the names of the formal parameters must consist of lowercase letters only.) The output parameters are assigned to run-time variables, also with the same names as the formal parameters in the **function** declaration. The return value of the called function is assigned to the run-time variable specified with the atribute **result** or to the run-time variable **result** if the specification is omitted.

<u>**Example: Calling a DLL with AdoScript**</u>

```
  # Functions returning void with input params only!
  CALL dll:"dll1.dll" function:"void f_void_void ()"
  CALL dll:"dll1.dll" function:"void f_void_long (long n)" n:4242
  CALL dll:"dll1.dll" function:"void f_void_double (double d)" d:3.14
  CALL dll:"dll1.dll" function:"void f_void_charptr (char *p)" p:"Hallo Welt"
  CALL dll:"dll1.dll" function:"void f_void_long_double_charptr (long n, double d,
char* p)" n:4711 d:2.34 p:"Yippie"
  # Functions returning long with input params only!
  CALL dll:"dll1.dll" function:"long f_long_void ()" result:bla
CC "AdoScript" INFOBOX (STR bla)
  CALL dll:"dll1.dll" function:"long f_long_long (long n)" result:bla n:4242
```

```
  CC "AdoScript" INFOBOX (STR bla)
  CALL dll:"dll1.dll" function:"long f_long_double (double d)" result:bla d:3.14
  CC "AdoScript" INFOBOX (STR bla)
  CALL dll:"dll1.dll" function:"long f_long_charptr (char *p)" result:bla p:"Hallo
Welt"
  CC "AdoScript" INFOBOX (STR bla)
  CALL dll:"dll1.dll" function:"long f_long_long_double_charptr (long n, double d,
char* p)" result:bla n:4711 d:2.34 p:"Yippie"
  CC "AdoScript" INFOBOX (STR bla)
  # Functions returning double with input params only!
  CALL dll:"dll1.dll" function:"double f_double_void ()" result:bla
  CC "AdoScript" INFOBOX (STR bla)
  CALL dll:"dll1.dll" function:"double f_double_long (long n)" result:bla n:4242
  CC "AdoScript" INFOBOX (STR bla)
  CALL  dll:"dll1.dll"  function:"double  f_double_double  (double  d)"  result:bla
d:3.14
  CC "AdoScript" INFOBOX (STR bla)
  CALL  dll:"dll1.dll"  function:"double  f_double_charptr  (char  *p)"  result:bla
p:"Hello World"
  CC "AdoScript" INFOBOX (STR bla)
  CALL dll:"dll1.dll" function:"double f_double_long_double_charptr (long n, double
d, char* p)" result:bla n:4711 d:2.34 p:"Yippie"
  CC "AdoScript" INFOBOX (STR bla)
  # Functions with output params only!
  CALL dll:"dll1.dll" function:"void f_get_long (long* bla)"
  CC "AdoScript" INFOBOX (STR bla)
  CALL dll:"dll1.dll" function:"void f_get_double (double* bla)"
  CC "AdoScript" INFOBOX (STR bla)
  CALL dll:"dll1.dll" function:"void f_get_charptr (char** bla)"
  CC "AdoScript" INFOBOX (bla)
  # Mixture
  CALL dll:"dll1.dll" function:"void f_all (long nin, double din, char* pin, long*
nout, double* dout, char** pout)" nin:2 din:3.14 pin:"It works!"
  CC "AdoScript" INFOBOX ("nout: " + STR nout + "\ndout: " + STR dout + "\npout: "
+ pout)
```

16.6   SET / SETL / SETG

SET { *VarAssignment* } .

SETL { *VarAssignment* } .

SETG { *VarAssignment* } .

VarAssignment : *LValue* **:** *anyExpr* .

LValue :   *varName* | *ArrayLValue* .

ArrayLValue :   *varName* **[** *CommaExpr* **]** .

SET assigns a new or an existing AdoScript-Variable to a new value. As a standard the life time of a variable is defined in an AdoScript in which a value will be assigned. Through using **SETG** the variable remains during the whole ADONIS meeting and can therefore be used for later executions of the same or of other AdoScripts. If a variable exists in the current scope the value **SET** will be over written. If a local variable should be created, even if a variable with the same name exists in a higher scope, **SETL** has to be used.

Example:

Count the executions of an AdoScript

```
IF (type (numcalls) = "undefined") {
  SETG numcalls:1
}
ELSE {
  SET numcalls:(numcalls + 1)
}
CC "AdoScript" INFOBOX ("Anzahl der Ausführungen: " + STR numcalls)
```

16.7   LEO

LEO *ParseCmd* { *AccessCmd* } .

ParseCmd :   **parse:***stringExpr* .

AccessCmd :   **get-elem-count:***varName* |
                set-cur-elem-index:*intExpr* |
                get-keyword:*varName* |
                is-contained:*varName* [:*strExpr*] |
                get-str-value:*varName* [:*strExpr*] |
                get-int-value:*varName* [:*strExpr*] |
                get-real-value:*varName* [:*strExpr*] |
                get-tmm-value:*varName* [:*strExpr*] |
                get-time-value:*varName* [:*strExpr*] |
                get-modifier:*varName***:***strExpr* .

With **LEO** a LEOgram can be parsed which is shown as a string. This will be transmitted with **parse**. This can be useful e.g., if an external call delivers structured data in the according format. With **get-elem-count** the number of elements can be written into a variable . With **set-cur-elem-index** a current element can be set, of which the following reading instructions refers:

get-keyword   Detect key word.

is-contained   Detect existance of an attribute .

get-*T*-value   Detect attribute value of type *T* detect. If no attribute name is detected, the detection refers to a value of the element. For T one of the following types can be inserted: string, integer, real, measure, time.

get-modifier   detect modifier of an attribute.

16.8   IF ... ELSIF ... ELSE

IF *booleanExpr* { *StatementSequence* }
     { **ELSIF** *booleanExpr* { *StatementSequence* } }
     [**ELSE** { *StatementSequence* }] .

With **IF** a determined branch can be programmed. When executing the IF condition (*booleanExpr*) will be executed first. If the result is not **0**, the **IF** instruction part will be executed. Otherwise, the **ELSIF** conditions will be evaluated one after one and if the result is not **0**, the according instruction section will be executed. If none of the **IF**/**ELSIF** conditions give a result of not **0**, the **ELSE** construction part will be executed (if one exists). After one of the construction parts was executed, no further conditions of the **IF** condition will be evaluated, but the AdoScript will continue after the **IF** condition.

16.9   WHILE

```
WHILE booleanExpr { StatementSequence } .
```

With **WHILE** a determined loop can be programmed. When executing the condition (*booleanExpr*) will be executed first. If the result is not **0**, the **WHILE**-condition part will be executed. After that the condition will be evaluated again etc. If the result of the condition evaluation has the value **0**, the execution of the AdoScript with the next element (the element after the bracketes of the **WHILE** element) will be continued.

16.10   FOR - Numeric Form

```
FOR varName from:numExpr to:numExpr [ by:numExpr ]
{ StatementSequence } .
```

FOR repeats an instruction for every value which follows a <u>local</u> variable (*varName*) of the start value **from**, to the aim value **to**. The repetition schedule (steps) is defined in the **by** part (**1** by default). The running variable can be used in the instruction part of the expressions.

The loop will be executed in steps bigger than 0 as long as the running variable is not smaller than the aim value; at steps smaller than 0 the loop will be executed until the running variable is under the aim value.

16.11   FOR - Stringtoken Form

```
FOR varName in:strExpr [ sep:strExpr ]
    { StatementSequence } .
```

This version of the FOR loop repeats an instruction for every variable of the type string, which is quoted with **in**. Optionally, the user can define a separator with the **sep** command. If there is no special definition entered, it is interpreted as (**" "**).

Example:

After execution of the codes

```
SET result:0
FOR ei in:"12 23 34" {
  SET result:(result + VAL ei)
}
```

the **result** gets the value **69** (=12+23+34).

16.12   BREAK

```
BREAK .
```

The **BREAK** command provides an exit from the statements including **WHILE** or **FOR** command.

16.13 NEXT

```
NEXT .
```

The **NEXT** command provides an exit from the statements including **WHILE** or **FOR** command and to start the next iteration of the loop.

16.14 EXIT

```
EXIT [ intValue ] .
```

The **EXIT** command ends the running of AdoScript. In the case of hierarchical AdoScripts only the lowest running level will be finished. If the AdoScript belongs to a string and was called with the **EXECUTE** command, the **EXIT** command will be followed by the statement after the **EXECUTE**.

If the *intValue* is specified, the Exit-Code (default: 0) will be given as the response to the AdoScript call. If the call is the **EXECUTE** command, a variable can be created.

16.15 FUNCTION

```
FUNCTION functionName [ :global ]
        { FormalFuncParameter } return:expression .
```

FormalFuncParameter: [**reference**] *paramName* :*TypeName* .

TypeName: **string** | **integer** | **real** | **measure** |
                    time | **array** | **expression** | **undefined** .

The **FUNCTION** command allows the definition of new functions from the LEO expressions (see chap. 17.3, p. 674). These functions can be used in other expressions.

Example: Please define and call a factorial function:

```
FUNCTION fak n:integer
    return:(cond (n <= 1, 1, n * fak (n – 1)))
SET m:(fak (10))
```

16.16 PROCEDURE

```
PROCEDURE [ global ] ProcedureName [ MainParameter ]
        { FormalProcParameter } { StatementSequence } .
```

MainParameter: *TypeName*:*paramName* .

FormalProcParameter: *paramName*:*TypeNameOrReference* .

TypeNameOrReference: *TypeName* | **reference** .

TypeName: **string** | **integer** | **real** | **measure** | **time** | **expression** | **un-**
                    defined .

ProcedureName: *keyword* .

With the **PROCEDURE** parametrical subprograms - procedures can be defined. Therefore longer scripts can be structured in clearly arranged parts and the repetition of code fragments can be avoided. This shortens the execution time as less code has to be parsed.

With a procedure a new AdoScript command will be defined. The name of the procedure is therefore to be written as a keyword (in capital letters).

In case of parameters you have to distinguish between main parameters and attribute parameters and between value parameters and reference parameters.

The main parameter is a nameless parameter which will be quoted directly after the key word. When calling the procedure, the attribute parameter has to be quoted together with its name (such names always start with a lower-case letter). The order of the attribute parameters is not important.

In case of the value parameters, their values will be handed over to the subprogram. Additionally, they can be implemented in form of expressions, which will calculate the values each time they are called. In case of the referenced parameters, the procedure modifies the transfered value. Apart form that, the reference parameters can also be used for getting back the procedure values. Each reference parameter has to be defined as an attribute parameter.

Example: Types of parameters during the procedure call

```
MYPROC 123 val:"test" result:a
```

It calls the procedure named **MYPROC** where the main parameter gets the value **123** and the attribute parameter **val** gets the value **"test"**. The call of the procedure causes saving the result value in the variables **a**, seen here as the reference-result-parameters **result**.

The procedure definition is executed through the key word **PROCEDURE**, followed by the procedure name. Subsequenty, there are formal paramters and - included in brackets - the procedure body. The formal parameters determine which parameters have to be quoted during a process call. The procedure body contains the executable code of the procedure (AdoScript), which can easily be accessed.

In order to define the main parameter it is necessary to enter in the procedure definition the procedure name, the type of the parameter, followed by a colon and a name for the parameter. This name will be needed to get access to the main parameter within the procedure body, but it is of no importance.

The attribute parameters are defined by the name of the parameter followed by a colon and the type of the parameter. The name and the overtaken value or referenced variable have to be quoted when calling the procedure. If it is a reference-result-parameter, the reference should be given as a type name. For the value parameter the following types are available:**string**, **integer**, **real**, **time** and **measure**.

Example:

```
PROCEDURE MYPROC integer:n val:string result:reference
{
    SET result:(val + STR n)
}
```

The procedure **MYPROC** is defined with an **integer**-main parameter, a **string** parameter **val**, and a reference-result-parameter **result**. The main parameter is accessible within the procedure body over the variable **n** accessed. When leaving the procedure body, the result value will be assigned to the quoted call variable.

16.17 AdoScript Examples

● Reverse Text (see chap. 16.17.1, p. 620)

● Quicksort (see chap. 16.17.2, p. 620)

16.17.1   Reverse Text

The code

```
# ITEM "Reverse text..." modelling
CC "AdoScript" EDITBOX text:"Miss Moneypenny"
IF (endbutton = "cancel")
  EXIT
ENDIF
SET rev:""
WHILE (LEN text)
  SET rev:(rev + text SUB (LEN text - 1))
  SET text:(copy (text, 0, LEN text - 1))
ENDWHILE
CC "AdoScript" INFOBOX ("*** reverse ***\n" + rev)
```

lets the user enter any text, calculates the reverse text and displays it in an InfoBox.

In order to calculate the reverse text there the following function can be defined:

```
FUNCTION rev s:string
    return:(cond (LEN s, rev (copy (s, 1, -1)) + s SUB 0, ""))
CC "AdoScript" INFOBOX (rev ("Napoleon"))
```


16.17.2   Quicksort

The procedure **QUICKSORT** sorts the signs contained in a string alphabetically. **repltok** is a support function, which replaces in a string a line with another one via a determined index. The **PARTITION** is a help procedure, which divides the lines (elements) contained in a string into two halves and a pivot element. However, in one half only the smaller, and in the other half only the bigger elements are the pivot elements.

```
PROCEDURE QUICKSORT list:reference start:integer end:integer
{
  IF (start < end) {
    PARTITION list:list start:(start) end:(end) result:split
    QUICKSORT list:list start:(start) end:(split - 1)
    QUICKSORT list:list start:(split + 1) end:(end)
  }
}
PROCEDURE PARTITION list:reference start:integer end:integer
                    result:reference
{
  SETL pivot:(token (list, end, "\n"))
  SETL bottom:(start - 1)
  SETL top:(end)
  SETL done:0
  WHILE (NOT done) {
      WHILE (NOT done) {
        SET bottom:(bottom + 1)
        IF (bottom = top) {
          SET done:1
          BREAK
        }
        IF (token (list, bottom, "\n") > pivot) {
          SET list:(repltok (list, top,
```

```
                                           token (list, bottom, "\n")))
              BREAK
          }
        }
        WHILE (NOT done) {
          SET top:(top - 1)
          IF (top = bottom) {
            SET done:1
            BREAK
          }
          IF (token (list, top, "\n") < pivot) {
            SET list:(repltok (list, bottom,
                               token (list, top, "\n")))
            BREAK
          }
        }
      }
    SET list:(repltok (list, top, pivot))
    SET result:(top)
  }
  FUNCTION repltok list:string index:integer newtok:string
      return:(set (r, ""), set (i, 0),
              fortok (t, list, "\n",
                      (set (r, r + cond (LEN r, "\n", "") +
                            cond (i = index, newtok, t)),
                       set (i, i + 1))),
              r)
```

It is possible to call **QUICKSORT** for example this way:

```
    SET text:("Caesar,Nero,Titus," +
              "Trajan,Konstantin,Augustus")
    SET text:(replall (text, ",", "\n"))
23    QUICKSORT list:text start:0 end:(tokcnt (text, "\n"))
    CC "AdoScript" INFOBOX (text)
```

16.18 MessagePorts

MessagePorts are internal instances in ADONIS, which are parts of any component and are used to receive messages. Actually a message is a string, the LEOgramm that contains commands typical for each component.

When a message is received, the MessagePort decodes it and executes the desired command. Some commands return a value. Mostly the return value is also a LEO text, which makes it possible that structured values are returned.

Sending messages can be achieved in AdoScript (see chap. 16., p. 609) using **SEND** or **CC**. **CC** is generally preferred. One advantage of this command is the handling of a returned LEO text, i.e. assigning encoded values to run-time variables, which is done by **CC** completely. Attention: Use **CC** only where a **"RESULT"** string or an empty string is returned.

Existing MessagePorts in ADONIS are:

- **"AdoScript"** (see chap. 16.18.1, p. 622) - some useful dialogues and other general support

- **"Core"** (see chap. 16.18.2, p. 628) - interface to the core

- **"CoreUI"** (see chap. 16.18.3, p. 640) - dialogues to model and attribute profile selection, tabular view

- **"Application"** (see chap. 16.18.4, p. 641) - interface to the application and its main window

- **"Modelling"** (see chap. 16.18.5, p. 644) - interface to the Modelling Component

- **"Analysis"** (see chap. 16.18.6, p. 651) - interface to the Analysis Component

- **"Simulation"** (see chap. 16.18.7, p. 651) - interface to the Simulation Component

- **"Evaluation"** (see chap. 16.18.8, p. 653) - interface to the Evaluation Component

- **"ImportExport"** (see chap. 16.18.9, p. 655) - interface to the Import/Export Component

- **"Documentation"** (see chap. 16.18.10, p. 659) - interface to the Documentation Component

- **"AQL"** (see chap. 16.18.11, p. 662) - executing AQL queries

- **"UserMgt"** (see chap. 16.18.12, p. 662) - interfaces to the User Management (Administration Toolkit)

16.18.1   Commands of the "AdoScript" MessagePort

General:

```
SET_MP_TYPE_CHECKING ( on | off ) .
```

This executes the type examination of the parameters in the MessagePorts. It is turned off in defult. However, it is recommended to turn it on at the "AppInitialized" event, as this makes the search of the errors in AdoScripts much more simple.

```
SLEEP ms:intValue .
```

Stops ADONIS for *intValue* milliseconds. During this time, the mouse pointer shows an hourglass.

Browser:

```
BROWSER [ title:strValue ] [ content:strValue ]
        [ fieldsep:strValue ] [ with-handlecolumn ]
        [ max-size ] [ alignment:strValue ]
        [ header:strValue ] [ print-header:strValue ] .
    --> RESULT ecode:intValue .
```

Creates a table (ADONIS browser) with the given content (**content**), in a window with the given title (**title**). **fieldsep** defines which signs change cells in a table. If the optional parameter **with-handlecolumn** is included, the first column will appear in grey.

The parameter **max-size** means that the browser window will be displayed at its maximum size. **alignment** can influence the assignment of the data columns. The string of characters must contain the letters "L" for flush left, "C" for centered and "R" for flush right and include at least one letter per column. The first column will not be considered, if this one is already grey. If the optional parameter **header** and/or **print-header** are defined, the head data for the saving or printing will be put on the same level.

Example:

```
  CC "AdoScript" BROWSER title:"My browser window"
```

```
  content:";Column 1;Column 2\nRow 1;Value 11;Value 12\nRow 2;Value 21;Value
22\nRow 3;Value 31;Value 32"
  with-handlecolumn alignment:"LR"
  header:"Save\nthis header" print-header:"Print\nthis header"
```

EDIT_BROWSER [**title:***strValue*] [**content:***strValue*]
       [**fieldsep:***strValue*] [**header:***strValue*]
       [**print-header:***strValue*] [**no-special-buttons**] [**with-handlecolumn**]
       [**max-size**] [**alignment:***strValue*]
    --> **RESULT ecode:***intValue* **text:***strValue*.

Creates an editable table (ADONIS browser) with the given content (**content**) in a window with the defined title (**title**). If the optional parameter **with-handlecolumn** is included, the first column will appear in grey. The parameter **alignment** is allowed to influence the assignment of the data columns. The chain of characters must both contain the letter "L" for flush left, "C" for centered and "R" for flush right and include at least one letter per column. The first column will not be considered if it is already grey. If the optional parameter **header** and/or **print-header** are defined, the head data for the saving or printing will be put on the same level. The parameter **max-size** means that the browser window will be displayed at its maximum size. The parameter **no-special-buttons** means that the buttons 'Save', 'Print' and 'Search' will be faded out. The edited content will be displayed in the result parameter **text**.

Files:

DB_FILE_LIST .
    --> **RESULT ecode:***boolValue* **files:***strValue* .

Returns in **files** a list of all files in the database which are connected to the current application library. The separator between the file names is "*".

DIRECTORY_DIALOG [**path:***strValue*] .
    --> **RESULT endbutton:***EndButton* **path:***strValue*

EndButton : **"ok"** | **"cancel"** .

Opens the *file* selection dialogue of the operation system. The Parameter **path** specifies the start directory. The result value **path** contains the complete path defined by the user, if the dialogue was closed with the **endbutton** "OK".

DIR_CREATE path:*strValue* .
    --> **RESULT ecode:***intValue* .

The **ecode** has a value unequal to **0**, if there is an error.

DIR_LIST path:*strValue* [**filemask:***strValue*] .
    --> **RESULT ecode:***intValue* **files:***strValue* **dirs:***strValue* .

Returns a list with all files and directories in the directory **path**. **ecode** has a value unequal **0**, if an error occurs. The file names are separated by "*".

DIR_REMOVE path:*strValue* .
    --> **RESULT ecode:***intValue* .

The **ecode** has a value unequal to **0**, if there is an error.

FILE_COPY from:*filename* **to:***filename* .
    --> **RESULT ecode:***intValue* .

The **ecode** has a value unequal to **0**, if there is an error.

```
FILE_DELETE file:filename .
    --> RESULT ecode:intValue .
```

The **ecode** has a value unequal to **0**, if there is an error.

```
FILE_DIALOG ( open | saveas ) [ path:strValue ]
        [ filter1:strValue type1:strValue ] [ default-ext:strValue ] .
    --> RESULT endbutton:strValue path:strValue .
```

Opens the *file* selection dialogue of the operating system. **open** and **saveas** specify, whether the dialogue will be an "Open" or a "Save" dialogue. The Parameter **path** specifies the start directory. The result value **path** contains the complete path specified by the user, if the dialogue was closed using the **endbutton** "Open" (resp. "Save").

```
FILE_EXISTS file:strValue .
    --> RESULT exists:boolValue .
```

Tests, if the file *strValue* specified with **file** exists. **exists** has the value **1** if a file with this name exists.

```
FREAD file:filename [ binary:boolValue ] [ base64:boolValue ] .
    --> RESULT text:strValue ecode:intValue .
```

Reads the complete content of the file *filename*. The optional parameter specifies in binary whether the file should be read in binary mode (binary:1), or in text mode (binary:0). In text mode, all line breaks will be shown as \n, in binary mode they stay unchanged. With base64, the resulting **text** will not contain the original binary code of the file, but its base64 representation.

The variable **text** contains the input of the file. The variable **ecode** has a value unequal to **0** if there is an error. Otherwise the file content is quoted in **text**. The **ecode** has the value **2** if the file is too big (bigger than 2^31 characters, ~2 GB) and therefore is not saveable within the string variable.

```
FWRITE file:filename text:strValue
        append:boolValue binary:boolValue base64:boolValue .
    --> RESULT ecode:intValue written:intValue .
```

Transforms the text entered with the **text** into a new or already existing file. If **append** is not quoted, the file will contain the same text. Otherwise the text will be written at the end of the file. The variable **ecode** has a value not equal to **0** if there is an error.

```
GET_CWD .
    --> RESULT cwd:strValue .
```

The temporary working catalogue will be given back as the absolute path.

```
SET_CWD path:strValue .
    --> RESULT ecode:intValue .
```

Defines with **path** a value *strValue* for the current working directory.

```
GET_TEMP_FILENAME .
    --> RESULT filename:strValue
```

This command asks the operating system about the temporary data name. It is a matter of a complete name of a file from the TEMP catalogue, which has not existed yet. The environment variable TEMP of the operating system must be correct. (Normally this is the case in every system.)

Output Window:

```
CREATE_OUTPUT_WIN winid:strValue title:strValue .
```

```
     --> RESULT ecode:intValue .

OUT winid:strValue text:strValue .
     --> RESULT ecode:intValue .

SET_OUTPUT_WIN_SUBTITLE winid:strValue subtitle:strValue .
     --> RESULT ecode:intValue .
```

CREATE_OUTPUT_WIN creates a dockable window with a text output field. **OUT** appends a line to the text field, **SET_OUTPUT_WIN_SUBTITLE** creates a subtitle to **title**.

Percentage Window:

```
PERCWIN_CREATE [ title:strValue ] .

PERCWIN_SET [ percentage:doubleValue ] [ text:strValue ] .
     --> RESULT ecode:boolValue .

PERCWIN_DESTROY .
```

PERCWIN_CREATE creates a status window with a progress bar, **PERCWIN_SET** defines the interface text and/or the percaentage, **PERCWIN_DESTROY** deletes the status window.

Example:

```
CC "AdoScript" PERCWIN_CREATE title:"Progress:"
SET step:0
FOR i from:0 to:100 by:2
{
  SET step:(step + 1)
  CC "AdoScript" PERCWIN_SET percentage:(i) text:("Step " + STR step)
  CC "AdoScript" SLEEP ms:(200)
}
CC "AdoScript" INFOBOX "Finished!"
CC "AdoScript" PERCWIN_DESTROY
```

Simple UI:

```
EDITBOX text:strValue [ title:strValue ] [ oktext:strValue ]
        [ fontname:strValue ] [ fontheight:intValue ] [ fileeditor ].
     --> RESULT endbutton:strValue text:strValue .
```

This command opens a modal dialogue window, in which the user can edit text.

```
EDITFIELD caption:strValue [ title:strValue ] [ text:strValue ] .
     --> RESULT ecode:boolValue [ text:strValue ] .

INFOBOX strValue [ title:strValue ] .

LISTBOX entries:strValue [ toksep:strValue ]
        [ selection:strValue ] [ title:strValue ]
        [ boxtext:strValue ] [ oktext:strValue ]
        [ w:intValue h:intValue ] [ extra:{ Extra } ] .
     --> RESULT endbutton:strValue selection:strValue [ extraValues ] .
```

Extra: { *CheckBox* } .

CheckBox: **CHECKBOX** *cbText* [**checked:***intValue*] **result-var:***varName* .

cbText: *strValue* .

Opens a modal dialogue window, in which the user can select a *single* value from a list of values. The values will be taken over to a string, separated through the **toksep** character (default: blank (**" "**)). With **selection**, a value can be preselected.

After closing the dialogue, **endbutton** contains the information, with which button the closing was initiated and **selection** the selected value.

```
MLISTBOX entries:strValue [ toksep:strValue ]
         [ selection:strValue ] [ title:strValue ]
         [ boxtext:strValue ] [ oktext:strValue ]
         [ w:intValue h:intValue ] [ extra:{ Extra } ] .
   --> RESULT endbutton:strValue selection:strValue [ extraValues ].
```

Extra: { *CheckBox* } .

CheckBox: **CHECKBOX** *cbText* [**checked:***intValue*] **result-var:***varName* .

cbText: *strValue* .

The same as **LISTBOX**, but with *multiple* selection. The selected values (before and after execution) will be separated through the **toksep** character.

Example:

```
  CC "AdoScript" MLISTBOX
  entries:"First Entry;Second Entry;Third;Fourth;Fifth Entry" toksep:";"
  title:"Example!"  oktext:"Click  me!"  boxtext:"Choose   your   entry:"   selec-
tion:"Second Entry;Third"
  IF (endbutton = "ok")
  {
    CC "AdoScript" INFOBOX (selection)
  }
```

Opens a MLISTBOX with the entries "First Entry", "Second Entry", "Third", "Fourth" and "Fifth Entry". The second and third values are preselected. After the value selection by the user and clicking "OK", the result of the selection is displayed in an INFOBOX.

MSGWIN *strValue* .
MSGWIN hide .

The first form shows a hint window which shows the user that a long task is being executed at the moment (e.g. **MSGWIN "Please wait..."**). The second form closes the hint window. If more hints will be added in directly one after the other, it is enough to call **MSGWIN** hide once at the end.

Example:

```
  FOR i from:1 to:50000
  {
    CC "AdoScript" MSGWIN ("Important message!\nCounting up to 50000: " + STR i)
  }
  CC "AdoScript" MSGWIN hide
```

> **ATTENTION:** If you forget the **msgwin hide** on ending the procedure, ADONIS remains blocked until a restart!

QUERYBOX *strValue* [**title:***strValue*]
    [**ok** | **ok-cancel** | **yes-no** | **yes-no-cancel** | **retry-cancel**]

```
      [ def-ok | def-cancel | def-yes | def-no | def-retry ] .
      --> RESULT endbutton:strValue


ERRORBOX strValue [ title:strValue ]
      [ ok | ok-cancel | yes-no | yes-no-cancel | retry-cancel ]
      [ def-ok | def-cancel | def-yes | def-no | def-retry ] .
      --> RESULT endbutton:strValue


WARNINGBOX strValue [ title:strValue ]
      [ ok | ok-cancel | yes-no | yes-no-cancel | retry-cancel ]
      [ def-ok | def-cancel | def-yes | def-no | def-retry ] .
      --> RESULT endbutton:strValue
```

QUERYBOX, **ERRORBOX** and **WARNINGBOX** each opens a message window with some text, waiting for the user to click a button. The message text is specified with *strValue*, the window title with **title**. Depending on the box type, a different graphical symbol is shown: a question mark in a QUERYBOX, an X in the ERRORBOX and a call sign.

> **Hint:** These symbols come from the operating system and thus can look quite different, according to the version used.

For highlighted end buttons, there are five possible forms which are a combination of "OK", "Cancel", "Yes", "No" and "Repeat". If nothing else is quoted, the "OK" button appears. A included **def** attribute determines the default button. The **endbutton** result quotes the confirmed end button in form of a string in lower-case letters and has one of the following values: **"ok"**, **"cancel"**, **"yes"**, **"no"**, **"retry"**.

```
VIEWBOX text:strValue [ title:strValue ]
      [ fontname:strValue ] [ fontheight:intValue ] [ fileeditor ] .
```

This command opens a modal dialogue window, which shows a (longer) text in a large, write-protected text field.

TreeListBox:


```
TLB_CREATE title:strValue [ oktext:strValue ] [ canceltext:strValue ]
          [ boxtext:strValue ] [ button-w:intValue ]
          [ no-cancel:BoolValue ] [ no-help:BoolValue ]
          [ x:intValue ] [ y:intValue ] [ w:intValue ] [ h:intValue ]
          [ max-w:intValue ] [ max-h:intValue]
          [ min-w:intValue] [ min-h:intValue ]
          [ searchable:BoolValue] [ sorted:BoolValue ] [ flat:BoolValue ]
          [  columns:intValue ] [ multi-sel:BoolValue ]
          [ no-child-sel:BoolValue ] [ no-parent-sel:BoolValue ] .
          [ checklistbox:BoolValue ] [ setdblclick:BoolValue ] .
      --> RESULT ecode:intValue .


TLB_ADD_BUTTON text:strValue name:strValue
          [ index:intValue ] [ disable_if_no_selection:BoolValue ] .
      --> RESULT ecode:intValue .


TLB_EXPAND id:intValue
          [ parentid:intValue ] [ expand:BoolValue ] .
      --> RESULT ecode:intValue .


TLB_EXPAND_ALL .
      --> RESULT ecode:intValue .


TLB_EXPAND_TO id:intValue .
```

```
    --> RESULT ecode:intValue .

TLB_INSERT id:intValue text:strValue
        [ parentid:intValue ] [ is-parent:BoolValue ] .
    --> RESULT ecode:intValue

TLB_REMOVE id:intValue [ parentid:intValue ] .
    --> RESULT ecode:intValue .

TLB_SELECT id:intValue [ parentid:intValue ] [ select:BoolValue ] .
    --> RESULT ecode:intValue .

TLB_SELECT_ALL .
    --> RESULT ecode:intValue .

TLB_SHOW .
    --> RESULT ecode:intValue selectedids:idlist [ endbutton:strValue ] .
```

Errorcodes:

> 0 = No error
>
> 1 = Dialogue aborted by the user
>
> 2 = Before calling this function, `TLB_CREATE` has to be executed
>
> 3 = An argument or parameter is missing
>
> 4 = An ID provided is invalid

<u>**Web Service:**</u>

```
SERVICE Start | Stop .
    --> RESULT ecode:intValue .

Start: start [ port:intValue ] [ backlog:intValue ] .

Stop: stop .
```

Starts or stops the ADONIS web service server.

16.18.2   Commands of the "Core" MessagePort

Every core command returns an error code (ecode), which is examinded after every call back. A value 0 means that the call ends successfully, any value other than 0 means that there is an error (and it will be displayed). There are different error codes for different errors. No concrete values should be used in the AdoScripts, as the error codes change in different ADONIS versions.

The IDs container is represented through a string, the enumeration of the IDs is separated through blanks (**" "**). The number of IDs in a container string **ids** can be calculated through **tokcnt (ids, " ")**. The ID number **x** can be separated through the **token (ids, x, " ")**.

> **ATTENTION:**   Every model, which was created or loaded through the "Core"-MessagePort must be released with DISCARD_MODEL.

The commands below allow for the retrieval of such information concerning the model: the model name, the version number in the short format, the version number in the long format, the thread ID, the model type name, the library ID, the library type ("bp" or "we") and the library access status ("write" - write access, "read" - read only "none" - model not loaded).

```
GET_CURRENT_LIBS .
    --> RESULT ecode:intValue applib:strValue
        bplib:strValue welib:strValue
        applibid:intValue bplibid:intValue welibid:intValue .


GET_ALL_MODELTYPES [ libtype: libType ] [ sep:strValue ] .
    --> RESULT ecode:intValue modeltypes:strValue .


libType : "bp" | "we" .

GET_USER_PREFERENCES preference:strValue .
    --> RESULT ecode:intValue val:strValue .




GET_ACCESS_MODE modelid:id .
    --> RESULT ecode:intValue access:AccessMode isolated:boolValue .


AccessMode : "read" | "write" .

SET_MODEL_ACCESS_MODE modelversionid:id read-access:boolValue .
    --> RESULT ecode:intValue .
```

Hint: read-access:1: read-only

```
GET_CLASS_ID GetClassIdByClassName | GetClassIdByObject .

GetClassIdByClassName : [ relation ] classname:strValue
        [ bp-library | we-library ] .
    --> RESULT ecode:intValue classid:intValue .


GetClassIdByObject : objid:id .
    --> RESULT ecode:intValue classid:intValue isrel:intValue .

GET_CLASS_NAME classid:id .
    --> RESULT ecode:intValue classname:strValue isrel:intValue .




GET_ALL_OBJS modelid:id .
    --> RESULT ecode:intValue objids:strValue .

GET_ALL_OBJS_OF_CLASSID modelid: id classid: id .
    --> RESULT ecode:intValue objids:strValue .

GET_ALL_OBJS_OF_CLASSNAME modelid:id classname:strValue .
    --> RESULT ecode:intValue objids:strValue .

GET_ALL_OBJS_WITH_ATTR_VAL modelid:id classid:id .
        attrid:id val:str .
    --> RESULT ecode:intValue objids:strValue .

GET_OBJ_ID GetObjIdByObjName | GetObjIdByObjIds .
    --> RESULT ecode:intValue objid:intValue .


GetObjIdByObjName : modelid:id classid:id objname:strValue .


GetObjIdByObjIds : modelid:id classid:id
        objid1:id objid2:id .

GET_OBJ_NAME objid:id .
```

```
    --> RESULT ecode:intValue objname:strValue .



GET_CONNECTORS objid:id [ in ] [ out ] .
    --> RESULT ecode:intValue objids:strValue .

GET_ALL_CONNECTORS modelid:intVal .
    --> RESULT ecode:intValue objids:strValue .

GET_CONNECTOR_ENDPOINTS objid:id .
    --> RESULT ecode:intValue fromobjid:intValue toobjid:intValue .



GET_ALL_NB_ATTRS classid:id .
    --> RESULT ecode:intValue attrids:strValue .



GET_REC_ATTR_ROW_COUNT objid:id attrid:id .
    --> RESULT ecode:intValue count:intValue .

GET_REC_ATTR_ROW_ID objid:id attrid:id index:intValue .
    --> RESULT ecode:intValue rowid:id .

GET_ALL_REC_ATTR_ROW_IDS objid:id attrid:id .
    --> RESULT ecode:intValue rowids:strValue .

GET_REC_CLASS_ID attrid:id .
    --> RESULT ecode:intValue classid:id .

GET_RECORD_MULTIPLICITY attrid:id .
    --> RESULT ecode:intValue multiplicity:intValue .

GET_FACET_ENUMERATIONDOMAIN attrid:id .
    --> RESULT ecode:intValue val:stralue .



ADD_REC_ROW objid:objId attrid:attrId [ predrowid:predrowId ] .
    --> RESULT ecode:intValue rowid:id .

MOVE_RECORD_ROW modelid:id objid:id
        attrid:id rowid:id index:intValue .
    --> RESULT ecode:intValue

REMOVE_REC_ROW objid:id attrid:id rowid:id .
    --> RESULT ecode:intValue .



GET_APPMODEL_ID appmodelname:strValue .
    --> RESULT ecode:intValue appmodelid:intValue .

GET_ALL_APPMODEL_IDS .
    --> RESULT appmodelids:idList .

GET_APPMODEL_INFO appmodelid:id .
    --> RESULT ecode:intValue appmodelname:strValue
        bpmodelids:strValue wemodelid:intValue
        onthreads:boolValue onversions:boolValue .
```

```
GET_ATTR_ID classid:ClassID attrname:strValue .
    --> RESULT ecode:intValue attrid:id .


ClassID : id | bp-model | we-model .

GET_ALL_ATTRS OfModelType | OfClass .
    --> RESULT ecode:intValue attrids:strValue .


OfModelType : modeltype:strValue [ with-hidden-attrs ] .


OfClass : classid:id [ without-iattrs ] [ with-cattrs ] .

GET_ALL_ATTRS_OF_TYPE classid: id attrtype:attrType .
    --> RESULT ecode:intValue attrids:strValue .


attrType : "integer" | "double" | "string" | "distribution" |
           "time" | "enumeration" | "enumerationlist" | "longstring" |
           "programcall" | "interref" | "expression" | "record" |
           "attrprofref" | "date" | "datetime" .

GET_ATTR_NAME attrid:id .
    --> RESULT ecode:intValue attrname:strValue .

GET_ATTR_TYPE attrid:id .
    --> RESULT ecode:intValue attrtype:strValue .

GET_ATTR_VAL objid:id attrid:id
         [ as-string ] [ format:strValue ] [ sep:strValue ] [ core-value ] .
    --> RESULT ecode:intValue val:anyValue .
```

If **as-string** is specified, **val** will be of type STRING, regardless of the real type of the attribute. In all other cases val will be of type REAL (for DOUBLE attributes), INTEGER (for INTEGER attributes), or TIME (for TIME attributes). All other attribute types are automatically returned as STRING.

If the attribute is of the type INTEREF and **format** is given, the *formatStrValue* will be received as a response. However, *formatStrValue* will contain the following changes: **%o** replaced with the object name, the **%c** with the class name, **%m** replaced with the model name and **%t** with the model type name. In order to separate multiple references in the INTERREF attribute, **sep** can be used. By default it is **"\r\n"**.

If the parameter **core-value** is set, the content of the attribute will be shown, however, in all other cases, the content is shown on the user interface.

```
SET_ATTR_VAL [ as-string ] objid:id
         attrid:id val:anyValue .
    --> RESULT ecode:intValue .



GET_INTERREF objid:id attrid:id index:intValue .
    --> RESULT ecode:intValue tmodeltype:strValue
        tmodelname:strValue tmodelver:strValue
        ( type:"modelreference" | ( type:"objectreference"
        tclassname:strValue tobjname:strValue ) ) .

GET_INTERREF_COUNT objid:id attrid:id .
    --> RESULT ecode:intValue count:intValue .

GET_INTERREF_TYPE attrid:id .
    --> RESULT ecode:intValue type:Type .
```

Type : **"model"** | **"instance"** .

 ADD_INTERREF objid:*intValue* **attrid:***intValue Target*.
     --> **RESULT ecode:***intValue* .

Target : *TargetByName TargetByID* .

TargetByName : **tmodelname:***strValue* **tmodeltype:***strValue*
         [**tclassname:***strValue* **tobjname:***strValue*] .

TargetByID : **tobjid:***intValue* | **tmodelid:***intValue* .

 REMOVE_INTERREF objid:*id* **attrid:***id* **index:***intValue* .
     --> **RESULT ecode:***intValue* .

REMOVE_ALL_INTERREFS objid:*id* **attrid:***id* .
     --> **RESULT ecode:***intValue* .

GET_DANGLING_INTERREFS modelid:*id* .
     --> **RESULT ecode:***intValue* **reftext:***RefText* .
RefText : { **REF srcmodelid:***id* **srcobjid:***id* **srcattrid:***id*
         srcobjname:*strValue* **targettype:***TargetType* **tmodeltype:***strValue*
         tmodelname:*strValue* **tversion:***strValue* **tclassname:***strValue*
         tobjname:*strValue* } .
TargetType : **model** | **object** .

The hanging references will be displayed in the form of a LEOgramm. e.g. the number of the hanging
references can be shown via the command

    LEO parse:(refText) get-elem-count:count

.

GET_DANGLING_INTERREFS_OF_AP apversionid:*id* .
     --> **RESULT ecode:***intValue* **reftext:***RefText* .
RefText : { **REF srcmodelid:***id* **srcobjid:***id* **srcattrid:***id*
         srcobjname:*strValue* **targettype:***TargetType* **tmodeltype:***strValue*
         tmodelname:*strValue* **tversion:***strValue* **tclassname:***strValue*
         tobjname:*strValue* } .
TargetType : **model** | **object** .

GET_INCOMING_INTERREFS *Target* .
     --> **RESULT ecode:***intValue* **refText:***References* .

Target : **objid:***id* | **modelid:***id* .

References : { **REF** *SourceInfo TargetInfo* .

SourceInfo : **srcmodelid:***id* **srcobjid:***id*
         srcattrid:*id* **srcobjname:***strValue* .

TargetInfo : **targettype:***TargetType* *TargetModelInfo* [*TargetObjectInfo*] .

TargetType : **"object"** | **"model"** .

TargetModelInfo : **tmodeltype:***strValue* **tversion:***strValue* .

TargetObjectInfo : **tclassname:***strValue* **tobjname:***strValue* .

MOVE_INCOMING_INTERREFS fromobjid:*id* **toobjid:***id*.
     --> **RESULT ecode:***intValue* .

```
LOAD_MODEL modelid:id [ read-access ] .
    --> RESULT ecode:intValue isloaded:intValue .


SAVE_MODEL modelid:id .
    --> RESULT ecode:intValue .


SAVE_MODEL_AS SaveModelAsNewVersion | SaveModelAsNewModel .

SaveModelAsNewModel : modelid:id basename:strValue
        version:strValue [ mgroupids:idlist ] [ moverefs ] .
    --> RESULT ecode:intValue newthreadid:id
        newmodelid:id refids:idlist .
```

It creates and saves a new model (with its new name).

```
SaveModelAsNewVersion : modelid:id version:strValue .
    --> RESULT ecode:intValue newmodelid:id .
```

It creates and saves a new version of a model. The model name stays the same.

```
DISCARD_MODEL modelid:id .
    --> RESULT ecode:intValue .



EXECUTE_PROGRAMCALL objid:id attrid:id .
    --> RESULT result:intValue .



EVAL_EXPRESSION coreExpr [ objid:id | modelid:id ] .
    --> RESULT ecode:intValue result:value .

value : double | integer | string | time .
```

Calculates a CoreExpression (see chap. 10., p. 561) without the need for an EXPRESSION attribute. **objid** must be quoted, if the expression is object-related and the model id needs a model related expression. The type of the **result** is the same as the result of the expression.

Example:

```
    EVAL_EXPRESSION (aval ("time 1") + aval ("time2")) objid:(objid)



UPDATE_EXPR_ATTRS modelid:id [ synchronous:intValue ] .

GET_EXPR_TEXT classid:id attrid:id .
    --> RESULT ecode:intValue expr:strValue .

UPDATE_ALL_EXPR_ATTRS [ synchronous:boolValue ] .

GET_EXPR_TEXT classid:id attrid:id .
    --> RESULT ecode:intValue expr:strValue .

GET_EXPR_TEXT objid:id attrid:id .
    --> RESULT ecode:intValue expr:strValue .

SET_EXPR_TEXT objid:id attrid:id expr:strValue .
    --> RESULT ecode:intValue .
```

```
GET_EXPR_UPDATE .
    --> RESULT ecode:intValue synchronous:OnOrOff asynchronous:OnOrOff
          count:intValue interval:intValue
          counttotal:intValue countstale:intValue .

OnOrOff : "on" | "off" .

SET_EXPR_UPDATE synchronous:boolValue asynchronous:boolValue
    count:intValue interval:intValue .
    --> RESULT ecode:intValue .



CREATE_APP_MODEL strValue bpmodelids:strValue
wemodelid:id [ points-on-thread ] .
    --> RESULT ecode:intValue appmodelid:id .



SET_CHECK_ACCESS_STATE State .
State:  on | off .



GET_MODELGROUP_NAME mgroupid:id .
    --> RESULT ecode:intValue mgroupname:strValue .

SET_MODELGROUP_NAME mgroupid:id mgroupname:strValue .
    --> RESULT ecode:intValue .

GET_MGROUP_SUBGROUPS [ mgroupid:id ] .
    --> RESULT ecode:intValue submgroupids:strValue

GET_MODELGROUP_CHILDREN mgroupid:id [ recursive ] .
    --> RESULT ecode:intValue submgroupids:strValue .

GET_MODELGROUP_PARENT mgroupid:id .
    --> RESULT ecode:intValue parentmgroupid:intValue .

GET_MODELGROUP_ACCESS mgroupid:id .
    --> RESULT ecode:intValue access:Access .

SET_MODELGROUP_ACCESS mgroupid:id usergroup:strValue access:Access .
    --> RESULT ecode:intValue .

Access : "none" | "read" | "write" .

GET_ROOT_MODELGROUP_ID .
    --> RESULT ecode:intValue mgroupid:intValue .

GET_MODELGROUP_MODELS mgroupid:id [ getversionids ] .
    --> RESULT ecode:intValue modelids:strValue .

GET_MODELGROUP_REFERENCES mgroupid:id .
    --> RESULT ecode:intValue modelids:strValue .

GET_MODELGROUP_REFERENCE_THREAD refid:id .
    --> RESULT ecode:intValue threadid:intValue .

GET_MODELGROUPS_OF_MODELTHREAD threadid:id .
    --> RESULT ecode:intValue refids:strValue .

GET_MODELGROUPS_OF_MODELVERSION modelid:id .
```

```
    --> RESULT ecode:intValue mgroupids:strValue .



CREATE_MODELGROUP supermgroupid:id mgroupname:strValue .
    --> RESULT ecode:intValue mgroupid:id .

DELETE_MODELGROUP mgroupid:id .
    --> RESULT ecode:intValue .



CREATE_MODELGROUP_REFERENCE mgroupid:id threadid:id .
    --> RESULT ecode:intValue refid:intValue

COPY_MODELGROUP_REFERENCE refid:intValue targetmgroupid:intValue .
    --> RESULT ecode:intValue createdrefid:intValue .

MOVE_MODELGROUP_REFERENCE refid:id targetmgroupid:id .
    --> RESULT ecode:intValue .

DELETE_MODELGROUP_REFERENCE refid:intValue .
    --> RESULT ecode:intValue .



CREATE_MODEL modeltype:strValue modelname:strValue
        version:strValue mgroups:idList .
    --> RESULT ecode:intValue modelid:id
        threadid:id refids:strValue .
```

> **Hint:** When creating models, the user should give the IDs of the model groups (*idString*), in which the models should be contained. The IDs of many model groups are separated by blanks.

```
RENAME_MODEL modelid:id basename:strValue version:strValue .
    --> RESULT ecode:intValue .

DELETE_MODEL modelid:id .
    --> RESULT ecode:intValue .

UPDATE_MODEL_LIST  .
    --> RESULT ecode:intValue .



CREATE_OBJ modelid:id classid:id objname:strValue .
    --> RESULT ecode:intValue objid:id .

DELETE_OBJ modelid:id objid:id .
    --> RESULT ecode:intValue .

DELETE_OBJS modelid:id objids:idList .
    --> RESULT ecode:intValue errobjs:idList .



CREATE_CONNECTOR modelid:id fromobjid:id
        toobjid:id classid:id .
    --> RESULT ecode:intValue objid:id .

DELETE_CONNECTOR ConnectorID | FromToIDs .
    --> RESULT ecode:intValue .
```

ConnectorID : **modelid:**id **objid:**id .

FromToIDs : **modelid:**id **fromobjid:**id
            toobjid:id **classid:**id .

UPDATE_ALL_ATTRPROFS .
    --> **RESULT ecode:**intValue .

UPDATE_SINGLE_ATTRPROF apversionid:id .
    --> **RESULT ecode:**intValue .

CREATE_ATTRPROF_DIRECTORY apdirname:strValue [**superapdirid:**id] .
    --> **RESULT ecode:**intValue **apdirid:**id .

RENAME_ATTRPROF_DIRECTORY apdirid:id **apdirname:**strValue .
    --> **RESULT ecode:**intValue .

DELETE_ATTRPROF_DIRECTORY apdirid:id .
    --> **RESULT ecode:**intValue

GET_ROOT_ATTRPROFDIR_ID .
    --> **RESULT ecode:**intValue **apdirid:**id .

GET_ATTRPROF_DIRECTORY_NAME apdirid:id .
    --> **RESULT ecode:**intValue **apdirname:**strValue .

GET_ALL_ATTRPROF_SUBDIRS [**apdirid:**id] .
    --> **RESULT ecode:**intValue **apdirids:**ids .

GET_ATTRPROF_SUPERDIR apdirid:id .
    --> **RESULT ecode:**intValue **superapdirid:**intValue .

GET_ATTRPROF_CLASS_OF_THREAD apthreadid:id .
    --> **RESULT ecode:**intValue **apclassid:**id .

GET_ATTRPROF_THREAD_NAME apthreadid:id .
    --> **RESULT ecode:**intValue **apthreadname:**strValue .

GET_ATTRPROF_THREAD_ID_OF_NAME apthreadname:strValue **apclassname:**strValue .
    --> **RESULT ecode:**intValue **apthreadid:**intValue .

GET_ATTRPROF_THREAD_OF_VERSION apversionid:id .
    --> **RESULT ecode:**intValue **apthreadid:**strValue .

GET_ALL_ATTRPROF_THREADS_IN_DIR apdirid:id .
    --> **RESULT ecode:**intValue **apthreadids:**ids .

GET_ALL_ATTRPROF_VERSIONS_OF_THREAD apthreadid:id .
    --> **RESULT ecode:**intValue **apversionids:**ids .

GET_REFERENCED_ATTRPROF_VERSION_ID modelid:id
            objid:id **attrid:**id .
    --> **RESULT ecode:**intValue **apversionid:**intValue **apthreadname:**strValue .

RENAME_ATTRPROF_THREAD apthreadid:id **apthreadname:**strValue .
    --> **RESULT ecode:**intValue

```
DELETE_ATTRPROF_THREAD apthreadid:id .
    --> RESULT ecode:intValue .



CREATE_ATTRPROF_VERSION apthreadid:id apversionstr:strValue .
    --> RESULT ecode:intValue apversioniod:intValue .

CREATE_ATTRPROF_VERSION_EXT apclassid:id apdirid:id
        apthreadname:strValue apversionstr:strValue .
    --> RESULT ecode:intValue apversionid:id .

DELETE_ATTRPROF_VERSION apversionid:id .
    --> RESULT ecode:intValue .

GET_ATTRPROF_VERSIONSTRING apversionid:id .
    --> RESULT ecode:intValue apversionstr:strValue .

GET_ATTRPROF_VERSION_USAGE apversionid:id .
    --> RESULT ecode:intValue usage:strValue .

GET_ATTRPROF_CLASS_OF_VERSION apversionid:id .
    --> RESULT ecode:intValue apclassid:intValue .

GET_ALL_ATTRPROFS_IN_MODEL modelid:id .
    --> RESULT ecode:intValue apversions:strValue .



GET_ALL_MODEL_VERSIONS [ modeltype:strValue ] [ only-write-access ]
    --> RESULT ecode:intValue modelversionids:ids .

GET_ALL_MODEL_THREADS [ modeltype:strValue ] [ only-write-access ] .
    --> RESULT ecode:intValue modelthreadids:ids .

GET_ALL_MODEL_VERSIONS_OF_THREAD modelthreadid:id .
    --> RESULT ecode:intValue modelversionids:ids .

GET_ALL_REFERENCING_MODELS modelids:ids .
    --> RESULT ecode:intValue sourcemodelids:strValue .

GET_MODEL_BASENAME modelid:intValue .
    --> RESULT ecode:intValue basename:strValue .

GET_MODEL_CHANGECOUNTER modelid:intValue .
    --> RESULT ecode:intValue changecounter:intValue .

GET_MODEL_ID GetModelIdByModelName | GetModelIdByObject .
    --> RESULT ecode:intValue modelid:intValue .

GetModelIdByModelName : modelname:strValue I                 [ version:strValue ]
modeltype:strValue .

GetModelIdByObject : objid:id .

GET_MODEL_INFO modelid:id .
    --> RESULT ecode:intValue modelname:strValue ver:strValue
        version:strValue threadid:id modeltype:strValue
        libid:id libname:strValue access:Access .

Access : "none" | "write" | "read" .

GET_MODEL_MODELTYPE modelid:id .
```

```
    --> RESULT ecode:intValue modeltype:strValue .

GET_MODEL_THREAD_OF_VERSION modelversionid:id .
    --> RESULT ecode:intValue modelthreadid:intValue .

GET_MODEL_VERSION modelversionid:id .
    --> RESULT ecode:intValue version:strValue .

GET_REFERENCED_MODELS modelid:id references:{ References } .
    --> RESULT ecode:intValue modelids:strValue .


References : { From } .

From : FROM modelTypeName { Follow } .

Follow : FOLLOW classname:classname attrname:attrname
         [ recattrname:recordattrname ] depth:intValue type:RefFollowType .

RefFollowType : main | sub .
```

It calculates the referenced models, but the specified references can be followed. Each model type can have its own (specially defined) way of how references should be perceived.

```
GET_ATTRPROFCLASS_ID apclassname:strValue .
    --> RESULT ecode:intValue apclassid:intValue .

GET_ATTRPROFCLASS_OF_ATTR attrid:id .
    --> RESULT ecode:intValue apclassid:intValue .



GET_ENV_STRING envvar:strValue .
    --> RESULT envstr:strValue .

SET_ENV_STRING envvar:strValue envstr:strValue .
    --> RESULT ecode:(0|-1) .

GET_OS_INFO .
    --> RESULT os: os:"os2" | os: "win" winver_major:intValue
        winver_minor:intValue winver_build:intValue
        winver_platform:intValue winver_csdversion:intValue .

GET_PRODUCT_VERSION ecode:intValue .
    --> RESULT version:intValue .



IS_ATTRPROF_CLASS classid:intValue .
    --> RESULT is_apclass:boolValue .

IS_ATTRPROF_THREAD objid:id .
    --> RESULT is_apthread:boolValue .

IS_ATTRPROF_VERSION objid:id .
    --> RESULT is_apversion:boolValue .

IS_MODEL_LOADED modelid: .
    --> RESULT ecode:intValue isloaded:boolValue .

IS_VERSIONING_ENABLEDid .
    --> RESULT versioning:boolValue .
```

```
LOCK_OBJECT objid:intValue [ type:LockType ] .
    --> RESULT ecode:intValue already_locked:boolValue .

LockType : "write" | "shared" .

UNLOCK_OBJECT objid:intValue .
    --> RESULT ecode:intValue not_locked:boolValue .



SAVE_LIBRARY libid:id .
    --> RESULT ecode:intValue .



ECODE_TO_ERRTEXT ecode:intValue .
    --> RESULT errtext:strValue .



CREATE_COPYBUFFER index:intValue .
    --> RESULT ecode:intValue .

FILL_COPYBUFFER index:intValue
        instids:strValue relinstids:strValue .
    --> RESULT ecode:intValue .

PASTE_COPYBUFFER index:intValue modelid:intValue moverefs:boolValue .
    --> RESULT ecode:intValue
        instids:strValue relinstids:strValue  reftext:strValue.

DELETE_COPYBUFFER index:intValue .
    --> RESULT ecode:intValue .
```

The following commands are only available in the Administration Toolkit:

```
GET_LIB_ID libname:strValue .
    --> RESULT ecode:intValue errtext:strValue libid:intValue
        type:LibType .

GET_LIB_NAME libid:id .
    --> RESULT ecode:intValue errtext:strValue libname:strValue
        type:LibType .

LibType: applib | bplib | welib .

GET_ALL_APPLIBS .
    --> RESULT ecode:intValue errtext:strValue applibids:strValue .

LOAD_LIB libid:id [ write-protected:boolValue ] .
    --> RESULT ecode:intValue errtext:strValue
        applibid:intValue bplibid:intValue welibid:intValue .

DISCARD_LIB .
    --> RESULT ecode:intValue errtext:strValue libid:intValue .
```

16.18.3   Commands of the "CoreUI" MessagePort

```
MODEL_SELECT_BOX [ loaded-models ] [ multi-sel ] [ show-all-versions ]
         [ threads ] [ presel-modelids:strValue ] [ without-models ]
         [ mgroup-sel ] [ presel-mgroupids:strValue ] [ with-app-models]
         [ SingleMTFilter | MultiMTFilter ]
         [ title:strValue ] [ boxtext:strValue ] [ oktext:strValue ]
         [ w:intValue h:intValue ] [ min-w:intValue min-h:intValue ]
         [ extra:{ Extra } ] .
  --> RESULT endbutton:strValue [ modelids:idList | threadids:idList ]
         [ mgroupids:idList ] [ appmodelids:idList ] [ extraValues ] .
```

SingleMTFilter : `modeltype:strValue .`

MultiMTFilter : `modeltype1:strValue modeltype2:strValue ... .`

Extra : `{ CheckBox } .`

CheckBox : `CHECKBOX chbText [ checked:intValue ] result-var:varName .`

loaded-models: if quoted, only the loaded models will be displayed. Otherwise, all the models stored in the database are shown.

multi-sel: if not quoted, only one model can be selected.

show-all-versions: if the versioning is activated in the application library, quotation of this attribute means that the user can choose any model version. Otherwise, the user can only choose the latest available model version.

threads: if the versioning is activated in the application library, quotation of this attribute means, that only the whole thread can be selected and no version numbers can be displayed.

without-models: if quoted, no models will be displayed. It is useful at the model group selection.

The **title** is the text from the dialogue title bar, **boxtext** is the address of the model selection list, **oktext** is the text of the OK button (action naming for choosing the models). The dialogue extension can be defined via **w** (width) and **h** (height). The minimum size of the dialogue can be defined with **min-w** (minimum width) and **min-h** (minimum height).

mgroup-sel: if quoted, modelgroups with the parameter **presel-mgroupids:** can be selected.

with-app-models: if quoted, the list with the application models defined in the current library will be built.

extra: the model selection list can be extended with additional check boxes. The **extra** LEOgramm defines a list of such check boxes and specifies variables, which are added to the check box conditions (0 for not selected, 1 for selected) after the dialogue execution.

Example:

```
 ...
MODEL_SELECT_BOX
 ...
extra:{ CHECKBOX "With submodels" checked:0 result-var:sm }
```

The **result-var** variable appears in the **RESULT** string of the command. In this particular case it would be **sm:0**, or **sm:1**.

```
ATTRPROF_SELECT_BOX [ multi-sel ] [ with-mgmt-functions ] [ show-all-versions ]
    [ apgroup-sel ] [ without-attrprofs]
    [ title:strValue ] [ boxtext:strValue ] [ oktext:strValue ]
```

```
      [ w:intValue h:intValue ] [ min-w:intValue min-h:intValue ] .
  --> RESULT endbutton:strValue attrprofids:strValue apgroupids:strValue .
```

EXEC_MT_FILTER_DLG .

This command has no parameters and returns no result.

SET_OBJ_FOREGROUND objid:*id* **color:***colorSpec* .

Defines the font colour of the named object in the tabular view.

SET_OBJ_BACKGROUND objid:*id* **color:***colorSpec* .

Defines the background colour of the named object in the tabular view.

RESET_OBJ_FOREGROUND objid:*id* .

RESET_OBJ_BACKGROUND objid:*id* .

16.18.4   Commands of the "Application" MessagePort

```
GET_PATH [ strValue ] [ path-only [ append-delimiter ] ] .
     --> RESULT path:strValue .
```

EXEC_COMP_POPUP .

```
EXEC_PRTSETUP_DLG .
     --> "OK" │ "cancel" .
```

SET_ACTIVE_COMP *intValue* .

```
GET_ACTIVE_COMP .
     --> intValue .
```

ENABLE_COMP { *Component* } .

DISABLE_COMP { *Component* } .

```
Component:   [ all ] [ acquisition ] [ modeling ] [ analysis ]
             [ simulation ] [ evaluation ] [ importexport ] .
```

```
GET_COMP_ENABLED intValue .
     --> strValue .
```

```
GET_ACCESS_PERM Component .
     --> RESULT ecode:intValue access:boolValue .
```

Component :   **"aqc"** │ **"modelaccess"** │ **"modeldescr"** │ **"modelstate"** │
       "stdqueries" │ **"defqueries"** │ **"reltables"** │ **"anaeval"** │
       "pathsim" │ **"volsim"** │ **"steadywlsim"** │ **"fixedwlsim"** │
       "agents" │ **"flowmark"** │ **"rescomp"** │ **"evalqueries"** │
       "dyneval" │ **"adl"** │ **"xml"** │ **"fdl"** │ **"case40"** │ **"objectif"** │
       "ccc" │ **"docutk"** │ **"attrprof"** │ **"classhier"** │ **"nonewdb"** .

Hint: access has the value **1** if the component is enabled, otherwise **0**.

CLOSE .

EXIT .

GET_VERSION .
    --> **RESULT productname:***strValue* **toolkit:***strValue*
       version:*strValue* **patch:***strValue* **buildstr:***strValue* .

GET_DATE_TIME [**date-format:***strValue*] [**time-format:***strValue*] .
    --> **RESULT date:***strValue* **time:***strValue* .

GET_USER .
    --> **RESULT user:***strValue* .

GET_USER_DISPLAYNAME .
    --> **RESULT user:***strValue* .

In case of the Single-Sign-on, it displays the user names instead of the GUID strings.

GET_DB_NAME .
    --> **RESULT dbname:***strValue* .

GET_ONLINE_SINCE [**date-format:***strValue*] [**time-format:***strValue*] .
    --> **RESULT date:***strValue* **time:***strValue* .

GET_MAX_USER_COUNT .
    --> **RESULT maxuser:***intValue* .

GET_CUSTOMER_NUMBER .
    --> **RESULT ecode:***intValue* **customernumber:***strValue* .

GET_SCREEN_RES .
    --> **RESULT ecode:***intValue* **w:***intValue* **h:***intValue* .

```
MESSAGE_SEND ( userids:strValue [ usernames:strValue ] |
        usernames:strValue [ userids:strValue ] )
        subject:strValue message:strValue
        [ express:boolValue ] .
    --> RESULT ecode:intValue messageid:intValue .
```

```
SET_STATUS strValue .
```

```
SET_MENU_ITEM_HDL component:Component
        item:strValue { AdoScript } .
    --> RESULT ecode:intValue .
```

```
SET_MENU_ITEM_CHECKED component:Component
        item:strValue checked: boolValue .
    --> RESULT ecode:intValue .
```

```
REMOVE_MENU_ITEM component:Component item:strValue .
    --> RESULT ecode:intValue .
```

Component : **"acquisition"** | **"modeling"** | **"analysis"** |
        "simulation" |**"evaluation"** | **"importexport"** | **"all"** .

```
INSERT_CONTEXT_MENU_ITEM context:Context
        item:strValue [ pos:strValue ] .
    --> RESULT ecode:intValue .
```

```
REMOVE_CONTEXT_MENU_ITEM context:Context
        item:strValue .
    --> RESULT ecode:intValue .
```

```
SET_CMI_SELECT_HDL context:Context
        item:strValue { AdoScript } .
    --> RESULT ecode:intValue .
```

Context : **"drawingarea.general"** | **"drawingarea.mobject"**
        "drawingarea.connector" |**"drawingarea.swimlane"**
        "explorer.db" |**"explorer.windows"** .

```
SET_SI_VISIBLE name:strValue visible:boolValue .
    --> RESULT ecode:intValue .
```

```
SET_ICON_CHECKED name:strValue checked:boolValue .
    --> RESULT ecode:intValue .
```

```
SET_ICON_VISIBLE [ component:Component ]
        name:strValue visible:boolValue .
    --> RESULT ecode:intValue .
```

Component : **"acquisition"** | **"modeling"** | **"analysis"** |
        "simulation" |**"evaluation"** | **"importexport"** | **"all"** .

Shows / hides the smarticons in the quickaccess bar.

```
SET_ICON_CLICK_HDL [ component:Component ]
        name:strValue { AdoScript } .
```

```
    --> RESULT ecode:intValue .

Component :   "acquisition" | "modeling" | "analysis" |
        "simulation" |"evaluation" | "importexport" | "all" .

INSERT_ICON component:Component IconOrSep
        [ pos:intValue ] .
    --> RESULT ecode:intValue .

IconOrSep : Icon | separator .

Icon : name:strValue bitmap:strValue text:strValue .

Component :   "acquisition" | "modeling" | "analysis" |
        "simulation" |"evaluation" | "importexport" | "all" .
```

16.18.5   Commands of the "Modeling" MessagePort

The error codes (**ecode**), IDs and ID containers are the same as in the Core MessagePort (see chap. 16.18.2, p. 628).

Active Model:

```
ACTIVATE_MODEL intValue .

GET_ACTIVE_MODEL .
    --> intValue
```

This command works only with **SEND**, and not with **CC**:

```
    SEND "GET_ACTIVE_MODEL" to:"Modelling" answer:temp
    SET modelid:(VAL temp) # to change string into Integer

GET_ACT_MODEL .
    --> RESULT modelid:intValue .
```

GET_ACT_MODEL can be used together with **CC**. If there is no model window opened, the **modelid** has the value -1.

```
modelId : intValue .
```

Open / Modified / Save / Close:

```
EXEC_NEW_DLG [ modelname:strValue ] [ version:strValue ]
        [ modeltype:strValue ] [ target-mgroupid:intValue ]
        [ show-models [ show-all-versions ] ] .
    --> intValue .

OPEN modelids:strValue [ with-submodels ]
        [ write-protected ] [ minimized ] .
    --> RESULT ecode:intValue opened:strValue invalid:strValue .

IS_OPENED modelid:intValue .
    --> RESULT isopened:intValue .
```

If the model is opened with the model ID, the **isopened** gets the value 1. Otherwise the value is 0. Therefore the variable can be put into defined conditions: **IF (isopened)** ...

```
GET_OPENED_MODELS .
    --> RESULT ecode:intValue modelids:strValue .
```

The single **modelids** are separated by blanks.

```
CREATE_WINDOW_FOR_LOADED_MODEL modelid:intValue .
    --> RESULT ecode:intValue .
```

```
SET_MODIFIED [ id ] [ modified: intValue ] .
```

```
GET_ALL_MODIFIED .
    --> RESULT modelids:strValue .
```

Returns a string with the IDs of all models that contain unsaved changes.

```
GET_MODIFIED_COUNT [ id ] .
    --> RESULT modified:intValue .
```

It gives the number of unsaved changes.

```
SAVE  [ modelid:intValue ] .
    --> RESULT ecode:intValue .
```

```
SAVE_ALL [ quiet ] .
    --> RESULT ecode:intValue .
```

```
CLOSE [ modelid: intValue ] [ quiet [ save ] ].
    --> RESULT ecode:intValue .
```

```
CLOSE_ALL [ quiet [ save ] ] .
    --> RESULT ecode:intValue .
```

```
CLEAR_UNDO_REDO modelid: intValue .
```

Deletes the contents of the undo/redo manager.

Print:

```
EXEC_PRINT_DLG [ modelid:id ] [ printer:strValue ]
        [ layout:strValue ] [ Scaling ]
        [ orientation:Orientation ] [ auto-execute ] .
    --> RESULT ecode:intValue .
```

```
Scaling: [ factor:intValue | fitonepage |
        pages-height:intValue | pages-width:intValue ] .
```

```
Orientation: "automatically" | "portrait" | "landscape" .
```

Calls the dialogue window "Print model" for the model with a given ID.

Cardinalities:

```
CHECK_CARDINALITIES [ modelid:intVal ] [ silent ] [ nosuccessmessage ].
    --> RESULT ecode:intValue .
```

If no **modelid** is supplied, the cardinality check is performed with the active model. With **silent** no error messages are shown to the user. With **nosuccessmessage**, the user will not receive a success prompt (if no errors are found during the check).

Representation:

```
GET_REPRESENTATION [ modelid:id ] .
    --> RESULT ecode:intValue representation:Representation classid:id .

SET_REPRESENTATION [ modelid:id ] representation:Representation
        [ classid:id ] .
    --> RESULT ecode:intValue .

Representation: "graphical" | "tabular" .
```

Modes:

```
SET_VIEW_MODE [ modelid:modelId ] mode-name:strValue .
    --> RESULT ecode:intValue .

GET_VIEW_MODE [ modelid:modelId ] .
    --> RESULT ecode:intValue modeltype:strValue modename:strValue .

GET_ALL_VIEW_MODES [ modelid:modelId ] .
    --> RESULT ecode:intValue modenames:strValue .
```

The model names on the list **modenames** are separated via **"\n"**. If the **ecode** has the value **1**, it means that an error was found.

```
GET_VISIBLE_CLASSES [ modelid:modelId ] .
    --> RESULT ecode:intValue classids:strValue .

GET_VISIBLE_RELATIONS [ modelid:modelId ] .
    --> RESULT ecode:intValue relationids:strValue .

modelId : intValue .
```

Drawing Area:

```
DYE intValue [ error-mark ] [ set-value:intValue ] [ make-visible ] .
```

Colours the outline of an object or of a connector in the "textmarker" - style. There are two different forms of colouring, which can be distinguished through the shown/hidden **error-mark**. Error marks will be confirmed and reset by the user by clicking on the highlighted object. The other form has an internal counter, which is through **DYE** raised and through **UNDYE** lowered or through **DYE set-value** will be set.

UNDYE *intValue* [**error-mark**] .

UNDYE_ALL [**modelid:***intValue*] .

SET_FOCUS_NODE objid:*intValue* .

REBUILD_DRAWING_AREA [**modelid:***id*] .

GET_VISIBLE_AREA [**modelid:***intValue*] .
    --> **RESULT ecode:***intValue* **x1:***measureValue* **y1:***measureValue*
        x2:*measureValue* **y2:***measureValue* .

GET_OBJECTS_WITHIN_AREA [**modelid:***modelId*]
        x1:*measureValue* **y1:***measureValue*
        x2:*measureValue* **y2:***measureValue* .
    --> **RESULT ecode:***intValue* **objids:***strValue* **relationids:***strValue* .

GET_PREV_SWIMLANE [**objid:***intValue*] .
    --> **RESULT ecode:***intValue* **objid:***intValue* .

GET_NEXT_SWIMLANE [**objid:***intValue*] .
    --> **RESULT ecode:***intValue* **objid:***intValue* .

SET_DRAWING_AREA_SIZE [**modelid:***modelId*]
        w:*measureValue* **h:***measureValue* .
    --> **RESULT ecode:***intValue* .

GET_DRAWING_AREA_SIZE [**modelid:***intValue*] .
    --> **RESULT ecode:***intValue* **w:***measureValue* **h:***measureValue* .

SET_ZOOM_FACTOR [**modelid:***intValue*] **zf:***intValue* .
    --> **RESULT ecode:***intValue* .

GET_ZOOM_FACTOR [**modelid:***intValue*] .
    --> **RESULT ecode:***intValue* **zf:***intValue* .

SET_GFX_SELECTED_AREA [**modelid:***modelId*]
        x1:*measureValue* **y1:***measureValue*
        x2:*measureValue* **y2:***measureValue* .
    --> **RESULT ecode:***intValue* .

GET_GFX_SELECTED_AREA [**modelid:***modelId*] .
    --> **RESULT ecode:***intValue* **x1:***measureValue* **y1:***measureValue*
        x2:*measureValue* **y2:***measureValue*.

REMOVE_GFX_SELECTED_AREA [**modelid:***modelId*] .
    --> **RESULT ecode:***intValue* .

SET_LAYOUT [*id* | **modelid:***id*] **layout:***strValue*
        [**orientation:***Orientation*] [*Scaling*] .
    --> **RESULT ecode:***intValue* .

Scaling : **factor:***intValue* | **fitonpage** |
        pages-height:*intValue* | **pages-width:***intValue*] .

Orientation : **"automatically"** | **"portrait"** | **"landscape"** .

SET_CONNECTOR_MARKS [*id* | **modelid:***id*]
        [**create:***boolValue*] .
    --> **RESULT ecode:***intValue* .

EXEC_GFX_DLG [*id* | **modelid:***id*] [**mode:***Mode*]
        [*RegionParams* | *PagesParams*]

```
            [ destination:Destination [ FileParams ]
            [ auto-execute ] .
        --> RESULT ecode:intValue .
```

Calls the dialogue "Generate graphics" for the model with the given ID.

```
GENERATE_GFX [ id | modelid:id ]
            mode:Mode [ RegionParams | PagesParams ]
            [ destination:Destination ] [ FileParams ] .
        --> RESULT ecode:intValue gfx-format:GfxFormat .
```

Generates a model graphic.

```
Mode: "region" | "pages" .
```

```
RegionParams: [ scale:intValue ] .
```

```
PagesParams: [ layout:strValue ] [ orientation:OrientationValue ]
            [ factor:intValue | fitonpage | pages-height:intValue | pages-
width:intValue ]
            [ dpi:intValue ] .
```

```
OrientationValue: "portrait" | "landscape" .
```

```
Destination: "file" | "clipboard" .
```

```
FileParams: filename:strValue .
```

```
GfxFormat: "bmp1" | "bmp24" | "bmp" | "pcx8" | "pcx24" | "pcx" |
            "jpeg" | "png" | "emf" | "svg" | .
```

```
GEN_GFX_FILE modelid:id filename:strValue gfx-format:ImageType ]
            [ scale:intValue ] [ bpp:intValue ] [ quality:intValue ] .
        --> RESULT ecode:intValue .
```

Generates a graphic file from the given model.

```
GEN_GFX_STR modelid:id gfx-format:ImageType ]
            [ scale:intValue ] [ bpp:intValue ] [ quality:intValue ] .
        --> RESULT ecode:intValue gfx:strValue .
```

```
ImageType: "bmp" | "gif" | "ico" | "jpeg" | "png" |
            "targa" | "tiff" | "wbmp" | "xpm" | .
```

Generates a graphic as base64 string.

```
COMPUTE_REGION_IMAGE_MAP [ modelid:id ] [ scale:intValue ] .
        --> RESULT ecode:intValue imgmap:strValue .
```


Object Selection:


```
DESELECT objid:intValue .
        --> RESULT ecode:intValue .
```

```
DESELECT_ALL [ modelid:intValue ] .
        --> RESULT ecode:intValue .
```

```
FIND objid:intValue .
        --> RESULT ecode:intValue .
```

Selects an object, or a connector as with a mouse click. Already highlighted objects or connectors are deselected.

```
GET_SELECTED [ modelid:id ] .
    --> RESULT ecode:intValue objids:strValue classid:id .

SELECT objid:intValue .
    --> RESULT ecode:intValue .

SELECT_ALL  [ modelid:intValue ] [ with-swimlanes ] .
    --> RESULT ecode:intValue .

SET_ATTR_ACCESS_MODE [ modeltype:ModelType
        modelid:id objid:id ] access-mode:AccessMode .
    --> RESULT ecode:intValue .

ModelType:  all │ strValue .

AccessMode:  "default" │ "write-protected" │ "full" .
```

Objects:

```
COPY_SELECTED [ modelid:modelId ] .
    --> RESULT ecode:intValue .

CUT_SELECTED [ modelid:modelId ] .
    --> RESULT ecode:intValue .

PASTE [ modelid:modelId ] x:measureValue y:measureValue .
    --> RESULT ecode:intValue .

ALIGN_SELECTED [ modelid:modelId ] vertically │ horizontally .
    --> RESULT ecode:intValue .

RUN_MODEL_NUMBERING modelid:intValue .
        [ no-result-window:boolValue ] .
    --> RESULT ecode:intValue .

RUN_NAME_GENERATION modelid:intValue .
    --> RESULT ecode:intValue changes:intValue .

SET_MOUSE_ACCESS ObjSpec [ access:boolVal ] .
    --> RESULT ecode:intValue .

ObjSpec: ClassByName │ SingleObj .

ClassByName: [ modelid:modelId ] classname:strValue .

SingleObj: objid:intValue .

SET_OBJ_VISIBLE [ objid:id ] [ visible:boolVal ] .
    --> RESULT ecode:intValue .

SET_ALL_OBJS_VISIBLE modelid:id [ visible:boolVal ] .
    --> RESULT ecode:intVal .

SET_OBJ_POS [ objid:id ] x:measureValue y:measureValue .
    --> RESULT ecode:intValue .
```

Notebook:

```
EXEC_NOTEBOOK objid:intValue .
    --> RESULT ecode:intValue .

CLOSE_NOTEBOOK objid:intValue .
    --> RESULT ecode:intValue .

CLOSE_ALL_NOTEBOOKS  [ modelid:intValue ] .
    --> RESULT ecode:intValue .

SHOW_NOTEBOOK_CHAPTER objid:intValue chapter:intValue chapter-page:intValue .
    --> RESULT ecode:intValue .

SET_NOTEBOOK_POS objid:intValue x:intValue y:intValue .
    --> RESULT ecode:intValue .

SET_NOTEBOOK_SIZE objid:intValue w:intValue h:intValue .
    --> RESULT ecode:intValue .

GET_NOTEBOOK_POS_SIZE objid:intValue .
    --> RESULT ecode:intValue x:intValue y:intValue
        w:intValue h:intValue .

REFRESH_PROFILEREFS .

modelId : intValue .

objId : intValue .
```

Autosave:

```
SET_AUTOSAVE Enabling changes:intValue .

Enabling : on | off | enabled:intValue .

GET_AUTOSAVE .
    --> RESULT enabled:boolValue changes:intValue .
```

Window:

```
SET_WINDOW_POS [ modelid: intValue ] x:intValue y:intValue .
    --> RESULT ecode:intValue .

SET_WINDOW_SIZE [ modelid: intValue ] w:intValue h:intValue .
    --> RESULT ecode:intValue .

GET_WINDOW_POS_SIZE [ modelid: intValue ] .
    --> RESULT ecode:intValue x:intValue y:intValue
        w:intValue h:intValue .
```

```
GET_WINDOW_STATE [ modelid:intValue ] .
    --> RESULT ecode:intValue state:StateValue .

SET_WINDOW_STATE [ modelid: intValue ] state:StateValue .
    --> RESULT ecode:intValue .

GET_MAX_ADONIS_WINDOW_SIZE .
    --> RESULT ecode:intValue w:intValue h:intValue .
```

Hint: Instead of **GET_MAX_ADONIS_WINDOW_SIZE**, **GET_MAX_MODEL_WINDOW_SIZE** should be used in the future.

```
GET_MAX_MODEL_WINDOW_SIZE .
    --> RESULT ecode:intValue w:intValue h:intValue .

MINIMIZE_ALL .

StateValue : "min" | "normal" | "max" .

modelId : intValue .
```

16.18.6   Commands of the "Analysis" MessagePort

```
RUN_ANALYTIC_EVALUATION modelid:intValue
        [ dpy:intValue ] [ hpd:intValue ]
        [ volume:intValue ] [ simoption:intValue ] [ accuracy:intValue ]
        [ maxlooplen:intValue ] [ maxpaths:intValue ]
        [ auto-close:boolValue ] [ no-result-window ]
        [ auto-save-results ] [ no-error-messages:boolValue ] .
    --> RESULT ecode:intValue errmsg:strValue .



EXEC_ANALYTIC_EVALUATION_START_DLG  [ modelid:intValue ]
        [ dpy:intValue ] [ hpd:intValue ]
        [ simoption:intValue ] [ accuracy:intValue ]
        [ maxlooplen:intValue ] [ maxpaths:intValue ]
        [ auto-close:boolValue ]
        [ no-result-window ] [ auto-save-results ] .
    --> RESULT ecode:intValue .
```

16.18.7   Commands of the "Simulation" MessagePort

```
EXEC_PATH_ANALYSIS_DLG [ modelid:intValue ]
        [ runs:intValue ] [ dpy:realValue ] [ hpd:realValue ]
        [ simoption:intValue ] [ auto-read-only:boolValue ]
        [ auto-close:boolValue ] [ auto-loop:boolValue ]
        [ setLoopDetectionCount:intValue ]
        [ randseed:intValue ] [ programcalls:boolValue ]
        [ loaded-models:boolValue ] [ no-error-messages:boolValue ] .
    --> RESULT ecode:intValue errmsg:strValue canceled:boolValue.



EXEC_VOLUME_ANALYSIS_DLG [ runs:intValue ]
        [ dpy:intValue ] [ hpd:intValue ]
        [ simoption:intValue ] [ auto-volume:boolValue ]
        [ auto-read-only:boolValue ] [ auto-close:boolValue ]
```

```
              [ auto-component:boolValue ] [ auto-loop:boolValue ]
              [ setLoopDetectionCount:intValue ] [ randseed:intValue ]
              [ programcalls:boolValue ] [ path-analysis:boolValue ]
              [ calculation:boolValue ] [ protocol:ProtocolSettings ]
              [ appmodelname:strValue ] [ no-error-messages:boolValue ] .
        --> RESULT ecode:intValue errmsg:strValue canceled:boolValue .
```

ProtocolSettings: **none** | **short-format** | **long-format** [**protfile:***strValue*] .

```
EXEC_STEADY_WORKLOAD_ANALYSIS_DLG [ runs:intValue ]
              [ day:intValue ] [ month:intValue ]
              [ simoption:intValue ] [ auto-read-only:boolValue ]
              [ auto-close:boolValue ] [ auto-component:boolValue ]
              [ randseed:intValue ] [ programcalls:boolValue ]
              [ activity-analysis:boolValue ] [ calculation:boolValue ]
              [ animation:boolValue ] [ protocol:ProtocolSettings ]
              [ appmodelname:strValue ] [ no-error-messages:boolValue ] .
        --> RESULT ecode:intValue errmsg:strValue canceled:boolValue .
```

ProtocolSettings: **none** | **short-format** | **long-format** [**protfile:***strValue*] .

```
EXEC_FIXED_WORKLOAD_ANALYSIS_DLG [ auto-read-only:boolValue ]
              [ auto-close:boolValue ] [ auto-component:boolValue ]
              [ randseed:intValue ] [ programcalls:boolValue ]
              [ activity-analysis:boolValue ] [ calculation:boolValue ]
              [ animation:boolValue ] [ protocol:ProtocolSettings ]
              [ simoption:intValue ]
              [ day-1:intValue ] [ month-1:intValue ]
              [ day-2:intValue ] [ month-2:intValue ] [ year-2:intValue ]
              [ day-3:intValue ] [ month-3:intValue ] [ year-3:intValue ]
              [ appmodelname:strValue ] [ no-error-messages:boolValue ] .
        --> RESULT ecode:intValue errmsg:strValue canceled:boolValue .
```

ProtocolSettings: **none** | **short-format** | **long-format** [**protfile:***strValue*] .

```
RUN_PATH_ANALYSIS modelid:intValue [ runs:intValue ]
              [ dpy:intValue ] [ hpd:intValue ]
              [ simoption:intValue ] [ auto-read-only:boolValue ]
              [ auto-close:boolValue ] [ auto-loop:boolValue ]
              [ setLoopDetectionCount:intValue ] [ randseed:intValue ]
              [ programcalls:boolValue ] [ no-error-messages:boolValue ]
              [ no-result-window ] [ auto-save-results:SaveMode ] .
        --> RESULT ecode:intValue errmsg:strValue .
```

SaveMode : **only-proc** | **activities-avg** | **activities-total** .

ecode returns the value 0 only if the simulation was executed successfully. Otherwise it has the value 1 (meaning that an error was found or that the user cancelled the process).

```
RUN_VOLUME_ANALYSIS appmodelname:strValue
              [ runs:intValue ] [ dpy:intValue ] [ hpd:intValue ]
              [ simoption:intValue ] [ adopt-results:ResultPeriodType ]
              [ auto-read-only:boolValue ] [ auto-close:boolValue ]
              [ auto-volume:boolValue ] [ auto-component:boolValue ]
```

```
                 [ auto-loop:boolValue ] [ setLoopDetectionCount:intValue ]
                 [ randseed:intValue ] [ programcalls:boolValue ]
                 [ path-analysis:boolValue ] [ calculation:boolValue ]
                 [ protocol:ProtocolType ] [ protfile:strValue ]
                 [ no-error-messages:boolValue ] [ no-result-window ]
                 [ process-results:{ ResultProcessingJobList } ] .
           --> RESULT ecode:intValue errmsg:strValue .

ProtocolType: none | short-format | long-format .

ResultProcessingJobList: { ResultProcessingJob } .

ResultProcessingJob: JOB mode:ResultMode
           [ classid:id relationclassid:id ]
           period-type:ResultPeriodType
           [ handler:procedureName ] [ show-browser ] .

ResultMode: "process-related" | "person-related" |
           "we-related" | "capacity-planning" | "resource-process-related" |
           "resource-related" | "resource-we-related" .

ResultPeriodType: "null" | "per-process" | "per-month" | "per-year" .

boolValue: 1 | 0 .



GET_TIME_BASE .
      --> RESULT ecode:intValue dpy:DaysPerYear hpd:HoursPerDay .

DaysPerYear : realValue .

HoursPerDay : realValue .



CHECK_ALL_TRANSITION_CONDITIONS modelid:intValue .
      --> RESULT ecode:intValue errtext:strValue text:strValue .
```

The system will check all the transition condiditons in a model. As a result the chain of characters will be given (**text**). It will get the format

```
{ RelationID CheckResult }
```

where the *CheckResult* can get the following values:

```
-   EXPECTED_PARENTHESIS
-   EXPECTED_LOGICAL_OPERATOR
-   ILLEGAL_OPERATOR
-   ILLEGAL_TERM
-   INCOMPLETE_COMPARISON
-   INCOMPLETE_CONDITION
-   NO_ERROR
-   TRANS_PROBABILITY
-   TREE_ALREADY_EXISTS
-   TREE_CONSISTS_OF_TRUE
-   UNKNOWN_OPERATOR
```

16.18.8   Commands of the "Evaluation" MessagePort

```
LOCK_SHELL [ sim:SimAlg ] .
```

```
        --> RESULT ecode:intValue

SimAlg:  path-analysis | volume-analysis |
         steady-workload-analysis | fixed-workload-analysis |
         ccc-is-analysis | ccc-plan-analysis .



UNLOCK_SHELL .
        --> RESULT ecode:intValue



EXEC_DYNAMIC_EVAL_MODULE_DLG moduleName
        [ applmodel:ApplModelName ]
        [ hpd:HoursPerDay ] [ dpy:DaysPerYear ]
        [ runs:intValue ] [ randseed:boolValue ]
        [ start:EvalStartPeriod end:EvalEndPeriod ]
        [ volume-check:BoolValue ] [ checkhist ] .
    --> RESULT ecode:intValue canceled:BoolValue .

ApplModelName : StrValue .

HoursPerDay : realValue .

DaysPerYear : realValue .

EvalStartPeriod : strValue .

EvalEndPeriod : strValue .



EXEC_DYNAMIC_EVAL_START_DLG moduleName
        [ appmodelid:ApplModelId ]
        [ hpd:HoursPerDay ] [ dpy:DaysPerYear ]
        [ runs:intValue ] [ randseed:boolValue ]
        [ start:EvalStartPeriod end:EvalEndPeriod ]
    --> RESULT ecode: intValue canceled:BoolValue
        appmodelid: intValue appmodelname: strValue hpd:realValue
        dpy:realValue runs:intValue
        randseed:BoolValue start:strValue end:strValue .

AppModelId : intValue .

HoursPerDay : realValue .

DaysPerYear : realValue .

EvalStartPeriod : strValue .

EvalEndPeriod : strValue .



RUN_DYNAMIC_EVALUATION moduleName
        appmodelid:applModelName
        [ hpd:HoursPerDay ] [ dpy:DaysPerYear ]
        [ runs:intValue ] [ randseed:boolValue ]
        [ volume-check:BoolValue [ checkhist ]
        [ start:EvalStartPeriod end:EvalEndPeriod ] .
    --> RESULT ecode:intValue .
```

AppModelId : *intValue* .

HoursPerDay : *realValue* .

DaysPerYear : *realValue* .

EvalStartPeriod : *strValue* .

EvalEndPeriod : *strValue* .

16.18.9   Commands of the "ImportExport" MessagePort

ADL_IMPORT *fileName* [**otherlib**] *ModelSettings AttrProfSettings*
        [**protfile:***fileName*] [**import-versioned-file**] [**appmodels**]
        [**silent**] [**abort-with-mixed-libs**] .
    --> **RESULT ecode:***intValue* **errtext:***strValue*
        modelids:*idCont* **attrprofids:***idCont* .

ModelSettings:
        [**existing-models:***ExistingTreatment*] *ModelSubOption* s
        [**with-mgroups**] [**target-mgroupid:***id*] .

ModelSubOption s:
        [**update-interrefs**] [**delete-recordrows**] [**delete-unchanged-objects**]
        [**delete-unchanged-connectors**] [**adopt-version-into-name**]
        [**models-prefix:***strValue*] [**models-suffix:***strValue*] .

AttrProfSettings:
        [**existing-attrprofs:***ExistingTreatment*] *AttrProfSubOption* s
        [**with-apgroups**] [**target-apgroupid:***id*] .

AttrProfSubOptions:
        [**update-attrprofrefs**] [**delete-attrprof-recordrows**]
        [**attrprofs-prefix:***strValue*] [**attrprofs-suffix:***strValue*] .

ExistingTreatment:
        overwrite | **paste** | **rename** | **ignore** | **increase-version** .

ADL_IMPORT_APPMODELS *fileName* [**target-mgroupid:***intValue*]
        [**protfile:***strValue*] [**otherlib**]
        [**import-versioned-file**] [**silent**] .
    --> **RESULT ecode:***intValue* **errtext:***strValue* **modelids:***strValue* .

(ADL import of the application models.)

ADL-EXPORT : *ADLExportByOptions* | *ADLExportByName* | *ADLExportByID* .
    --> **RESULT ecode:***intValue* **errtext:***strValue*
        modelids:*strValue* **attrprofids:***strValue* .

ADLExportByOptions:

ADL_EXPORT *fileName ModelOptions AttrprofOptions*
        [**export-30**] [**export-39**] [**no-overwrite**] [**verbose**] .

ModelOptions :   [**modelids:***strValue*] [**modelgroupids:***strValue*]

```
                     [ with-referenced-models ] [ with-models ]
                     [ with-mgroups ] [ mgroups-recursive ]
                     [ with-referenced-attrprofs ] .

AttrprofOptions :   [ attrprofids:strValue ] [ apgroupids:strValue ]
                     [ with-attrprofs ] [ with-apgroups ] [ apgroups-recursive ] .
```

ADLExportByName:

```
ADL_EXPORT fileName modelname:strValue
           [ version:strValue ] modeltype:strValue
           [ with-submodels ] [ with-referenced-models ] [ verbose ] .
```

ADLExportById:

```
ADL_EXPORT fileName modelids:strValue
           [ with-submodels ] [ with-referenced-models ] [ verbose ] .
```

fileName: The name of the created ADL file

```
ADL_EXPORT_APPMODELS fileName appmodelids:strValue
                     [ with-referenced-models ] [ export-threads ]
                     [ no-overwrite ] [ verbose ] .
    --> RESULT ecode:intValue errtext:strValue
            modelids:strValue .
```

fileName: The name of the created ADL file

```
ADL_EXPORT_APPMODELS fileName appmodelids:strValue
        [ with-referenced-models ] [ export-threads ]
        [ no-overwrite ] [ verbose ] .
    --> RESULT ecode:intValue errtext:strValue
        modelids:strValue .
```

FileName: The name of the created ADL file

```
EXEC_ADL_EXPORT_DLG .
```

```
EXEC_ADL_IMPORT_DLG .
```

```
SHOW_EXPORT_DLG [ title:strValue ] [ filename:strValue ]
        [ filedescription:strValue ] [ fileextension:strValue ]
        [ mode:dialogMode ] .
    --> RESULT endbutton:EndButton filename:strValue
        modelids:idStrValue mgroupids:idStrValue
        attrprofids:idStrValue apgroupids:idStrValue
        with-models:boolValue with-referenced-models:boolValue
        with-mgroups:boolValue mgroups-recursive:boolValue
        with-attrprofs:boolValue with-referenced-attrprofs:boolValue
```

                                   with-apgroups:*boolValue* **apgroups-recursive:***boolValue* .

dialogMode : **"adl"** │ **"xml"** .

EndButton : **"ok"** │ **"cancel"** .

SHOW_IMPORT_START_DLG [**title:***strValue*]
       [**filedescription:***strValue*] [**fileextension:***strValue*]
       [**otherlib-opt-visible:***boolValue*]
       [**existing-models-vis-items:***strValue*]
       [**auto-rename-models-opt-visible:***boolValue*]
       [**update-interrefs-opt-visible:***boolValue*]
       [**find-by-id-opt-visible:***boolValue*]
       [**delete-untouched-recrows-opt-visible:***boolValue*]
       [**existing-attrprofs-vis-items:***strValue*]
       [**auto-rename-attrprofs-opt-visible:***boolValue*]
       [**update-attrprofrefs-opt-visible:***boolValue*]
       [**filename:***strValue*] [**otherlib:***boolValue*]
       [**existing-models:***ExistingTreatment*] [**auto-rename-models:***boolValue*]
       [**models-prefix:***strValue*] [**models-suffix:***strValue*]
       [**update-interrefs:***boolValue*] [**adopt-version-into-name:***boolValue*]
       [**find-by-id:***boolValue*] [**delete-recordrows:***boolValue*]
       [**delete-unchanged-objects:***boolValue*]
       [**delete-unchanged-connectors:***boolValue*]
       [**delete-untouched-recrows:***boolValue*]
       [**existing-attrprofs:***ExistingTreatment*]
       [**auto-rename-attrprofs:***boolValue*]
       [**attrprofs-prefix:***strValue*] [**attrprofs-suffix:***strValue*]
       [**update-attrprofrefs:***boolValue*]
       [**delete-attrprof-recordrows:***boolValue*]
       [**with-protocol:***boolValue*] [**protocol-filename:***strValue*]
       [**mode:***dialogMode*] .
   --> **RESULT endbutton:***EndButton*
       filename:*strValue* **otherlib:***boolValue*
       existing-models:*ExistingTreatment* **auto-rename-models:***boolValue*
       models-prefix:*strValue* **models-suffix:***strValue*
       update-interrefs:*boolValue* **adopt-version-into-name:***boolValue*
       find-by-id:*boolValue* **delete-recordrows:***boolValue*
       delete-unchanged-objects:*boolValue*
       delete-unchanged-connectors:*boolValue*
       delete-untouched-recrows:*boolValue*
       existing-attrprofs:*ExistingTreatment* **auto-rename-attrprofs:***boolValue*
       attrprofs-prefix:*strValue* **attrprofs-suffix:***strValue*
       update-attrprofrefs:*boolValue* **delete-attrprof-recordrows:***boolValue*
       with-protocol:*boolValue* **protocol-filename:***strValue* .

EndButton : **"ok"** │ **"cancel"** .

dialogMode : **"adl"** │ **"xml"** .

ExistingTreatment :
       "overwrite" │ **"paste"** │ **"rename"** │ **"ignore"** │ **"increase-version"** .

SHOW_IMPORT_SELECT_DLG [**title:***strValue*]
       [**with-models:***boolValue*] [**with-mgroups:***boolValue*]
       [**mgroups:***MGroups*] [**import-mgroups:***boolValue*]
       [**with-attrprofs:***boolValue*] [**with-apgroups:***boolValue*]
       [**apgroups:***APGroups*] [**import-apgroups:***boolValue*]

```
              [ mode:dialogMode ] .
      --> RESULT endbutton:EndButton sel-models:SelModels
          import-mgroups:boolValue mgroupid:id
          sel-attrprofs:SelAttrProfs import-apgroups:boolValue
          apgroupid:id .
```

MGroups : { *MGroup* | *Model* } .

MGroup : **MGROUP name:***strValue* .

Model : **MODEL name:***strValue*
       version:*strValue* **type:***strValue* .

APGroups : { *APGroup* | *AttrProf* } .

APGroup : **APGROUP name:***strValue* .

AttrProf : **ATTRPROF name:***strValue*
       version:*strValue* **apclassname:***strValue* .

SelModels : { *Model* } .

Model : **MODEL path:***strValue* **name:***strValue*
       version:*strValue* **type:***strValue* .

SelAttrProfs : { *AttrProf* } .

AttrProf : **ATTRPROF path:***strValue* **name:***strValue*
       version:*strValue* **apclassname:***strValue* .

dialogMode : **"adl"** | **"xml"** .

EndButton : **"ok"** **"cancel"** .

The following commands are only available in the Administration Toolkit:

```
UDL_EXPORT file:fileName
        ( userids:strValue [ usergroupids:strValue ] |
        usergroupids:strValue [ userids:strValue ] ).
    --> RESULT ecode:intValue errtext:strValue .
```

```
UDL_IMPORT file:fileName
        [ existing-users:ExistingUsersTreatment ]
        [ existing-usergroups:ExistingUsergroupsTreatment ]
        [ existing-sysusers:ExistingSysusersTreatment ]
        [ invalid-sysusers:InvalidSysusersTreatment ]
        [ from-libraryid:intValue  to-libraryid:intValue ]
        [ silent ] .
    --> RESULT ecode:intValue errtext:strValue .
```

ExistingUsersTreatment: **overwrite** | **rename** | <u>**ignore**</u> | **actualise** .

ExistingUsergroupsTreatment: **overwrite** | <u>**adopt**</u> | **rename** | **ignore** | **actualize** .

ExistingSysusersTreatment: **overwrite** | <u>**ignore**</u> | **actualize** .

InvalidSysusersTreatment: <u>**ignore**</u> | **convert** .

16.18.10   Commands of the "Documentation" MessagePort

```
EXEC_MENUENTRY strExpr [ no-interref-warnings ] .
    --> RESULT ecode:intValue
```

The **ecode** gets the value 0, only when the documentation is generated successfully, otherwise it has the value 1.

```
EXEC_EXPORTDIALOG [ dialogTitle ]
        [ filedescription:fileDesc ] [ fileextension:fileExt ]
        [ allowedmodeltypes:ModeltypeList ] [ sortMode:SortMode ]
        [ submodels:boolValue ] [ selmodelids:idCont ] .
    --> RESULT ecode:intValue filename:strValue
        modelids:idCont selmodelids:idCont refmodelids:idCont .
```

```
ModelTypeList  "Breadth-First Search" | "Depth-First Search" | "Lexical Ordering" .
```

```
SortMode:  "Breadth-First Search" | "Depth-First Search" | "Lexical Ordering" .
```

allowedmodeltypes: model type names, which are separated with **;**. The **ecode** gets the value **0**, if the button "Delete" has been used; when the "OK" button is used, it gets the value **1**.

```
EXEC_OPTIONSDIALOG .
```

```
EXEC_ACFILTER attribute:strExpr [ modeltype:strModelType ].
```

Opens the attribute and class filter. The **attribute** is the name of the class attribute "LibraryMetaData", which contains information about the attributes/classes, which are chosen on the purpose of the documentation generation. Choosing one model type (**modeltype**) from the list of all the model types, allows the system to show, which attribute and class filters are available.

```
ACFILTER_DISABLE .
```

```
ACFILTER_ENABLE .
```

```
DOCU_EXPORT menuName [ filename:fileName ]
        [ modelids:idCont ] [ refmodelids:idCont ]
        [ noInterRefWarnings ] [ silentmode ] [ get-offline-target-instances ] .
    --> RESULT ecode:intValue gfxscaling:arrayValue .
```

ATTENTION: 1 = ok, 0 = Error!

```
NoInterRefWarnings :
        no-interref-warnings | no-interref-warnings:boolValue |
        no-interref-warnings:attribute:attrName .
```

menuName: The text from the menu entry "Documentation".

fileName: File name for the target file.

idCont: String containing the model IDs separated with blanks.

no-interref-warnings: Suppreses the error message [asgmlexp-08] (this message is shown, if the referenced target of an INTERREF attribute does not exist any longer).

silentmode: Suppresses all the error messages and support windows.

`USERSETTINGS_RESTORE_FROM_DB` .

Re-establishes the settings defined by the user for the documentation generation by using the values saved in the ADONIS database.

`USERSETTINGS_SAVE_TO_DB` .

Saves the settings defined by the user for the purpose of the documentation in the ADONIS database.

`USERSETTINGS_SET_TO_DEFAULT` .

It brings back the standard values defined in the ADONIS database and deletes the settings defined by the user.

`XML_ADD_CALLBACK` *procedureName* **name:***strValue* [**type:***strValue*] .
       --> **RESULT ecode:***intValue*

`XML_BREAK` .

`XML_CLOSE` (**read** | **write**) .

`XML_DISPLAY_ERROR` .

`XML_FIND_NODE` [**holdid:***intValue*] [**nodename:***strValue*]
     [**attribute:***strValue*]
     [**index:***intValue*] .
      --> **RESULT ecode:***intValue* **nodeid:***intValue* .

`XML_GET_ATTRIBUTE` *attributeName* [**node:***intValue*] .
       --> **RESULT ecode:***intValue* **value:***strValue* .

`XML_GET_CHILD_NODE` [**node:***intValue*] [**index:***intValue*] .
       --> **RESULT ecode:***intValue* **child:***intValue* .

`XML_GET_NAME` [**node:***intValue*] .
       --> **RESULT ecode:***intValue* **name:***strValue* .

```
XML_GET_PARENT_NODE [ node:intValue ] [ level:intValue ] .
      --> RESULT ecode:intValue parent:intValue .


XML_GET_VALUE [ node:intValue ] .
      --> RESULT ecode:intValue value:strValue .


XML_HOLD_NODE intValue [ node:intValue ] .
      --> RESULT ecode:intValue parent:intValue .


XML_OPEN read .
      --> RESULT ecode:intValue .


XML_OPEN write [ encoding:strValue ] .
      --> RESULT ecode:intValue .


XML_PARSE .
      --> RESULT ecode:intValue .


XML_RELEASE intValue .


XML_SET_SCRIPT strValue .
      --> RESULT ecode:intValue .


XML_VALIDATE .
      --> RESULT ecode:intValue .


XML_WRITE_ATTRIBUTE strValue [ value:strValue ] .
      --> RESULT ecode:intValue .


XML_WRITE_CONTENT strValue .
      --> RESULT ecode:intValue .


XML_WRITE_END_NODE strValue .
      --> RESULT ecode:intValue .


XML_WRITE_PLAIN strValue .
      --> RESULT ecode:intValue .
```

```
XML_WRITE_START_NODE strValue .
     --> RESULT ecode:intValue .
```

16.18.11   Commands of the "AQL" MessagePort

```
CHECK_AQL_EXPRESSION expr:strValue .
     --> RESULT ecode:intValue .
```

```
EVAL_AQL_EXPRESSION expr:strValue ( modelid:intValue | modelscope ) .
     --> RESULT ecode:intValue objids:strValue .
```

16.18.12   Commands of the "UserMgt" Message Port

Hint: Some of the commands of the MessagePort "UserMgt" are only available in the ADONIS Administration Toolkit, some others in both toolkits.

Commands for both ADONIS toolkits:

```
GET_USER_ID user:strValue .
     --> RESULT ecode:intValue userid:intValue .
```

```
GET_ALL_USERS .
     --> RESULT ecode:intValue users:strValue .
```

```
GET_SYSUSER_ID user:strValue domain:strValue
     logon-name-type:LNType .
     --> RESULT ecode:intValue userid:intValue
```

```
LNType : "samaccountname" | "principalname" .
```

```
GET_ALL_SYSUSER_IDS .
     --> RESULT ecode:intValue userids:strValue .
```

```
GET_ALL_USERGROUPS_OF_CURRENT USER .
     --> RESULT ecode:intValue usergroups:strValue .
```

Commands only for the ADONIS Administration Toolkit:

```
GET_ALL_USERGROUPS .
     --> RESULT ecode:intValue usergroups:strValue[;strValue(...)] .
```

```
GET_ALL_USERS_OF_USERGROUP usergroup:strValue .
     --> RESULT ecode:intValue users:strValue[;strValue(...)] .
```

```
GET_ALL_USERGROUPS_OF_USER user:userName .
     --> RESULT ecode:intValue usergroups:{strValue[;strValue]} .
```

<u>*userName*</u> : *strValue* .

GET_USERGROUP_ID usergroup:*strValue* .
    --> **RESULT ecode:***intValue* **usergroupid:***intValue* .

CREATE_USER user:*strValue* **password:***strValue*
        library:*strValue* [**usergroups:***strValue[;strValue(...)]*]
        [**infotext:***strValue*] [**bpmtk:***boolValue*] [**admtk:***boolValue*] .
    --> **RESULT ecode:***intValue* **userid:***intValue* .

CREATE_USERGROUP usergroup:*strValue* [**users:***strValue[;strValue(...)]*] .
    --> **RESULT ecode:***intValue* **usergroupid:***intValue* .

DELETE_USER user:*strValue* .
    --> **RESULT ecode:***intValue* .

DELETE_USERS users:*userNames* .
    --> **RESULT ecode:***intValue* .

<u>*userNames* :</u> *strValue[;userNames]* .

DELETE_USERGROUPS usergroups:*strValue[;strValue(...)]* .
    --> **RESULT ecode:***intValue* .

DELETE_SYSUSERS userids:*intValue[;intValue(...)]* .
    --> **RESULT ecode:***intValue* .

CHANGE_USER_SETTINGS user:*strValue*
        [**password:***strValue*] [**library:***strValue*] [**infotext:***strValue*]
        [**bpmtk:***boolValue*] [**admtk:***boolValue*] .
    --> **RESULT ecode:***intValue* **errmsg:***strValue* .

SET_USER_ACCESS_STR user:*strValue* **accessstr:***strValue* .
    --> **RESULT ecode:***intValue* .

GET_USER_ACCESS_STR user:*strValue* .
    --> **RESULT ecode:***intValue* **accessstr:***strValue* .

SET_USERGROUP_ACCESS_STR usergroup:*strValue* **accessstr:***strValue* .
    --> **RESULT ecode:***intValue* .

GET_USERGROUP_ACCESS_STR usergroup:*strValue* .
    --> **RESULT ecode:***intValue* **accessstr:***strValue* .

ADD_USERS_TO_GROUPS users:*strValue*
        usergroups:*strValue* .
    --> **RESULT ecode:***intValue* .

REMOVE_USERS_FROM_GROUPS users:*strValue[;strValue(...)]*
        usergroups:*strValue[;strValue(...)]* .
    --> **RESULT ecode:***intValue* .

```
BALANCE_SYSUSERGROUPS usergroup:strValue systemgroup:strValue
        domain:strValue library:strValue
        [ bpmtk:boolValue ] [ admtk:boolValue ]
        [ infotext:strValue ] [ delete-type:DelType ] .
    --> RESULT ecode:intValue errmsg:strValue .

DelType : "remove" | "delete" | "semi-delete" .
```

16.19 Event Handler

Event Handlers are AdoScripts which are executed when certain events occur. They are stored in the application library.

Possible events are:

ActivateModelWindow (Modelling)

> This event is used, *after* activating a model window.
>
> The ID of the affected model is referenced to the variable `modelid` (type: INTEGER).
>
> **Note:** In case of switching between windows, the first event is always the *DeactivateModelWindow* and then the *ActivateModelWindow*.

AfterAutoConnect (Modelling)

> This event is used *after* automatic creation or deletion of connectors. This event is not activated for every connector but it concerns the whole procedure.
>
> The ID of the affected model is referenced to the variable `modelid` (type: INTEGER).
>
> **Note:** In case of switching between windows, the first event is always the *DeactivateModelWindow* and then the *ActivateModelWindow*.

AfterCreateModelingConnector (Modelling)

> This event is used, *after* a connector is inserted into a model (in the model editor).
>
> The variable `modelid` stores the ID of the affected models, the `objid` the ID of the new connector, `classid` the ID of the new connector class, the `fromobjid` the ID of the outgoing objects and the `toobjid` the IDs of the target objects. The parameter `origin` can get the values: **0** (the user used a connector), **1** (the connector is created by means of the Copy-Paste or the Cut-Paste functions) or the value **2** (the connector was re-established by means of the function "Undo").

AfterCreateModelingNode (Modelling)

> This event is used, *after* entering an instance into a model (in model editor).
>
> The `modelid` stores the ID of the affected models, the `objid` the ID of the new instance and the `classid` the ID of the instance class. The parameter `origin` can get the values: **0** (the user modelled the instance), **1** (the instance is created by means of the Copy-Paste or the Cut-Paste functions) or the value **2** (the instance was re-established by means of the function "Undo").

AfterCreateModelWindow (Modelling)

> This event is used, *after* creating a new model window in a model editor.
>
> The variable `modelid` contains the ID of the affected model.

AfterDiscardModelWindow (Modelling)

> This event is used, *after* closing a model window in a model editor.
>
> The variable `modelid` contains the ID of the affected model.

AfterEditAttributeValue (Modeling)

> This event is triggered *after* an attribute value was edited in a notebook, the tabular view, or a quick-edit field in the drawing area.

The variable **instid** (type Integer) contains the ID of the modified instance, **attrid** (type Integer) contains the ID of the concerned attribute, **modelid** (type Integer) contains the ID of the model with the modified value, and **attrtypeid** (type Integer) the code of the attribute type (0=INTEGER, 1=DOUBLE, 2=STRING, 3=DISTRIBUTION, 4=TIME, 5=ENUMERATION, 6=ENUMERATIONLIST, 7=LONGSTRING, 8=PROGRAMCALL, 9 =INTERREF, 10=EXPRESSION, 11=RECORD 12=ATTRPROFREF, 13=DATE, 14=DATETIME).

Hint: This event is similar to **SetAttributeValue**. The difference is that AfterEditAttribute-Value is only triggered on attribute value changes *by the user*, while SetAttributeValue is available after *every* value modification. If both events are triggered, SetAttributeValue always comes *before* AfterEditAttributeValue.

AppExit (Application)
The application will be finished. This Event Handler has no parameters.

AppInitialized (Application)
The initialisation of the application is finished. This Event Handler has no parameters.

BeforeCreateRelationInstance (Core)
This event is used *before* a relation (connector) is created.

Simultaneously the variables **frominstid** (type: INTEGER) provides the ID of the closing instance, **toinstid** (type: INTEGER) the ID of the target instance, **relationclassid** (type: INTEGER) The ID of the relation class and **componentid** (type: INTEGER) the ID of the affected components (library, model ...). In the case of **EXIT**, it is expected that the Event Handler returns a value. If the value is not equal to **0**, the use of it will be ended. If the value is **-1** it continues without the error message, and if the value equals **-2** it continues with the error message.

BeforeCreateModelWindow (Modelling)
This event is used, *before* a new model window is created in the model editor.

The variable **modelid** contains the ID of the affected model.

BeforeDeleteAPVersions (Core)
This event is used, *before* the deletion of an attribute profile.

In such a case the **apversioids** is a container, which stores all the IDs of the attribute profiles waiting for deletion. In case of **EXIT** it is expected that the Event Handler returns a value. If the value is not equal to **0**, the use of it is cancelled. If it is **-1** it continues without the error message and if the value equals **-2** or **>0** it continues with the error message.

BeforeDeleteInstance (Core)
This event is used, *before* deletion of an instance (object, table cell or attribute profile).

The variable **instid** provides the ID of the deleted instance, the **classid** the class, which belongs to the instance and the **modelid** the ID of the affected model. In case of **EXIT** it is expected that the Event Handler returns a value. If the value is not equal to **0**, the use of it is cancelled. If it is **-1** it continues without the error message and if the value equals **-2** or **>0** it continues with the error message.

BeforeDeleteModel (Core)
This event is used, *before* a model is deleted.

The variable **modelid** contains the ID of the affected model.

BeforeDiscardInstance (Core)
This event is used, *before* an instance (object, table cell or attribute profile) is removed or released from memory.

The variable **instid** provides the ID of the instance, the **classid** the class, which belongs to the instance and **modelid** the ID of the affected model. Rarely are the variables **realinstanceid** (the Core-internal ID of the instance) and **realclassid** (the Core-internal ID of the class) used. In case of **EXIT** it is expected that the Event Handler returns a value. If the value is not equal to **0**, the use of it is cancelled. If it is **-1** it continues without the error message and if the value equals **-2** or **>0** it continues with the error message.

BeforeDiscardModel (Core)
This event is used, *before* a model is released from the memory.

The variable `modelid` contains the ID of the affected model.

BeforeDiscardModelWindow (Modelling)
This event is used, *before* a model window is closed in a model editor.

The variable `modelid` contains the ID of the affected model.<

BeforeSaveModel (Core)
This event is used, *before* a model is saved.

The variable `modelid` contains the ID of the affected model. The origin of the call is deposited in the `origin`. The possible forms are **"new"** ("Model" - "Neu"), **"saveas-new"** ("Model" - "Save as" Part 1), **"save"** ("Model" - "Save") and **"saveas-save"** ("Model" - "Save as" Part 2). In case of `EXIT` it is expected that the Event Handler returns a value. If the value is not equal to **0**, the use of it is cancelled. If it is **-1** it continues without the error message and if the value equals **-2** it continues with the error message.

Attention: Because of many exit values, it is not clear, if the model has been really saved!

ChangeComponent (Application)
This event is used, *after* the user has chosen the ADONIS component.

The variable `old` holds the number of old component, while the `new` has the newly chosen.

ChangeRelationInstanceFromEndpoint (Core)
The origin object of a connector was changed. The ID of this connector was assigned to the variable `relationinstanceid`, the ID of the existing origin object of the variable `oldfrominstanceid`, the ID of the new origin object of the variable `newfrominstanceid` and the ID of the model of the variable `componentid`.

ChangeRelationInstanceToEndpoint (Core)
The target object of the connector was changed. The ID of the connector was assigned to the variable `relationinstanceid`, the ID of the old target object of the variable `oldtoinstanceid`, the ID of the new aim object of the variable `newtoinstanceid` and the ID of the model of the variable `componentid`.

CreateApplicationModel
An application model was created. The ID of the application model was assigned to the variable `appmodid`, the name of the appliaction model of the variable `appmodname` and the value 0 or 1 of the variable `onthread` .

Note: The event happens after the applicaiton model was created.

Note: The variable `onthread` has the value **0** if the model is based on a model-related versioned application library, or if the model is based on a time-related versioned application library and version specific was defined. The variable has the value **1** if the model is based on a time-related versioned application library and version consequence specific was defined.

CreateAPThread (Core)
This event occurs, *after* an application model, specific for the version has been created.

The variable `apdirid` stores the IDs of the attribute profile catalogues, where the version sequence was created. The variable `apthreadid` gets the ID of the version sequence itself and the variable `apthreadname` the name of the new version sequence.

CreateAPVersion (Core)
This event occurs, *after* a new attribute profile version has been created.

The variable `apthreadid` contains the ID of the version sequence, to which the new version is attached, the variable `apversionstr` contains the version name saved as a string and `apversionid` the ID of the new attribute profile version.

CreateInstance (Core)
An object was created. This result occurs *after* the creation of an object.

The model ID was assigned to the variable `modelid`, a new object to the variable `instid` and the object class ID of the variable `classid`.

CreateMGroup (Core)

This event occurs *after* a creation of the model group.

The ID of this group was assigned to the variable `mgroupid`, the subgroup `supermgroupid` and to the group name of the variable `mgroupname`.

CreateModel (Core)

A model was created. The model ID of the model was assigned to the variable `modelid` .

Note: This event happens after the model has been created (also at the model import), but before the model window is visible.

CreateModelRef (Core)

This event happens *after* a model version has been assigned to a model group.

The ID of the model group was assigned to the variable `mgroupid`, which refers the version sequence to the variable `threadid` and to the ID, which refers the model group of the version sequence to the variable `modelrefid`.

CreateRelationInstance (Core)

A connector was created. The event occurs *after* the connector has been created.

The connector ID was assigned to the variables `relationinstanceid`, the class ID of the connector to the variable `relationclassid`, the ID of the outcoming objects to the variable `frominstanceid`, the ID of the target objects to the variable `toinstanceid` and the model ID to the variable `componentid`.

DeactivateModelWindow (Modelling)

This event happens, *after* a model window has been deactivated.

The variable `modelid` (type: INTEGER) is assigned to the deactivated model.

Note: It is not permitted to open a dialogue window within such Event Handlers, as it can not be used by the operating system.

DeleteApplicationModel (Core)

An application model was deleted. This event happens, *after* the application model has been deleted. It is no longer possible to attach to the model content.

The application model ID is assigned to the variable `appmodelid (appmodid)`.

DeleteAPVersion (Core)

This event happens, *after* the attribute profile version has been deleted.

The variable `apversionid` contains the ID, `apclassid` the class ID and `name` the name of the attribute profile.

DeleteInstance (Core)

An instance (object, table row or attribute profile) was deleted. This event happens *after* the deletion.

The ID of the deleted instance is assigned to the variable `instid`, the ID of the class of the instance to the variable `classid`, and the name of the affected model to the variable `modelid`.

DeleteMGroup (Core)

This event happens *after* the deletion of a model group.

The ID of the deleted group is assigned to the variable `mgroupid`.

DeleteModel (Core)

This event happens, *after* deletion of a model. It is no longer possible to attach content to the model.

The model ID is assigned to the variable `modelid`.

DeleteModelRef (Core)

This event happens, *after* a version sequence of a model has been removed from a model group.

The variable `mgroupid` contains the ID of the model group, the `threadid` the ID of the version sequence and the `modelrefid` the ID of the reference of the model group to a version sequence.

DeleteModelThread (Core)
A model order was deleted. This event happens, *after* deletion. It is no longer possible to attach content to the model.

The model order ID is assigned to the variable `threadid`.

DeleteRelationInstance (Core)
A connector was deleted. This event happens *after* deletion.

The ID of the connectors are assigned to the variable `relationinstanceid`, the class ID of the connector to the variable `relationclassid`, the ID of the outgoing object to the variable `from-instanceid`, the ID of the target object to the variable `toinstanceid` and the model ID to the variable `componentid`.

DiscardModel (Core)
A model was closed. This event happens, *after* closure. It is no longer possible to attach to the model content.

The ID of the closed model is assigned to the variable `modelid`.

DiscardRelationInstance (Core)
A relation instance was discarded fom memory and freed. This event is triggered *after* the discarding. The relation is no longer accessible.

The ID of the relation instance is assigned to the variable `relninstid` and the ID of the component concerned (model, library ...) to the variable `componentid`.

EndADLImport (Import/Export)
The ADL import was finished. This event happens *after* the end.

The value **1** is assigned to the variable `successful`, when the import was completed successfully, otherwise it gets the value **0**.

EndUpdateAttrProfs (Core)
The attribute profile selection list was actualised. This event happens *after* an update. This Event Handler has no parameter.

EndUpdateModels (Core)
The model selection list was updated. This event happens *after* an update. This Event Handler has no parameter.

MoveMGroup (Core)
A model group has changed its place within the model group structure. This event happens *after* it.

The ID of the moved group is assigned to the `mgroupid`, the old name to the main group `oldsupermgroupid`, and the new name to the new main group `newsupermgroupid`.

MoveModelRef (Core)
The event happens, *after* moving the version sequence of a model from one model group to another.

The ID of the moved model reference is assigned to the variable `modelrefid`, which is the new model model group `newmgroupid`.

OpenModel (Core)
A model was opened. This event happens, *after* the model has been opened, but *before* the model window is created.

The model ID of the model is assigned to the variable `modelid`.

PrintModel (Modelling)
This event happens, *before* printing a model.

The ID of the affected model is assigned to the variable `modelid`.

RenameApplicationModel (Core)

An application model was renamed. This event happens *after* changing the name.

The application model ID is assigned to the variable `appmodelid`, the old name to the variable `oldname`, and the new name to the variable `newname`.

RenameAttrProf (Core)

An attribute profile was renamed. This event happens *after* changing the name.

The attribute profile(thread)s ID is assigned to the variable `apthreadid`, the old name to the variable `oldname`, and the new name to the variable `newname`.

RenameInstance (Core)

An object was renamed. This event happens *after* changing the name.

The object ID is assigned to the variable `instid`, the old name to the variable `oldname` and the new name to the variable `newname`.

RenameLibrary (Core)

This event happens *after* changing a name of an application library (possible only in the Administration Toolkit).

The ID of the renamed application library is assigned to the variable `libid`, the old name to the variable `oldname` and the new name to the variable `newname`.

RenameMGroup (Core)

This event happens *after* changing a name of a model group.

The ID of the renamed group is assigned to the variable `mgroupid`, the new name `mgroupname`.

RenameModelThread (Core)

This event happens *after* renaming a model or a model thread.

The ID of the model thread is assigned to the variable `modelthreadid` (type: INTEGER), the old name to the variable `oldname` (type: STRING), and the new name to the variable `newname` (type: STRING). The boolean variable **external** states whether the renaming was performed by the currently active user (**0**) or via refreshing the model list (**1**, another user renamed).

Note: This event *does not* happen, if the basic name in the time and version-related application library is changed.

SaveLibrary (Core)

A library was saved. This event happens *after* saving.

The library ID is assigned to the variable `libid`.

Note: This event also happens when an attribute profile was changed!

SaveModel (Core)

A model was saved. This event happens *after* the saving action.

The ID of the model is asigned to the variable `modelid`. The origin of the call is deposited in the variable `origin`. Possible forms are **"new"** ("Model" - "New"), **"saveas-new"** ("Model" - "Save as" Part 1), **"save"** ("Model" - "Save") and **"saveas-save"** ("Model" - "Save as" Part 2).

Attention: In case there is Auto save activated, a pre-sight is advisable. Many changes in the EventHandler can cause recursion!

SetAttributeValue (Core)

An attribute value was changed. In this case, the changed attribute can mean a model attribute, an object attribute, a connector attribute, a table attribute or an attribute profile attribute.

The ID of the affected instance is assigned to the variable `instid`, the ID of the changed attribute to the variable `attrid`, the name of the model to the variable `modelid` and the type of the changed attribute to the variable `attrtypeid` (0=INTEGER, 1=DOUBLE, 2=STRING, 3=DISTRIBUTION, 4=TIME, 5=ENUMERATION, 6=ENUMERATIONLIST, 7=CORE_LONGSTRING, 8=PROGRAMCALL, 9=INTERREF, 10=EXPRESSION, 11=RECORD, 12=ATTRPROFREF, 13=DATE, 14=DATETIME).

SetModelVersion (Core)

This event happens, *after* the change of a model version number.

The version ID of a model is assigned to the variable **modelversionid**(type: INTEGER), the old versioning to the variable **oldname** (type: STRING) and the new versioning to the variable **new-name** (type: STRING).

Hint: This event happens only by the time-related versioning.

ShowSim1PathResult (Simulation)

Triggered, if the button "Path result" in the result window of the path analysis is clicked. The following variables are set:

simpath (type STRING): Contains the IDs of all models as well as the IDs of all objects and connectors in the current path. The ID string starts with the ID of the main process model followed by the ID of the objects and connectors in the main process model. Concerning sub process models, in addition to the model ID of the submodel also the model ID of the called process models are included; after that the ID of the objects and connectors in the sub process model follow.

pathprob (type DOUBLE): Probability of the current path.

pathcycletime (type DOUBLE): Execution time of the current path in seconds.

simruns (type INTEGER): Number of simulation runs.

The variable **return** (type INTEGER): must be set by the EventHandler. If the value equals **0** (default), at the end of the AdoScript the standard result window will be shown. The value **1** supresses this behaviour.

SimulationEnded (Simulation)

The path analysis was finished and the result window closed. The Event-Handler has no parameters.

StartADLImport (Import/Export)

This event happens, *before* the start of the ADL import. The Event-Handler has no parameters.

StartUpdateAttrProfs (Core)

The attribute profile selection list was updated. This event happens, *before* the start of the update. The Event-Handler has no parameters.

StartUpdateModels (Core)

The model selection list was updated. This event happens, *before* the start of the update. The Event-Handler has no parameters.

UpdateActions (Modelling)

This event happens, when a menu is updated, e.g. while opening/activating/closing a model window or while saving.

The model ID in the active window is assigned to the variable **modelid** (type: INTEGER). If there is no active window, the modelid has the value **-1**.

17. LEO

LEO is a META language, which can be defined as a special form of concrete languages. This concrete language is used for the display of complex data structure values in a readable form. Furthermore, LEO offers also the possibility to use the (arithmetical) expressions. Moreover, this concrete language can contain Scripting elements.

17.1 LEO Syntax

The syntax of LEO is based on the following three main principles:

- A LEOgramm is a sequence of *elements*. Every element starts with a key word in capital letters. Everything that is executed until the next keyword, belongs to this element.

- Elements have *attributes*, of which names starts with lower-case letters. Every attribute will be assigned to a value or at least an implicit default value. For attributes there are different data types. The order in which the attribute belongs within an element is quoted is not important.

- A value can be a literal or one of the LEO data types, this means a figure or a string or a calculated expression. In some cases an element or an attribute can be assigned to a LEOgramm.

At assignments of attributes the attribute name and the assigned value will be quoted together. The data type of an attribute can be text, figure or measure of length. A speciality are the *expressions*, which enable the calculation of values during the execution time. Furthermore you can also set *modificators* at an attribute which are attributes of the attributes.

<u>Examples for attribute assignments:</u>

```
firma:"BOC"
plz:1010
pos:right:20.4cm
color:blue
write-protected
e:(m * c * c)
```

The first attribute is of type text, the second of type figure. Concerning the third attribute it is a measure of length in cm as well as the modificator **right**, quoted.

The attribute **colour** will not be assigned to a value, but the modificator **blue** is set. Therefore modificators can also have an enumeration character.

Another feature is the fifth attribute **write-protected**, where its meaning depends on whether it is quoted or not. This way, in case of concrete languages it is possible to assign true values. The end builds here an attribute with an expression (see chap. 17.3, p. 674). These expressions are always in brackets and access the enviroment variables. The existing possibilities depend on the concrete language.

During the evaluation order, it is decided in the attributes of the evaluated program, if the order of the given attributes is of importance. Often, the attributes have default values, which are assigned to the attributes when they have not been filled in. If an undefined attribute is quoted, it will be ignored - as it could be a future extension or a deleted attribute. There can be many blanks and breaks between the attribute assignments.

A LEO element can also have a nameless attribute, where its value can be taken as the value of the element. If it is quoted, it stands directly behind the element name. Examples:

```
LEVEL 42
FRUIT "Gala" country:"Styria"
```

Between elements and between attributes there could be as many blanks and breaks as the user likes. A value of the data type text will be surrounded by the quotation marks and can contain as many characters as the user likes, but masking using the (/) is sometimes necessary. Numerical values must be quoted with the measure unit. Here the units m, cm, mm and pt are available, and should be used. As separation marks at flaoting-point numbers a decimal point has to be set. This is not necessary for values without a comma - therefore the user can e.g. write 1 instead of 1.0.

The LEO Grammar (see chap. 17.2, p. 672) is explained in the following chapter.

The comments will be introduced in LEO with a double cross (**#**). The text after the double cross will not be considered until the end of the line. For the following example only the **ATTR "Name"** remains:

```
# An example
ATTR "Name" # Name of the object
# ATTR "Description"
```

17.2   LEO Grammar

<u>**LEO-Program**</u> **:**	{ *Element* } .
***Element* :**	*Keyword* [*SimpleValueOrExpr*] { *Attribute* } [*Body*] .
***Keyword* :**	*uc_identifier* .
***Attribute* :**	*AttrName* [**:** *Modifier*] [**:** *SimpleValOrExprOrBody*] .
***AttrName* :**	*attr_identifier* | *ArrayLValue* .
***ArrayLValue* :**	*lc_identifier* **[** *CommaExpr* **]** .

      ATTENTION: **[ ]** ***ArrayLValue*** are terminal symbols!

***Modifier* :**	*attr_identifier* .
***SimpleValOrExprOrBody* :**	*SimpleValOrExpr* | *Body* .
***SimpleValOrExpr* :**	*SimpleVal* | **(** *CommaExpression* **)** .
***CommaExpression* :**	*OrExpr* | *CommaExpr* **,** *OrExpr* .
***OrExpr* :**	*AndExpr* | *OrExpr* **OR** *AndExpr* .
***AndExpr* :**	*EqualityExpr* | *AndExpr* **AND** *EqualityExpr* .
***EqualityExpr* :**	*AddExpr* [*EqualityOp* *AddExpr*] .
***EqualityOp* :**	**=** | **<** | **>** | **<>** | **!=** | **<=** | **>=** .
***AddExpr* :**	*MultExpr* | *AddExpr* **+** *MultExpr* | *AddExpr* **-** *MultExpr* .
***MultExpr* :**	*UnaryExpr* | *MultExpr* ***** *UnaryExpr* | *MultExpr* **/** *UnaryExpr* | *MultExpr* **SUB** *UnaryExpr* .
***UnaryExpr* :**	*PostfixExpr* | *UnaryOp* *UnaryExpr* .
***UnaryOp* :**	**-** | **LEN** | **NOT** | **STR** | **VAL** | **CM** | **CMS** | **PT** | **PTS** | **ROUND** | **FLOOR** | **CEIL** | **INT** | **ASC** | **CHR** .
***PostfixExpr* :**	*PrimaryExpr* | *ArraySubscription* | *FunctionCall* .
***ArraySubscription* :**	*PostfixExpr* **[** *CommaExpr* **]** .

ATTENTION:	**[ ]** at *ArraySubscription* are terminal symbols!
FunctionCall :	`lc_Identifier ( ExpressionSeq ) .`
PrimaryExpr :	`Value	expr_identifier	(CommaExpr) .`
ExpressionSeq :	`[ Expression { , Expression } ] .`
Expression :	`OrExpr .`
Value :	`SimpleValue	ArrayValue .`
SimpleValue :	`NumValue	MeasureValue	TimeValue	StringValue .`
MeasureValue :	`NumValue MeasureUnit .`
TimeValue :	`LeoTimeValue	AdoTimeValue .`
LeoTimeValue :	`NumValue	TimeUnit .`
NumValue :	`DecValue	HexValue .`
DecValue :	`DecValue1	DecValue2 .`
DecValue1 :	`[ - ] intValue [ . digit { digit } ] [ Exponent ] [ % ] .`
DecValue2 :	`[ - ] . digit { digit } [ Exponent ] [ % ] .`
Exponent :	`e [ - ] digit { digit } .`
HexValue :	`[ - ] $ { hexdigit } .`
MeasureUnit :	`m	cm	mm	pt .`
TimeUnit :	`s .`
StringValue :	`"{ letter }" .`
AdoTimeValue :	`years : days : hours : mins : secs .`
years :	`intValue .`
days :	`000	...	365 .`
hours :	`00	...	24 .`
mins, secs :	`00	...	59 .`
IntValue :	`digit{ digit } .`
ArrayValue :	`{ CommaExpr } .`
ATTENTION:	**{ }** at *ArrayValue* are terminal symbols!
expr_identifier :	`lc_identifier	qm_identifier .`
lc_identifier :	`lc_letter { lc_letter	uc_letter	digit } .`
qm_identifier :	`? attr_identifier .`
attr_identifier :	`lc_letter { lc_letter	uc_letter	digit	- } .`
uc_identifier :	`uc_letter { uc_letter	_ } .`
uc_letter :	`A	...	Z .`
lc_letter :	`a	...	z .`
hexdigit :	`digit	A	...	F	a	...	f .`

***digit* :**	0 | ... | 9 .
***Body* :**	{ *LeoProgram* } .

> **ATTENTION:** **{ }** *Body* are terminal symbols!

> **Hint:** Identifiers and numerical values must always be set (ie. Not NULL). Inverted commas and backslashes must be masked with "\": this means instead of " you have \" and \ will be as \\. There are two types of comments in LEO: the comment lines starts with # and contains the rest of the line, the longer comment starts with "(*" and ends with "*)".

17.3   Use of Expressions in LEO

In most cases values, which can be assigned to a specific attribute of a LEO element, are limited to a fixed data type. Besides the datatypes integer value, floating-point value, measure of length, time and string there is a datatype named **expression** in LEO.

An expression consists of values (numerical values, measure values, times, strings), variables and operators, or function calls. If no variables are contained it is a **constant expression**. As the result of a constant expression is always the same, a constant expression can be interpreted as a value. Thus a constant expression can be quoted everywhere in the LEOgram where a constant value is expected. This will be interpreted as if the result had been quoted directly.

Example:

Consider the following LEO elements:

```
EXAMPLE "abcabc" x:5cm
EXAMPLE (2 * "abc") x:(2cm + 3cm)
```

In the first element only constant values appear and in the second only contant expressions appear. Independent of the expression of the concrete language both elements are equivalent.

Variable expressions are used only in concrete languages for which they are intended - this means where there is a value of type Expression quoted. In languages where variable expressions appear, elements with which operation variables can be set. These operation variables are only visible in the program flow itself and can be used in all variable expressions. On positions where a variable expression can be quoted, a constant is allowed to be quoted.

Example:

The **SET** is a LEO element which allows for the creation of an enviroment variable, in the form of

```
SET { VariableName:value }
```

.

If the existing language of the previous example uses, on the required positions, variable expressions the following LEOgramm fragment is equivalent to both above executed elements:

```
SET text:"abc" y:10cm
EXAMPLE (2 * "abc") x:(y / 2)
```

If the attribute value is quoted as an expression, it has to be written in brackets. Within an expression blanks and breaks can be used.

When and how often the evaluation of the expressions is made, depends on where the concrete language is used. Without expressions the LEOgramm represents a value of a determined data structure. A LEOgramm with expressions represents an amount of values of a determined data structure. The evaluation of the expressions leads to an element of this amount.

17.3.1   Logical Operators

Logical operators deliver 0 ("false") or 1 ("true"). Concerning the operators null values are interpreted as "false" and all other values are "true".

a **OR** *b*   Logic or-connection. Delivers 1, if an operator is not 0 otherwise 0.

a **AND** *b*   Logic and-connection. Delivers 1, if both operators are not 0, otherwise 0.

NOT *a*   Logic negation. Delivers 1, if the operator is 0, otherwise 0.

17.3.2   Comparison Operators

When evaluating a comparison operator, the value of the left operator must be of the same type as the value of the right operator. The result is 0 ("false") or 1 ("true").

a < *b*   small operator.

a <= *b*   small-same-operator.

a = *b*   is-operator.

a <> *b*   not-operator.

a >= *b*   larger-than-operator.

a > *b*   larger-Operator.

17.3.3   Arithmetical Operators

For arithmetical operators the operators and the result are of type figure or measure.

a + *b*   Addition. *a* and *b* must be of type figure or both of type measure.

a - *b*   Subtraction. *a* and *b* must be an evaluation both of type figure or both of type measure.

a * *b*   Multiplication. At the evaluation for the operators the following type combinations are available:

Figure * figure delivers a figure,

figure * measure delivers measure,

measure * figures delivers measure,

figure * time delivers time and

time * figure delivers time.

a / *b*   Division. It is allowed to use the following operators in the combinations:

Figure / figure delivers figure,

measure / figure delivers measure,

time / figure delivers time,

measure / measure delivers figure,

time / time delivers figure.

- *a*   converse sign.

ROUND *x* Commercial truncation of the decimal place (from 5 it is rounded down).

FLOOR *x* Truncation (to the next number).

CEIL *x*   Truncation (to the next number).

17.3.4   Arithmetical Functions

abs (*x*)   Absolute amount.

max (*x*, *y*)   Maximum.

min (*x*, *y*)   Minimum.

pow (*x*, *y*)   Power (*x* high *y*).

sqrt (*x*)   square root.

exp (*x*)   Exponential function (*e high x*).

log (*x*)   natural logarithms.

log10 (*x*)   Logarithms on basis 10.

sin (*x*)   Sine.

cos (*x*)   Cosine.

tan (*x*)   Tangent.

asin (*x*)   Arcus sinus.

acos (*x*)   Arcus cosinus.

atan (*x*)   Arcus tangens.

sinh (*x*)   Sine hyperbolicus.

cosh (*x*)   Cosine hyperbolicus.

tanh (*x*)   Tangent hyperbolicus.

random ( )   coincidence figure >= 0 und < 1.

round (*x*)   rounding. The next integer will be delivered where as from including 0.5 as commas .

floor (*x*)   off rounding. If *x* has decimal places the next smaller figure will be returned otherwise *x*.

ceil (*a*)   up rounding. If *x* has decimal places the next larger figure will be returned otherwise *x*.

17.3.5   String Operators

_s_ + _t_   one following one *s* and *t*. Both operators are from the type string the result is also a string.

_n_ * _s_   *n*-one following one*s*. An operator must be from the type figure and the other from the type string. You could also write*s* * *n* . The result is from the type string.

_s_ / _t_   Number of appearing *t* in *s*. Both operators have to be from the type string, the result has to be from the character figure.

_s_ SUB _i_   Subscription of the string. s must be from the type string, _i_ from the type figure. The identification of a character starts with 0. The result is a string, which is the _i_-character of _s_.

LEN _s_   Length of a string. The operator _s_ is of the type string, and the result (the number of characters of _s_) is of type figure.

17.3.6   String Functions

search (_source_, _pattern_, _start_)
>   _source_ and _pattern_ are of type string, _start_ from type figure. The result is of type figure. It will be searched in _source_ after _pattern_, but the _start_ has a specified position. The identification starts with the character 0. If the search was successful the position will be shown, if not it will return -1.

bsearch (_source_, _pattern_, _start_)
>   _source_ and _pattern_ are of type string, _start_ from type figure. The result is of type figure. It will be searched backwards in _source_ after _pattern_, _start_ is the specific position in which it will be started. The identification of a character starts with 0, -1 is the end of the text. If the search was successful the position will be shown, if not it will return -1.

copy (_source_, _from_, _count_)
>   _source_ of type string, _from_ and _count_ are of type figure. The result is of type figure. The part-text which will be delivered back by _source_, is the starting position starting at _from_ and continuing for the length of _count_. For _count_ the value -1 will be displayed, the part-text from _source_ starting from _from_ to the end of the string.

replace (_source_ , _pattern_ , _new_)
>   _source_, _pattern_ and _new_ are of type string. The result is _source_, where the first appearance of _pattern_ via _new_ is set. If the _pattern_ is not in the _source_, the unchanged value of the _source_ is provided.

replall (_source_, _pattern_, _new_)
>   _source_, _pattern_ and _new_ are of type string. The result is of type string. The text displayed is created as a result of replacing (in the _source_) all _patterns_ with _news_.

lower (_source_)
>   The string will be returned, but all capital letters in _source_ will bhe replaced with the appropriate lower-case letters.

upper (_source_)
>   The string will be returned, but all small letters in _source_ will be replaced with the appropriate capital letters.

mstr (_source_)
>   The returned string will ibe passed as a parameter in inverted commas returned with required masks.

regex (_regExpr_ , _source_)
>   The comparison of _source_ with the regular expression _regExpr_. It corresponds to the regular expression _source_, it gets the value 1 or 0.

tokcnt (_source_, _separator_)
>   The number of tokens in _source_ separaetd by _separator_ (a single character), will be returned.

token (_source_, _index_, _separator_)
>   The token in _source_ with the index _index_ will be returned, but with the beginning of the token starting at 0. _separator_ is a single character which separates the tokens.
>   **Hint:** When changing the string into small or capital letters (**lower** or **upper**) only the letters
>   from a to z, or A to Z will be considered. Special characters will not be converted.

tokindex (_source_ , _token_ [, _separator_])
>   The result is the index of _token_ in _source_ (starting with 0). A result of -1 means that _token_ is not a token in _source_.

Example: **tokindex("this@is@a@simple@test", "simple", "@")** = 3

tokcat (*source1 , source2* [, *separator*])
>   From the strings *source1* and *source2* a new string is generated, which contains all the tokens from *source1* and *source2*.

tokunion (*source1 , source2* [, *separator*])
>   The set union of the second string. The result contains all tokens which appear in *source1* **and/or** in *source2*.
>   **Hint:** Different to **tokcat**, values existing more than once are only given back once.

tokisect (*source1 , source2* [, *separator*])
>   The intersection of the second string. The result contains all tokens, which appear **both** in *source1* **and** in *source2*.

tokdiff (*source1 , source2* [, *separator*])
>   The difference between two strings. The result contains all tokens from *source1*, which **do not** occur in *source2*.

tokstr (*a* [, *separator*])
>   Changing an array into a token string. *a* must be of type array, *separator* must be of type string with a length of 1 (otherwise the separator is a blank). The result is a text string containing the sequence of array elements which are seperated in the same order as before, separated by *separator*.

>   *Example:* **tokstr({4711, 4712, 4713})** -> **"4711 4712 4713"**.

>   *Example:* **tokstr({"first", 1.23}, ",")** -> **"first,1.23"**.
>   **Hint:** The opposite of **tokstr()** is **strarray()**: strarray(tokstr(a)) = a:

valtokstr (*a* [, *separator*])
>   Conversion of an array into a LEO text token string. *a* must be of type array, *separator* must be of type string with a length of 1 (otherwise the separator is a blank). valtokstr() should be used instead of **tokstr()** if the tokens contain the separator character, or belong to the different types. The result is a string, where all array elements in the same order are separated with *separator*.

>   *Example:* **valtokstr({"first", 1.23,","}, ",")** -> **"\ "first\",1.23,\",\""**.
>   **Hint:** The opposite of **valtokstr()** is **valarray()**: valarray(valtokstr(a)) = a:

strarray (*str* [, *separator*])
>   Conversion of a token string into an array. *str* and *separator* must be of type string (*separator* with a length of 1, otherwise a blank is the separator). The result is an array, where all elements are of type string, containing the tokens in the same sequence as before.

>   *Example:* **strarray("begin 4711 end")** -> **{"begin", "4711", "end"}**.
>   **Hint:** The opposite of **strarray()** is **tokstr()**: tokstr(strarray(s)) = s:

valarray (*str* [, *separator*])
>   Conversion of a token string into an array. *str* and *separator* must be of type string (*separator* with a length of 1, otherwise a blank is used as the separator). This function must be used instead of **strarray( )** if the tokens are LEO values (strings in apostrophes). For the result, each token is interpreted as a value and all the values are united in one array. Thus, values in an array can be of different types.

>   *Example:* **valarray("4711 4712 4713")** -> **{4711, 4712, 4713}**.
>   **Hint:** The opposite of **valarray()** is **valtokstr()**: valtokstr(valarray(s)) = s:

>   **Hint:** During the conversion of a string into small or capital letters (**lower** or **upper**), only the letters from a to z, or from A to Z are allowed. Special characters are not converted.

>   **Hint:** For **tokcnt**, **token**, **tokcat**, **tocunion**, **tokiset**, and **tokdiff**, *separator* is a single character separating the tokens (default: blank).

17.3.7   Array Operators

a **SUB** *i* | *a* **[** *i* **]**   Subscription for an array. *a* must be of type array, *i* of the type INTEGER. The result is the *i*-th element of the array *a*. The indexing of the array elements start with *0*.

> Array elements - as well as the results - can be of different types. The data type of the two elements from one array do not have to be the same. Use **type()** if you are not sure about the element type. For the multidimensional arrays the dimensions can be identified separately. There are two possibilities: **a [i,j]** or **a [i] [j]**.

Hint: sub is also the subscription operator for strings.

LEN *a*   The length of an array. The parameter *a* is of the type array; the result provides the basis of the *a* number of elements.

Hint: len is also the length operator for strings.

17.3.8   Array Functions

array (*n1* , ... , *nm*) m>=1   Creates an array with *m* dimensions and with *n* elements per dimension. As long as nothing is assigned to an array element, it is of type "undefined".

areplace (*a* , *i1* , ... , *in* , *val*) n>=1   Replaces an array element. *a* must be a name of an array variable; the array must have at least *n* dimensions; all the *i* values must be of the type integer, but *val* can contain all the types. The return value is *val*.

aappend (*a* , *i1* , ... , *in* , *val*) n>=0   Extends an array with an element. *a* is a name of an array variable. The array must have at least n+1 dimensions. *val* can contain all the types. The return value is *val*.

17.3.9   Conversion Operators

Conversion operators change a quoted value into a value of another type.

STR *val*   Changes a value into a string. Please see the LEO garmmar for the definition of the value.

VAL *str*   Changes a string into a numeric value, a measure or a time value.

CMS *measureVal*   Changes a measured value into a numerical value in centimetres.

> *Example:* CMS 2.3cm = 2.3.

PTS *measureVal*   Changes a measured value into a numerical value in points.

> *Example:* PTS 10pt = 10.

CM *realVal*   Changes a numerical value in centimeters into a measured value.

> *Example:* CM 1.23 = 1.23cm.

PT *realVal*   Changes a numeric value in points into a measured value.

> *Example:* PT 12 = 12pt.

uistr *(val, digits)*   Changes a numerical value into a string, taking into consideration the standards of the operating system used (e.g. decimal comma in German, decimal point in English operating systems). *digits* are used for the number of desired decimal places.

Hint: In LEO, floating-point numbers have to be written with a decimal point, not a comma.

uival *(str)*   Changes a string into a numeric value taking into consideration the standards of the relevant operating system.

CHR *intVal*   Returns the character with the given numerical value. The result depends on the codepage used.

> *Example:* CHR 65 = "A".

ASC *str* Delivers the code figure for the characters quoted in *str* (e.g. ASC "A" = 65). If the quoted string is longer than one character, the value of the first character will be displayed. Note that differences can appear here according to the language and the codepage of the language.

Example: ASC "A" = 65.

INT *realVal* Truncates the comma part of a figure (without rounding, e.g. INT 23,999999 = 23).

ATTENTION: In the case of very large real values, an overflow may occur, which generates a wrong result!

REAL *numVal* Creates a floating-point number from a numerical value (optionally integer or floating-point numbers can be chosen). It is useful with performing calculations with integers, or when the starting type is unclear and the floating-point number must be used.

base64encode (*source***)** Converts the normal text *source* into base64-encoded text.

base64decode (*source***)** Converts the base64-encoded text *source* into normal text.

17.3.10   RGB Colour Values Functions

rgbval (*colorname***)** 24-bit RGB colour coding with the given colour names (see chap. 17.3.10.1, p. 681). The colour name can get the right name (like **cornflowerblue**), or a 24-Bit-RGB-value (like **$6495ed** or **16777140**). The result is of the type integer.

rgbval (*colourname***, *factor*)** 24-bit RGB colour coding with the given colour names (see chap. 17.3.10.1, p. 681) and additionally a factor for elucidation or dimout. A *factor* of 1.0 refers to the original colour, the lower values are used for darker shades and the higher for brighter shades.

rgbval (*r***, *g*, *b*)**

r, g, b: [0,255] Creates an integer value from the three components according to the formula $2 \wedge 16 * r + 2 \wedge 8 * g + b$.

rgb2hsv (*rgb***)** Converts a 3-byte RGB value into a 3-byte HSV value. The result is of the type integer.

rgb2hsv (*r***, *g*, *b*)**

r, g, b: [0,255] Converts the three given RGB values into the 3-byte HSV value. The result is of the type integer.

hsvval (*colourname***)** 24-bit HSV colour coding with the given colour name (see chap. 17.3.10.1, p. 681). The colour name can be the plain name (like **cornflowerblue**), or a 24-bit RGB value (like **$6495ed**). The result is of the type integer.

hsvval (*h***, *s*, *v*)**

h:[0,360], s:[0,1], v: [0,1] Converts the three given HSV values into the 3-byte HSV value. The result is of the type integer.

hsv2rgb (*hsv***)** Converts the 3-byte HSV value into the 3-byte RGB value. The result is of the type integer.

hsv2rgb (*h***, *s*, *v*)**

h: [0,360], s: [0,1], v: [0,1] Converts the HSV components into the 3-byte RGB value. The result is of the type integer.

byte (*val***, *n*)** Extracts the value of the byte with the index *n* from the composed colour values. The lowest byte gets the value 0 (blue in RGB, value/brightness by HSV). The result is of type integer.

Example: From a RGB value, **byte(rgb,2)** retuns the red component [0,255], **byte(rgb,1)** the green component [0,255] and **byte(rgb,0)** the blue component [0,255].

Example: From a HSV value, **byte(hsv,2) / 255.0 * 360.0** returns the colour component ("hue") [0,360], **byte(hsv,1) / 255.0** the saturation component ("saturation") [0,1] and **byte(hsv,0) / 255.0** the brightness ("value") [0,1].

17.3.10.1   LEO Colours

The following colours are defined in LEO:

```
aliceblue | antiquewhite | aqua | aquamarine | azure | beige | bisque |
black | blanchedalmond | blue | blueviolet | brown | burlywood |
cadetblue | chartreuse | chocolate | coral | cornflowerblue | cornsilk |
crimson | cyan | darkblue | darkcyan | darkgoldenrod | darkgray |
darkgreen | darkkhaki | darkmagenta | darkolivegreen | darkorange |
darkorchid | darkred | darksalmon | darkseagreen | darkslateblue |
darkslategray | darkturquoise | darkviolet | deeppink | deepskyblue |
dimgray | dodgerblue | firebrick | floralwhite | forestgreen | fuchsia |
gainsboro | ghostwhite | gold | goldenrod | gray | green | greenyellow |
honeydew | hotpink | indianred | indigo | ivory | khaki | lavender |
lavenderblush | lawngreen | lemonchiffon | lightblue | lightcoral |
lightcyan | lightgoldenrodyellow | lightgreen | lightgray | lightmagenta |
lightpink | lightsalmon | lightseagreen | lightskyblue | lightslategray |
lightsteelblue | lightyellow | lime | limegreen | linen | magenta |
maroon | mediumaquamarine | mediumblue | mediumorchid | mediumpurple |
mediumseagreen | mediumslateblue | mediumspringgreen | mediumturquoise |
mediumvioletred | midnightblue | mintcream | mistyrose | moccasin |
navajowhite | navy | oldlace | olive | olivedrab | orange | orangered |
orchid | palegoldenrod | palegreen | paleturquoise | palevioletred |
papayawhip | peachpuff | peru | pink | plum | powderblue | purple | red |
rosybrown | royalblue | saddlebrown | salmon | sandybrown | seagreen |
seashell | sienna | silver | skyblue | slateblue | slategray | snow |
springgreen | steelblue | tan | teal | thistle | tomato | turquoise |
violet | wheat | white | whitesmoke | yellow | yellowgreen
```

Figure 400: Leo colour names

17.3.11   Comma Operators

All expressions of the expressions' sequence (the sequence should be separated through commas) are evaluated and the result of the last expression is returned. These expression sequences are used when a more complicated task is required and multiple expressions need to be used to accomplish it, each building on the last expressions changes (e.g. to while (see chap. 17.3.14, p. 683)). Should the comma operator in the parameter list of a function call be used, the comma operator contained in the expression must be written in round brackets. Otherwise the comma would be interpreted as a parameter separator.

Example:

```
fcall (ex1, (ex2, ex3))
```

leads to a call of

```
fcall (ex1, ex3).
```

17.3.12   Assignment Function (set)

Syntax:

set (*lvalue* **,** *expr* **)**

Assigns a constant or the result of an expression to a variable or an array element.

17.3.13   Condition Function (cond)

<u>**Syntax:**</u>

cond (*cond* 1 **,** *expr* 1 **, ... ,** *cond n* **,** *expr n* **,** *expr n*+1 **)**
      n > 0

The parameter number of the **cond** call has to be an uneven figure >= 3. The shortest form (*n* = 1) is also **cond (** *cond* 1**,** *expr* 1**,** *expr* 2 **)**.

The parameter *cond i* quotes the conditions which will be examined one after another. As soon as one condition evaluation is "true", the expression of the condition will be evaluated and the result will be re-delivered. Further conditions will not be examined any more. If none of the conditions is evaluated as "true", the result of the last expression will be re-delivered. Of all given *expr i* parameters, only one will be evaluated. The result can be of integer, double, measure value, time or string types.

<u>**Examples:**</u>

```
  cond (x > 5, "x is larger than 5",
               "x is smaller than or equal 5")

  cond (x > 5, "x is larger than 5",
        x < 5, "x is smaller than 5",
               "x equals 5")
```

17.3.14   Loop Function (while)

<u>**Syntax:**</u>

while (*cond* **,** *loopexpr* [**,** *resultexpr*] **)**

As long as the condition *cond* is achieved, the loop body *loopexpr* will be evaluated. The loop body must have side effects, which will lead to the situation that the first condition is no longer achieved. After ending the loop, or when the first condition is not achieved, the result *resultexpr* will be re-delivered.

<u>**Example:**</u>

The following expressions calculate the factorial of 5:

```
  set (r, 1), set (i, 1),
  while (i <= 5, (set (r, r * i), set (i, i + 1))),
  r
```

17.3.15   Loop Function (for)

<u>**Syntax:**</u>

for (*lvalue* **,** *from* **,** *to* **,** *loopexpr* [**,** *resultexpr*] **)**

<u>**Example:**</u>

The following code returns "ABCDEFG":

```
set (s, ""),
for (i, 65, 71, (set (s, s + CHR i))),
s
```

17.3.16   Loop Function (fortok)

<u>**Syntax:**</u>

fortok (*lvalue* **,** *source* **,** *separator* **,** *loopexpr* [**,** *resultexpr*] **)**

The variable *lvalue* runs through all tokens in *source* separated by *separator*. Each run means that *loopexpr* is evaluated.

if *resultexpr* is specified, its value will be returned. Otherwise fortok does not provide direct results.

<u>**Example:**</u>

The following expression calculates the factorial of 5:

```
set (r, 1),
fortok (t, "2 3 4 5", " ", set (r, r * VAL t)),
r
```

17.3.17   Error Treatment (try, error)

If errors occur during calculation, the function **try()** allows for handling them. This function has two parameters. During the calculation, the system tries to evaluate the first parameter. If successful, the result will be returned. If an error occurs, the second parameter is evaluated and the appropriate result is returned.

<u>**Example:**</u>

In the following code 1 will be returned, if an error like dividing by zero happens:

```
try (pow (x, y) / (x * y), 1)
```

The opposite is the function **error()** which has a string as parameter. It generates the error message with the parameter as the message text.

17.3.18   Type Finding (type)

<u>**Syntax:**</u>

type(*anyExpr* **)**

With the function **type()**, the type of an expression can be detected. The following results are possible: **"string"**, **"integer"**, **"real"**, **"measure"**, **"time"**, **"expression"**, **"undefined"**.

>   Hint: **"undefined"** refers to a variable that has not been initialised.

17.3.19   Valuation (valofvar)

<u>**Syntax:**</u>

`valofvar(`*strValue* `)`

The function **valofvar( )** returns a value of the variable and its name *strValue*.

> Hint: **"undefined"** is a variable which has not been initialised.

17.3.20   Variable List (curvars)

<u>**Syntax:**</u>

`curvars()`

The function **curvars( )** returns a string, containing the names and values of all variables from the current context. For each variable used, the result contains a line with the format `<varname>:<value>`. In conjunction with a keyword at the beginning, this is valid LEO text.

17.3.21   Line Numbering (curlineno)

<u>**Syntax:**</u>

`curlineno()`

The function **curlineno( )** provides the cell number of the LEO element, which contains the expression. This function is useful in AdoScripts, to get the currently executed code line. However, it can not be uesd in LEO code (like GraphRep).

<u>**Example:**</u>

```
CC "AdoScript" INFOBOX
   ("We are now in line no. " + STR curlineno())
```

17.3.22   Determination of Times (getTickCount)

<u>**Syntax:**</u>

`getTickCount()`

The function **getTickCount( )** returns the time passed (in milliseconds) since the system was started. It is used for example in performance monitoring.

<u>**Example:**</u>

```
SET t1: (getTickCount())
# ...
SET t2: (getTickCount())
SET sec:((t2-t1) * 0.001
CC "AdoScript" INFOBOX "The calculation took " + STR sec + " seconds."
```

> **Hint:** After 49.7 days of concurrent on time, the system value is set back to zero.

17.3.23 Internal Values (internal)

<u>**Syntax:**</u>

```
internal(strValue)
```

The function **internal()** enables accessing internal values. Currently, these values can be read:

- **internal("APPWIN")** returns the address (HWND) of the application window.

- **internal("TOPWIN")** returns the address (HWND) of the top window.

A window address is a value of type long, which can be passed to external DLLs and used there as a parent window for a dialogue.

<u>**Example:**</u> Calling an external DLL containing a dialogue via AdoScript

```
SET hwnd:(internal("APPWIN"))
CALL dll:"extdlg.dll" function:"void showdlg(long hwnd)" hwnd:(hwnd)
```

18. ADL Syntax

The syntax of the language for ADL files (see chap. 9.3, p. 558) is based on the following grammar:

adl-file :	[*version*] *attrprofs models application-models modelgroups* .
***version* :**	**VERSION** *identifier* .
***attrprofs* :**	{ *attributeprofileinstance* } | { *attributeprofile-hierarchy* } .
***attributeprofileinstance* :**	*attributeprofile-definition* { *instanceattribute_setting* } .
***attributeprofile-definition* :**	**ATTRIBUTEPROFILE** *identifier* ':' *identifier* ':' *identifier* [*version*] .
***attributeprofile-hierarchy* :**	{ **ROOTATTRPROFDIR** *identifier* ':' *identifier* { *attributeprofile* } { *attributeprofile-directory* } } .
***attributeprofile* :**	*attributeprofile-def2* { *instanceattribute_setting* } .
***attributeprofile-def2* :**	**ATTRIBUTEPROFILE** *identifier* ':' *identifier* [*version*] .
***attributeprofile-directory* :**	**ATTRPROFDIR** *identifier* ':' *number* { *attributeprofile* } .
***models* :**	{ *we-model* | *bp-model* } .
***we-model* :**	*wem-definition* { *instanceattribute_setting* } *model* .
***bp-model* :**	*bpm-definition* { *instanceattribute_setting* } *model* .
***wem-definition* :**	(**WORKING ENVIRONMENT MODEL** *identifier* ':' *identifier* | (**WORKING ENVIRONMENT MODEL** *identifier* ':' *identifier* [*version*] **TYPE** *identifier*) .
***bpm-definition* :**	(**BUSINESS PROCESS MODEL** *identifier* ':' *identifier* | (**BUSINESS PROCESS MODEL** *identifier* ':' *identifier* [*version*] **TYPE** *identifier*) .
***model* :**	{ *instance* } { *relation* } .
***instance* :**	*instance-definition* { *instanceattribute_setting* } .
***instance-definition* :**	**INSTANCE** *identifier* ':' *identifier* .
***relation* :**	*relation-definition* { *instanceattribute_setting* } .
***relation-definition* :**	**RELATION** *identifier* **FROM** *identifier* ':' *identifier* **TO** *identifier* ':' *identifier* .
***instanceattribute_setting* :**	**ATTRIBUTE** *identifier* **VALUE** *attrval* .
***application-models* :**	{ *application-model-definition model-attachments* } .
***application-model-definition* :**	**APPLICATION MODEL** *identifier* ':' *identifier* .
***model-attachments* :**	{ **BUSINESS PROCESS MODEL** *identifier* [*version*] **TYPE** *identifier* }

```
                          WORKING ENVIRONMENT MODEL identifier [ version ]
                          TYPE identifier
                          { BUSINESS PROCESS MODEL identifier [ version ]
                          TYPE identifier } |
                          BUSINESS PROCESS MODEL identifier [ version ]
                          TYPE identifier
                          { BUSINESS PROCESS MODEL identifier [ version ]
                          TYPE identifier } .
```

modelgroups : { **ROOTMODELGROUP** *identifier* ':' *identifier*
  { *reference* } { *hierarchy* } } .

reference : **BUSINESS PROCESS MODEL** *identifier* **TYPE** *identifier* |
  WORKING ENVIRONMENT MODEL '*identifier* **TYPE** *identifier* .

hierarchy : **MODELGROUP** *identifier* ':' *number* { *reference* } .

identifier : '**<**' *any_characters_but_newline* '**>**' .

attrval : *number* | '**"**' *any_characters* '**"**' | *recordval* .

recordval : { **RECORD** { **ATTRIBUTE** *identifier* **VALUE** *val* } **END** } .

val : *number* | '**"**' *any_characters* '**"**' .

In case of text being too long, the value represented by *val* will be split in the ADL file as follows:

The end of a line is denoted by a '**"**', then, at the beginning of the next line (after some blanks) there is another '**"**' and then the text continues (insert the string **"**, \n and **"** between the splitted *val* parts).

Meaning of the values:

- **AttrProfileName** contains the name of the ADONIS attribute profile.

- **AttrProfileClassName** contains the name of the ADONIS attribute profile classes.

- **LibName** contains the name of the ADONIS application library, on which the ADONIS attribute profile, the attribute profile of the ADONIS attribute profile group, or the ADONIS model are based.

- **AttrProfileGroupName** contains the name of the ADONIS attribute profile group.

- **ModelName** contains the name of the ADONIS model.

- **ModelTypeName** contains the name of the model type.

- **ObjName** contains the name of the object.

- **ClassName** contains the name of the class.

- **RelName** contains the name of the relation.

- **AttrName** contains the name of the attribute.

- **ApplModelName** contains the name of the ADONIS-application model.

- **ModelGroupName** contains the name of the ADONIS-model group.

   Hint: If in the above value the **sharp brackets** (**>**) are contained, they must be masked i.e. instead of '>' the '\>' must be quoted.

   Hint: If the value *text* is written in **quotation marks** (**"**), they must be masked i.e. instead of ' " ' the ' \" ' must be quoted.

   Hint: If the **backslashes** (****) are part of the value, they must be masked i.e. instead of '\' the '\\' must be quoted.

The value *VersNr* contains the version number i.e. the ADONIS version, with which the ADL export will be executed.

The value *Depth* says, on which hierachical level the attribute profile group and the modelgroup are situated and it contains a figure greater than or equal to 1.

The value *number* contains an integer (see chap. 5.7, p. 538) or a floating-point number (see chap. 5.8, p. 538).

The value *text* contains a concrete attribute value of a defined ADONIS attribute type (see chap. 5., p. 536), but with an exception for the attribute type "integer" or "floating-point number".

19. UDL Syntax

The syntax of the language for the UDL- files base on the following grammar:

<u>***udl-file*** **:**</u>	[*version*] [*users*] { *sysusers* } [*usergroups*] [*sysusergroups*] .
***version* :**	**VERSION** *identifier* .
***users* :**	{ **USER** *identifier* { *instanceattribute-setting* } } .
***sysusers:* :**	{ **SYSUSER** *identifier* **PROVIDER** *identifier* **LOGONTYPE** *identifier* **USERDISPLAYNAME** *identifier* **DOMAIN** *identifier* **DOMAINDISPLAYNAME** *identifier* *instanceattribute-setting* } } .
***instanceattribute-setting* :**	**ATTRIBUTE** *identifier* **VALUE** *val* .
***usergroups* :**	**USERGROUP** *identifier* { *instanceattribute-setting* } { *userreference* } { *modelgroupaccess* } } .
***userreference* :**	**USER** *identifier* .
***sysusergroups* :**	{ **SYSUSERGROUP** *identifier* { *instanceattribute-setting* } { *sysuserreference* } { *modelgroupaccess* } } .
***sysuserreference* :**	**SYSUSER** *identifier* .
***modelgroupaccess* :**	**MODELGROUP** *identifier* '**:**' *identifier* **ACCESS** *boolValue* .
***identifier* :**	'**<**' *any_characters_but_newline* '**>**' .
***val* :**	*number* │ '**"**' *any_characters* '**"**' .
***boolValue* :**	**0** │ **1** .

System user and system user groups:

This function exists from the ADONIS version 3.7. Thanks to it, the system users (in the domain registered user) can be transfered to ADONIS. If the **SYSTEMUSER** entry is found in the UDL file to be imported and the database support for the Single-Sign-on is available, the system will search for all the corresponding users in all the available domains. If such a user is found, he will be registered in ADONIS as a system user, who can use the domain administrator password as password. If there is no domain available, the user will be transformed to an internal standard user. In such cases the standard password is assigned. The system users are organised in system user groups and standard users in the user groups. These two cannot be unselected.

> **Hint:** If the values *UserName*,   *UsergroupName*, *SysUsergroupName*, *ModelgroupName* or *LibName* **contain brackets** (**<** or **>**), they must be masked i.e. instead of '\' the '\\' must be quoted.

> **Hint:** If the value *LibName* or *Text* **contains double inverted commas** (**"**), they must be masked i.e. instead of '\' the '\\' must be quoted.

> **Hint:** If the values *UserName*, *UsergroupName*, *SysUsergroupName*, *ModelgroupName*, *LibName* or *Text* **contain backslashes** (****), they must be masked i.e. instead of '\' the '\\' must be quoted.

If the text is too long, the value presented by the *val* is divided into UDL files as follows:

The end of the line is denoted with a '"', then, at the beginning of the next line (after some blanks) there is another '"' and then the text continues (insert the string **"**, \n and **"** between the splitted *val* parts).

<u>**User attributes:**</u>

Application library(STRING)

It contains the name of the application library, which is assigned to the user (it is not a LEO attribute).

Administration Toolkit (INTEGER)

If the user has the right to use the Administration Toolkit, the value is **1**, otherwise it equals **0**.

GPM Toolkit (INTEGER)

If the user has the right to use the GPM Toolkit, the value is **1**, otherwise it equals **0**.

User info (STRING)

The user info is a string, which can contain any type of text.

Access to components (STRING)

The string concerning access to the components starts with "**:**". It can be followed by one or many INTEGER values, which are always ended with "**:**" (e.g. :1:2:32:).

The INTEGER values respond to the bit mask values, which activate the following components:

```
CONF_ACQUISITION_COMPONENT = 1
CONF_REPORTS_COMPONENT = 2
CONF_STANDARDQUERIES_COMPONENT = 4
CONF_USERDEFQUERIES_COMPONENT = 8
CONF_PATHANALYSIS_COMPONENT = 16
CONF_WORKLOADANALYSIS_COMPONENT = 32
CONF_VOLUMEANALYSIS_COMPONENT = 64
CONF_IMPORTEXPORT_COMPONENT = 128
CONF_TRANSFORMATION_COMPONENT =256
CONF_FLOWMARK_COMPONENT = 512
CONF_EVALUATION_COMPONENT = 1024
CONF_RELTABLES_COMPONENT = 2048
CONF_SIM4_COMPONENT = 4096
CONF_EVALQUERIES_COMPONENT = 8192
CONF_VOLUMEDYNAMIC_COMPONENT = 16384
CONF_CASE40LINK_COMPONENT = 32768
CONF_DOCUMENTATION_COMPONENT = 65536
CONF_COSTCUTTING_COMPONENT = 131072
CONF_ANALYTIC_COMPONENT = 262144
CONF_AGENT_COMPONENT = 524288
```

Modelling info (STRING)

This attribute is used in the Modelling Component for storing various user-specific settings which have to remain available after the session ends.

Arrangement info (STRING)

The string of a user-specific arrangement function is equivalent to the pre-defined arrangement function. Only after the name of the arrangement function (**PROFILE** "Function name") the value for this type must be changed with `type:cust`.

User settings (STRING)

This attribute is used by various ADONIS components for storing various user-specific settings which have to remain available after the session ends.

Message (STRING)

The message received by the user. The entry must be created for each message.

Password (STRING)

This string contains the encrypted password for the user. This attribute can be optionally exported.

20. ADONIS BPMS Application Library

The **ADONIS BPMS Application Library** is created automatically during the ADONIS installation.

The **ADONIS BPMS Application Library** consists of **ADONIS BPMS BP Library** (contains classes and relations for the Business Process Modelling) and **ADONIS BPMS WE Library** (contains classes and relations for the Working Environment modelling).

> **Hint:** Detailed information concerning the ADONIS BPMS Application Library and the ADONIS standard method (and all the library specific functionality) are gathered in the User Manual "**Volume III - ADONIS Standard Method Manual**".

21. Ergonomics Compliance Statement for ADONIS

The increased usage of information and communication technology has changed the working day completely. In order to avoid additional psychological stress it is important that the work and tools for the work are designed accordingly. ADONIS is a tool for business modelling and globalisation which fully complies with the requirements of software ergonomics as the following research will show.

The ergonomics analysis for ADONIS contains:

- **Human criteria** (see chap. 21.1, p. 694)

- **Dialogue design** (see chap. 21.2, p. 697)

- **Conformity with the Windows User Interface Style Guide** (see chap. 21.3, p. 704)

21.1 Human Criteria

The **seven human criteria for the work design according to DIN EN ISO 9241-2** establishes requirements, recommendations and principles for the design of interactive Multimedia User Interfaces that integrate and synchronize different media (static media such as text, graphs, images, and dynamic media such as audio, animation and video). By taking those criteria into consideration while designing the work tasks become not only harmless, restrictionless and reasonable, but also what's most important for the work design - personally beneficial.

Those criteria are:

User orienation
> The work design should *take into account experiences and skills of the user group.*

Versatility
> The design of the work assignments should consider, *that an appropriate variety of skills and activities are applied.*

Integrity
> The work design should guarantee, *that the tasks to be achieved will be recognised as integrated work units and not fragments of those.*

Signification
> The design of the work assignments should guarantee that the tasks to be achieved provide *a clear and understandable description of the system's functions.*

Scope of action
> The work design should accommodate *an appropriate scope of action with regard to the order and rate of work.*

Report messages
> The design of the work assignments should provide *sufficient report messages on task performance, meaningful to the user.*

Development possibilities
> The design of the work assignments should *provide possibilities for further development of existing and new skills within the nature of the task.*

In the chapters below you will find examples of how to implement the above criteria in ADONIS.

21.1.1   User Orientation

The software ADONIS is offered as a package togethr with consulting and training days. Hereby BOC Group assures that persons potentially working with ADONIS will be introduced to the way of thinking of the illustrated business management of ADONIS and **ideally implement this in accordance with the tool**.

For support there is a user guide provided, that describes ADONIS in five parts:

● Introduction

● User's guide

● Method manual

● Method definition and administration manual

● Installation and database administration manual

Furthermore the above manuals are available in the context sensitive help that will display appropriate pages with respect to the user's given activity.

Through the metamodelling technique, ADONIS can be fully customised to customers' wishes. In that way the client's specific symbols, modelling segments and model types can be illustrated in ADONIS. In ADONIS the user will find already familiar methods and through them can quickly obtain an effective rate of output.

21.1.2   Versatility

The versatility arises from **the universal use the of capabilities of ADONIS for all activities of business modelling**. The individual activities are again located in the ADONIS components.

● Acquisition of data needed for modelling

● Model creation (processes and structures)

● Analysis of the models

● Evaluation of business processes

● Import-/Export of models to documents and also for exchanging with other tools

21.1.3   Integrity and Signification

The integrity is guaranteed through **Business Process Management Systems (BPMS) Paradigm** enabling a holisitc look at the Company, its processes and structures and completely depicting them inADONIS.

With the client/server architecture and multi-user operation of ADONIS, team project work is supported in order to solve the complex tasks.

The holistic view of the BPMS Paradigm, but also the possibility to immediately see the work results of the other users, allow teamwork on highest level.

21.1.4   Scope of Action

The modelling in ADONIS must be understood as a creative task. A graphical model editor supplies the user with illustrations of the models, while a tabular model editor allows a data-oriented approach.

Hint: In ADONIS there are several methods to achieve the same goal.

In order to comply with these requirements, there are several places in ADONIS, where you can find cross references to other functions. As an example **the call of the model group management** is displayed here:

● The management of the model group can be directly executed by the menu option **"Model" - "Modelgroup"**:
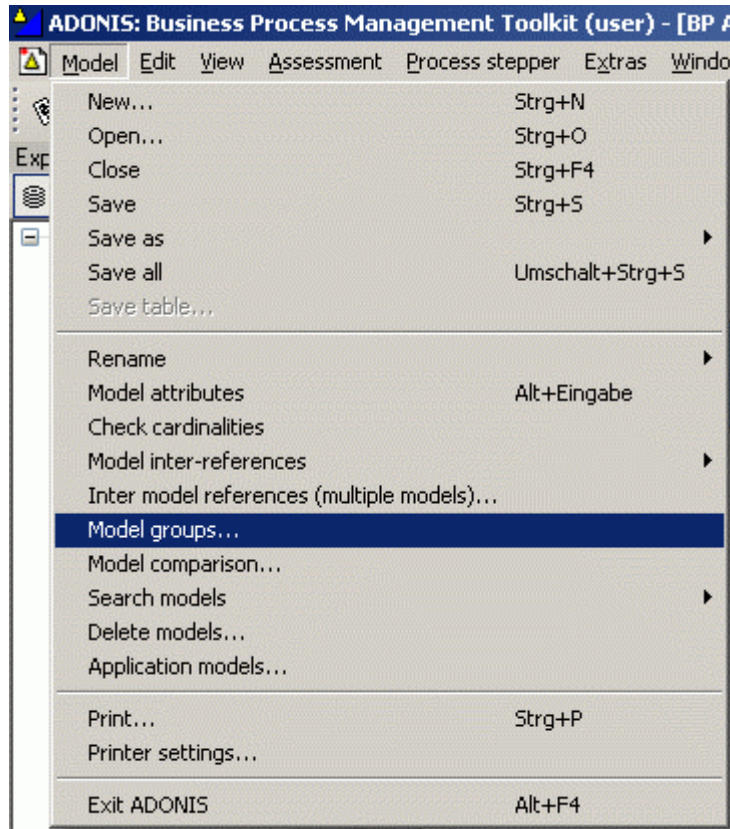
Figure 401: The direct way through the menu

● The model group management can be opened through the context menu in the dialogue where the model list is displayed. The figure shows the "Open model" dialogue with the opened context menu:
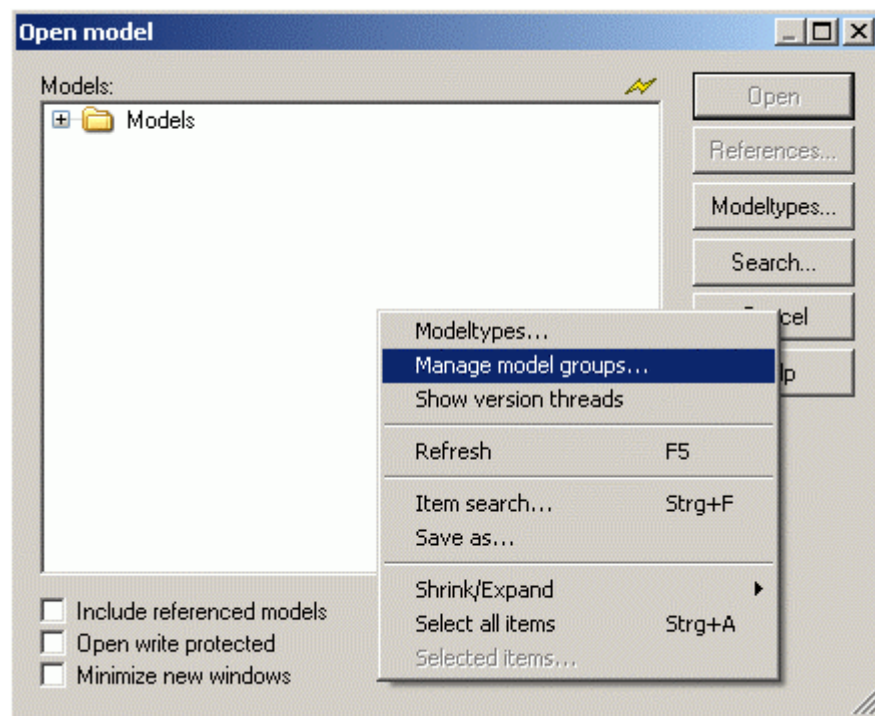
Figure 402: Indirect way through the context menu

21.1.5 Report Messages

In ADONIS the dialogues (see chap. 21.2, p. 697) are constructed in a way that **each user interaction will be answered with the corresponding report message.** Additional help for the report messages is also available in ADONIS

The dialogue commands allow the tool to guarantee or assess the quality of the performed work:

- The modelling method that is the basis for ADONIS meta model is highly defined. It allows the formal assessment of the model structure (i.e. cardinalities of the relations) and stops invalid modelling (i.e. connection from instances, where there is no relations provided).

- Quality checks can be carried out in the form of queries as well as by means of script-extension. Free to formulate queries allow ad-hoc checking of the model state.

- If the model is intended to be simulated, the ability to simulate already counts as the quality criterion that must be granted for carrying out the simulation.

ADONIS also supports manual report messages in the form of review mechanisms. Those mechanisms can be established in the client-specific, special modelling methodology (i.e. model states, notes display, errors visualisation).

21.2 Dialogue Design

The dialogue representation can be assessed in light of formulating principles. The principles are quoted in the **Norm DIN EN ISO 9241-10** (valid for the German speaking region)

Task appropriateness

A dialogue is task appropriate, when it *effectively and efficiently fulfills its tasks*.

Self-description ability

> The dialogue is able to describe itself, when *every individual dialogue step is immediately understandable* through the report message of the dialogue system or the user's question will be answered.

Expectation conformity

> A dialogue has expectations to *conform, when it is consistent and user's features correspond*.

Controllability

> A dialogue is controllable, when the user is able to start the dialogue flow, and can influence its speed and position, until the destination is reached.

Error tolerance

> A dialogue is error tolerant, when *the intended work result* can be, in spite of recognisable incorrectness of the input, achieved with *either none or minimal correction cost* from a user's point of view.

Individualisation

> A dialogue is individualised, when *the dialogue system permits adaptations on the work task request and the individual user preferences*.

Learn ability

> A dialogue has learning ability if it *supports the user in learning the dialogue system*.

21.2.1   Task Appropriateness

Efficiency of the work design is achieved in ADONIS through suitable separation of the working steps, dividing dialogues into logically related groups of control elements, and through the selected use of default buttons. Also the state of the default buttons is changed to stop premature closure of the dialogue.

> **Hint:** The default button is the button that will be executed when pressing the <Enter> key.

The figure below shows how the element division occurs in logically related groups in the form of registry cards on the basis of the ADL Import dialogues. It is obvious that the default button "OK", will only be active after the provision of the data name, because a previous continuation of that action would not make sense:
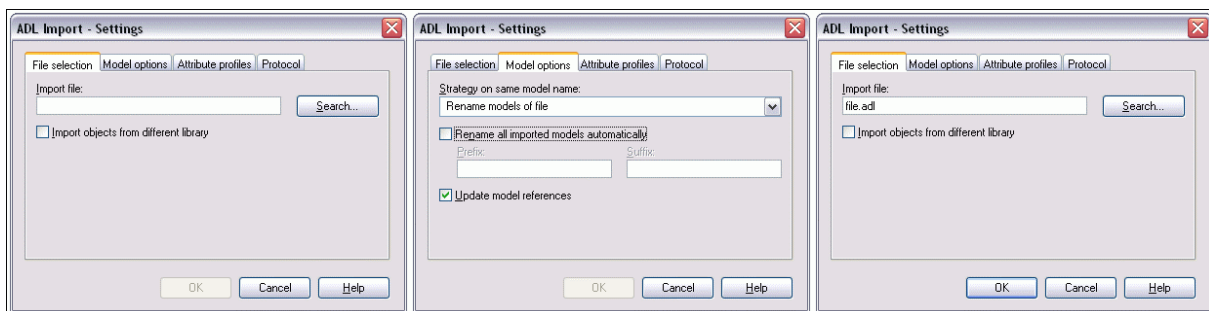
Figure 403: The figure shows the examplorary element division and behavoiur of the default button "OK"

21.2.2   Self-description Ability

By designing the ADONIS user interface, a special emphasis was put on clear timing of the control elements. Buttons are called with verbs where appropriate. Other texts are named in a detailed manner and are consistent with naming conventions.

The following figure illustrates the "open model" dialogue that provides some functions that are clearly explained through unambiguous wording. In every dialogue the context sensitive help is at the user's disposal and immediately provides information to the dialogue.
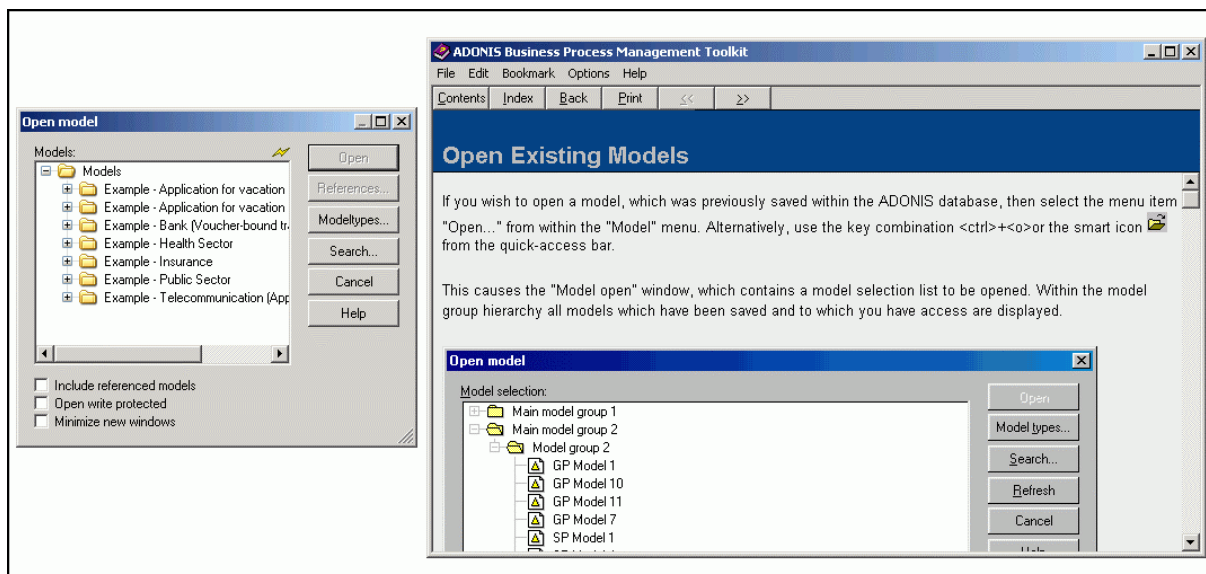
Figure 404: The elements of the "Open model" dialogue

Tooltips will provide comments for ambiguous names. The following figure illustrates a tip of the "Update" button in the "Open model" dialogue.
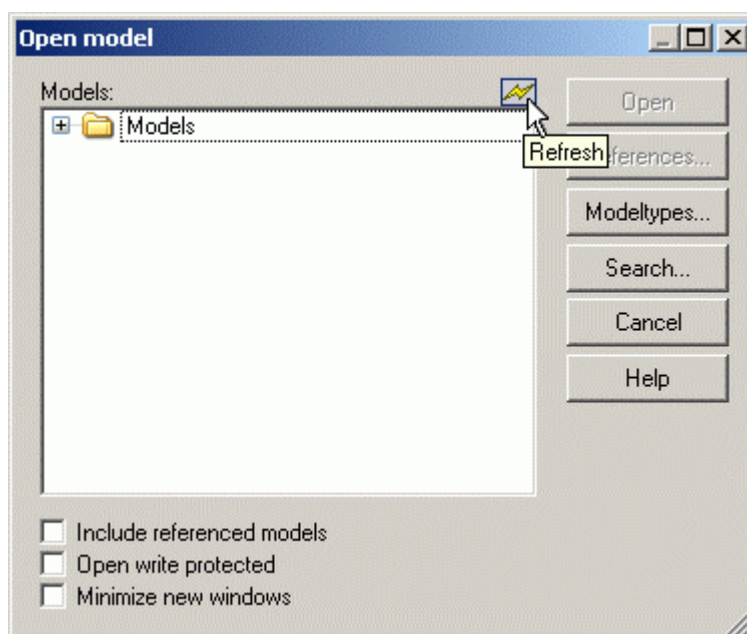
Figure 405: Tooltips

> **Hint:** A Tooltip is a small (classically light yellow) window that contains a short description of the button. It appears when the mouse pointer hovers over the button for some time.

A progress of fairly long actions (like opening a model, import, export, simulation etc.) will be visualised by the status bar. The status bar will display the result at the end of the action.
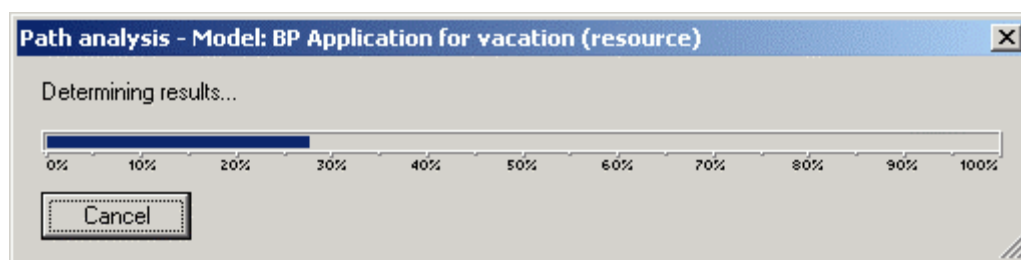
Figure 406: Progress bar for longer operations

If the operation lasts more than ten seconds, the status bar will display time information.

21.2.3   Expectation Conformity

ADONIS conforms to the Windows User Interface Style Guide (see chap. 21.3, p. 704). It also uses the icons and menus of the Microsoft Office products. ADONIS-specific functionality is illustrated through specially designed smart icons. The nomenclature in ADONIS is arranged uniformly; the dialogues consistently use conceptualities. In the same way the structure of the dialogue has been designed homogeneously: dialogues of the same kind represent elements of the same type. The following figure illustrates the comparison of the dialogue management "delete a model" and "model group management". On the left side, the model (group) list will be displayed, on the right side are the buttons with the possible operations. The "help" button is located at the last position. On the right side, above from the model (group) list, there is an icon to update the list.

> **Hint:** Icons are graphical symbols for application functions. Usually they are represented by small buttons on the toolbar.
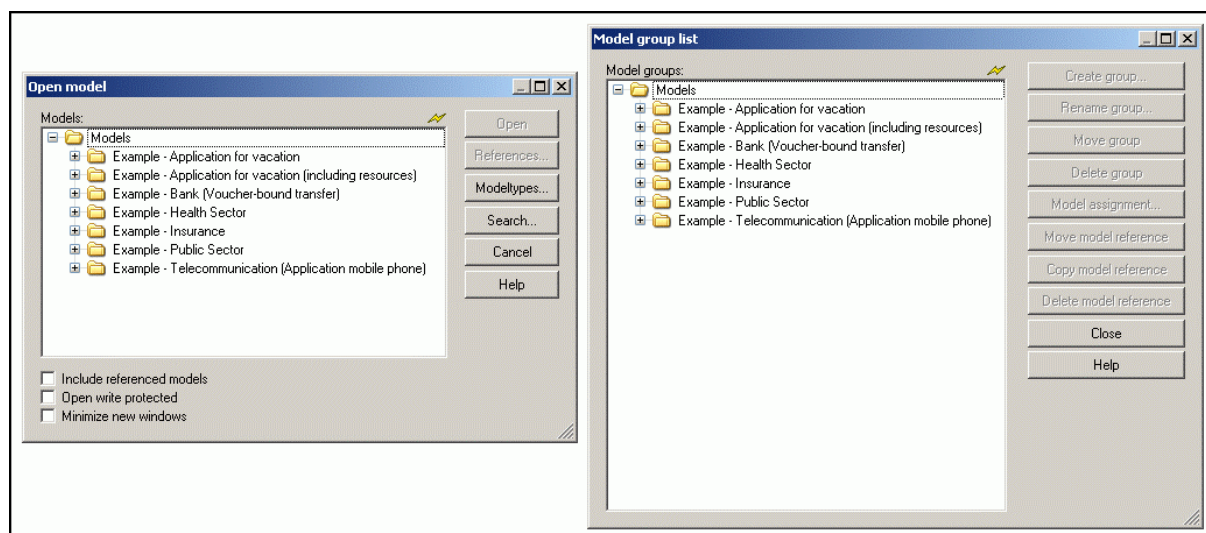
Figure 407: Comparison of the "Open a model" and "List of model groups" dialogues"

21.2.4   Controllability

ADONIS has been developed as a so-called MDI application. This allows simultaneous handling of several models. Arrow keys serve to switch between model windows, a selection window allows direct navigation.

Figure 408: Switching between open model windows

Hint: MDI = Multiple Document Interface, denotes that in the one user interface more than one document (in ADONIS models are documents) can be opened at the same time.

Some dialogues and the ADONIS Notebook are structured by means of register cards. Register cards serve to clearly group the elements from a logistical point of view. To switch between register cards press <Crtl>+<Tab> or alternatively <Shift>+<Crtl>+<Tab>. The ADL import dialogue is an example of a register card (see fig. 403); The ADONIS Notebook is shown in the figure below.

The ADONIS Notebook is implemented as a modeless window. This allows on one hand opening more than one notebook and on the other hand to work on the model while one or more notebooks are open.

Figure 409: A sample ADONIS Notebook

> **Hint:** Modeless dialogues do not exclusively receive the focus in the application. Thus it is possible to call and click on other windows (for example the main window or other dialogues) while the modeless dialogue remains open.

ADONIS is structured in components. As a result, the "usefulness" (see chap. 21.1.2, p. 695) of menus and toolbars is achieved.

The order of components corresponds to the workflow of the business modelling. Context menus offer advanced functionality. To increace clarity, such functions will remain hidden outside the context menus. Demanding or professional users can use those functions to gain extended controlability (see chap. 21.1.4, p. 695).

21.2.5 Error Tolerance

While carrying out work tasks, the user is supported by the error messages which are identifed by unified error codes (written in the square brackets). By clicking on the "Help" button, a help page with a precise explanation and solution suggestions is provided:
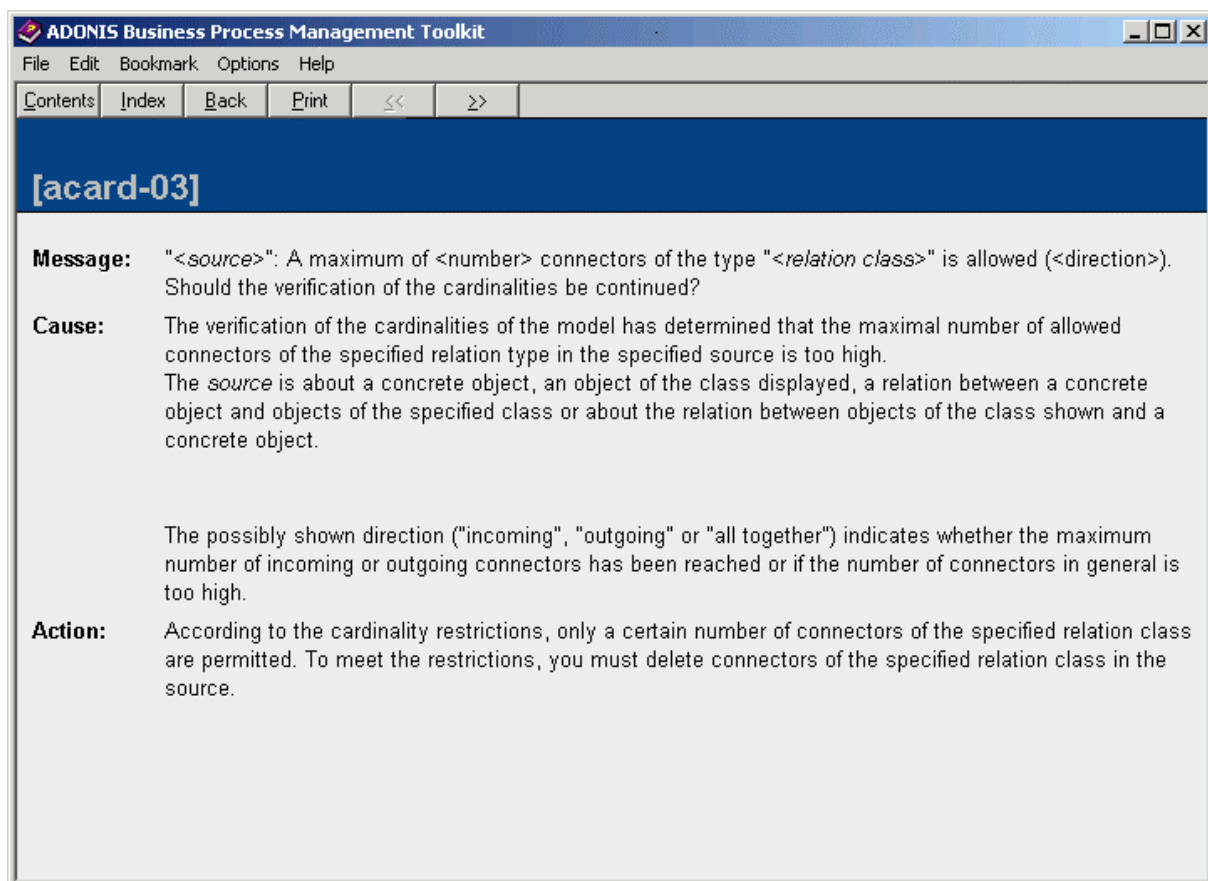
Figure 410: A typical error message

Numbers and date expressions will be automatically formed or corrected:

Figure 411: Automatical rounding up to the whole number

ADONIS Administration Toolkit

False input leads to later actions being not available (i.e. the confirmation button remains deactivated).

21.2.6 Individuality

The meta-model concept of ADONIS allows client-specific adaptations and the development of new modelling methods. For that reason it is possible to adapt graphical visualisations of the modelled objects, layout, size of the generated documents etc.

For individual users, components can be shown or hidden as needed. The toolbar (quick-access bar) can be modified using the AdoScript language.

21.2.7 Learn Beneficiality

ADONIS uses the icons and menu entries of the Micorosoft Office product series. The ADONIS-specific functionality is illustrated through the specially designed smart-icons being consequently re-used. The dialogue icons in dialogues and in the ADONIS Notebook, are a good example of this.

The figure shows the dialogue icons for adding, deleting and following of objects and model references:
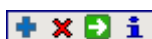
Figure 412: Some smart-icons

The learning ability benefits from the clear structure of the components in ADONIS. Furthermore, tooltips and shortcuts which appear directly on the surface area are easy to learn. The used key-combinations correspond to the Windows User Interface Style Guide and also to the stardard key combination of e.g. Microsoft Office:

Figure 413: Standard menu and standard key combination

Hint: A shortcut is a key combination that can be used to directly call program functions. Shortcuts are used i.e. for frequently used commands in menus.

ADONIS offers a possibility to preview the printout or the generated graphics, so that the user can see the result before performing the operation:
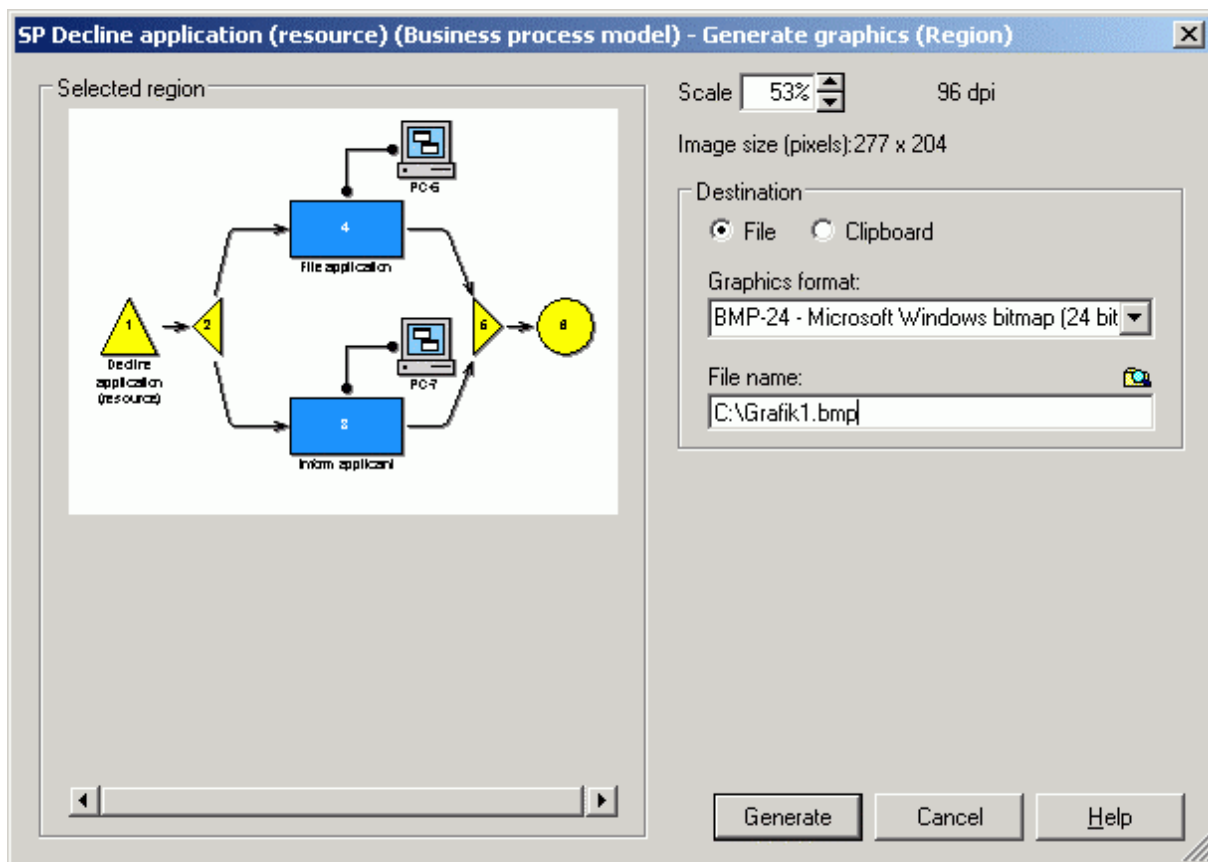
Figure 414: Graphics generation

The input in the Notebook is immediately applied to the visualised models in the model editor. As a result, a very quick feedback on performed changes is provided. Using this feature, the user can easily see the consequences of the input.

21.3 Conformity With the Windows User Interface Style Guide

ADONIS has been and currently is developed as a Microsoft Windows application and the user interface is designed according to the standards of the Microsoft Windows User Interface Style Guide.

Microsoft with its comprehensive **Windows User Interface Style Guide** (see chap. 21.3.1, p. 705) defines how a Windows application should be designed. This is thought to ensure a common "Look & Feel" of Windows applications and also ensures that the thoughtfully developed ergonomics of the style guide are also transferred to applications not developed by Microsoft. The ADONIS user interface takes these requirements into account.

Some of these points listed below illustrate this:

- ADONIS has been designed as a MDI application. This allows for the ability to work with several models at the one time. Each model is shown in a separate window. Using the menu "Window" they all can be accessed and administered.

- Colours, fonts and font sizes as well as international formats are based on the global Windows system settings.

- the font size can be adjusted to the requirements of the respective user. The user can change the font size of the visualised text of a model (on the drawing area) (menu "View", menu item "Font size") and can select the font size used in dialogues and menus when starting ADONIS (command line parameter "-h").

- All menus and dialogues can be accessed using either the mouse or the keyboard. The user is further supported by shortcuts.

- Wherever possible, Windows standard dialogues are used. This is of an advantage as the user will be familiar with these dialogues from other applications and they will always use the language of the operating system:

  - query and status messages

  - file handling (e.g. "Search" button to enter file names during the import, export, documentation generation, protocol files etc.)

  - printer settings

  By using standard dialogues the system settings for sound are also applied (acoustic signals).

21.3.1   Literature

- Microsoft: Official Guidelines for User Interface Developers and Designers. Microsoft Press, 1999.

- Microsoft: The Windows Interface Guidelines for Software Design. Microsoft Press, 1995.

22. Glossary

The glossary explains the expressions used in ADONIS.

The symbol "→" refers to further entries in the glossary.

- A -

ABL. **A**DONIS **B**inary **L**anguage; file format, in which →application libraries can be saved (exported).

Accelerator. Letter marked by underlining. Occurs in titles of menus and menu options, in field and list names of input fields and in chapter headings and attribute names of →ADONIS Notebooks. By entering the accelerator (sometimes in connection with the <Alt> key) you can call functions and navigate inside both windows and ADONIS Notebooks via the keyboard.

Access Rights. The access rights (read-write/read-only/no access) to an →ADONIS model group are assigned to the →ADONIS user groups by the →ADONIS administrator.

Acquisition (component). →Component of the →Business Process Management Toolkit. Supports the core activity "Acquisition" in organisation projects with the help of →Acquisition Tables.

Acquisition Tables (HOMER). Functionality in the →Acquisition Component in the →Business Process Management Toolkit. The acquisition tables help you to store and manage the data acquired and transfer into ADONIS.

ACR. **A**DONIS **C**omparable **R**epresentation; file format, in which simulation and analysis results, which are displayed in the →ADONIS browser can be saved. ACR-files are needed in the →Evaluation Component for the →comparison of results.

Activity. Smallest entity of a business process. Activities describe what is done in a →business process.

Activity, cooperative asynchronous. An →activity carried out by several →performers who work together but not at the same time.

Activity, cooperative synchronous. An →activity carried out by several →performers who work together at the same time.

ADL. **A**DONIS **D**efinition **L**anguage; in which →ADONIS models and →ADONIS model groups as well as →ADONIS attribute profile and →ADONIS attribute profile groups can be saved (exported).

ADL Export. Functionality in the →Import/Export Component in the →Business Process Management Toolkit and in the →Model Management in the →Administration Toolkit. Using the ADL export profiles, you can export →ADONIS models and →ADONIS model groups as well as →ADONIS attribute profile and →ADONIS attribute profile groups to →ADL files.

ADL Import. Functionality of the →Import/Export Component in the →Business Process Management Toolkit and in the →Model Management in the →Administration Toolkit. Using the ADL import functionality, you can import →ADL files (→ADONIS models and →ADONIS model groups) as well as →ADONIS attribute profile and →ADONIS attribute profile groups into ADONIS and store them in the →ADONIS database.

Administration Toolkit. ADONIS tool, needed by the →ADONIS administrator in order to manage →ADONIS users, →ADONIS libraries, →ADONIS models and →ADONIS attribute profiles. The Administration Toolkit consists of the →User Management, →Library Management, →Model Management, →Attribute Profile Management and →Component Management →components.

ADO*it*. IT Service and IT Architecture Management Tool (manufacturer: BOC Asset Management GmbH). Part of the →Management Office.

ADO*log*. Supply Chain Management Tool (manufacturer: BOC Asset Management GmbH). Part of the →Management Office.

ADONIS. Business Process and Knowledge Management Tool (manufacturer: BOC Asset Management GmbH). Part of the →Management Office.

ADONIS Administrator. Person, who is responsible for the configuration of ADONIS ("→Customising"), the →User Management, →Library Management, →Model Management, →Attribute Profile Management and →Component Management.

ADONIS Attribute Profile. →Attribute profile

ADONIS Attribute Profile Group. →attribute profile group

ADONIS BPMS Application Library. →Application library, automatically generated during the installation of a →ADONIS database for ADONIS version 3.9 and up and assigned to the standard user "Admin". The ADONIS BPMS Application Library consists of the →ADONIS BPMS BP Library and the →ADONIS BPMS WE Library. The ADONIS BPMS Application Library is the successor and an extension to the →ADONIS-Standard-Anwendungsbibliothek.

ADONIS BPMS WE Library Working Environment library, automatically generated during the installation of a →ADONIS database for ADONIS version 3.9 and up. The ADONIS BPMS WE Library is part of the →ADONIS BPMS Application Library. With the →classes and →relations defined in the ADONIS BPMS WE Library, models of the →model type →"Working Environment model" can be created.

ADONIS BPMS BP Library. , automatically generated during the installation of a →ADONIS database for ADONIS version 3.9 and up. The ADONIS BPMS BP Library is part of the →ADONIS BPMS Application Library. With the →classes and →relations defined in the ADONIS BPMS BP Library, models of the →model types →"Company Map", →"Business Process Model", →"Document Model", →Product Model, →IT System Model and →"Use Case Diagram" can be created.

ADONIS Browser. Special result window used in the →Business Process Management Toolkit(in the components →Modelling, →Analysis, →Simulation and →Evaluation) as well as in the →Administration Toolkit (in the components →Library Management and →Attribute Profile Management). The results are represented in tabular form and can be displayed graphically. They can also be printed out, saved in a file in →TXT format (table and text), →RTF format or →HTML format.

ADONIS Business Edition. Compared to the complete ADONIS Professional Edition, this is a restricted configuration (→ADONIS configuration) of the →Business Process Management Toolkit. The ADONIS Business Edition consists of the components →Modelling, →Analysis, →Documentation and →Import/Export.

ADONIS Business Edition. Compared to the complete ADONIS Professional Edition, this is a restricted configuration (→ADONIS configuration) of the →Business Process Management Toolkit. The ADONIS Business Edition consists of the components →Modelling, →Analysis, →Documentation and →Import/Export.

ADONIS Component. Part of ADONIS software. The standard ADONIS components are the →Business Process ManagementToolkit and the →Administration Toolkit as well as the →database administration. In addition, the →Process Cost Analysis,^→Dynamic Evaluation Module, →case/4/0 coupling and →objectiF coupling can be purchased as additional components.

ADONIS Configuration. A version of the →Business Process Management Toolkit in which the →components are configured according to the customer's needs. The minimal ADONIS configuration consists of the →modelling component.

ADONIS Database. All ADONIS data is stored in the ADONIS database.

ADONIS Database Administration. The ADONIS Database administration contains the programs necessary for generating, saving and restoring →ADONIS databases.

ADONIS Default Language. The language ADONIS starts by default with. The default language is set during installation.

ADONIS Directory. Directory in which ADONIS has been installed. Default: `C:\Program files\BOC\ADONIS39`.

ADONIS Explorer. Tool providing an overview of the →models/model groups stored in the ADONIS database and offering various manipulation possibilities.

ADONIS Inspector. Tool that supplies the →ADONIS user with various details about the active modelling environment.

ADONIS Installation Directory. →ADONIS directory

ADONIS Mail. Messaging function in ADONIS, using which →ADONIS users can exchange (→express) messages. The messages will be saved to the→ADONIS database .

ADONIS Model. Aggregation of →objects of various →classes that serve to represent the essential aspects of the real world for the →ADONIS user. The models serve as a basis for →components such as →analysis, →simulation and →evaluation.

ADONIS Model Group. Group, in which →ADONIS models are referenced. →Access rights for →ADONIS user groups and the →ADONIS users referenced therein are defined for an ADONIS model group.

ADONIS Navigator. Tool which shows an overview of the active →model. The part of the model, which is currently visible in the →model window is highlighted and can be used for navigating through the model.

ADONIS Notebook. Special dialogue window where the attributes of →objects, →connectors, →attribute profiles, →models or →libraries may be changed.

ADONIS Professional Edition. Complete ADONIS version of the →Business Process Management Toolkit (→ADONIS configuration). The ADONIS Professional Edition in its standard configuration consists of the components →Acquisition, →Modelling, →Analysis, →Simulation, →Evaluation, →Documentation and →Import/Export.

ADONIS Program Folder. Contains the program icons for the installed ADONIS toolkits, the →online manuals and (if applicable) the →ADONIS database administration.

ADONIS Rights. Rights of an →ADONIS user to access the →Administration Toolkit and/or →Business Process Management Toolkit.

ADONIS Standard Application Library. →Application library that is automatically generated when an →ADONIS database is generated for ADONIS 3.81 or earlier. Since ADONIS 3.9, the →ADONIS BPMS Application Library is generated automatically. The ADONIS Standard Application Library is assigned to the standard user "Admin". It consists of the →ADONIS Standard BP library and the →ADONIS Standard WE library.

ADONIS Standard BP Library. →Business process library that is automatically generated when an →ADONIS database (up to version 3.81) is generated. The ADONIS Standard BP Library is part of the →ADONIS Standard Application Library. You can create models of types →"Business Process Model"→"Document Model", →"Use Case Diagram" and →"Company Map" with the classes and relations defined in the ADONIS Standard BP Library.

ADONIS Standard WE Library. Working Environment library that is automatically generated when an →ADONIS database (up to version 3.81) is created. The ADONIS Standard WE Library is part of the →ADONIS Standard Application Library.. You can create models of the model type →"Working Environment Model" with the classes and relations defined in the ADONIS Standard WE Library.

ADONIS User. User of →ADONIS components. The ADONIS users stored in the →ADONIS database are displayed in the →user list. ADONIS users may be exported (optionally including →ADONIS user groups) to →UDL files or imported from →UDL files.

ADONIS User Group. Group, in which →ADONIS users are referenced. An ADONIS user group has →access rights to →ADONIS model groups and the →ADONIS models referenced within these. The ADONIS user groups stored in the →ADONIS database are displayed in the →user group list.

ADONIS user groups may (optionally including →ADONIS users) be exported into →UDL files or imported from UDL files.

ADONIS User Management (component). A →component of the →Administration Toolkit where →ADONIS users can be created, deleted and their settings can be changed →ADONIS user groups can be created, renamed and deleted. An ADONIS user can be assigned to ADONIS user groups and ADONIS users can be imported and exported.

ADONIS Window. Graphical user interface of ADONIS. You can call the individual →components from within the ADONIS window.

ADO*score*. Strategy and Performance Management Tool (manufacturer: BOC Asset Management GmbH). Part of the →Management Office.

AdoScript. Script language, using which you can implement customised functionality extensions in ADONIS. Moreover AdoScript enables the call of any external programs and functions in DLLs.

Agent. Created by a →ADONIS user in the →Simulation Component. Calculates user-defined simulation results during the simulation. An agent's results refer only to the objects specified by the user.

Agent Calendar. Description of the period of time during which an agent evaluates the simulation algorithm →Workload Analysis.

Alignment Function. Functionality in the →Modelling Component, which allows you to align the →objects and →connectors of an →ADONIS model with the help of pre-defined or user-defined parameters. Available in the "Edit menu".

Analysis (component). →Component of the →Business Process Management Toolkit, in which →queries and reports can be run on your →models. (i.e. you can evaluate your →business process models. The queries are formulated in a language called →AQL (However, you do not need to understand AQL syntax in order to run analysis queries).

Analytical Evaluation (Calculation). Evaluation mechanism for →business process models in the →Analysis Component of the →Business Process Managament Toolkit. The →models are not simulated by this kind of evaluation but calculated mathematically.

Animation. Passive component in the simulation algorithms →"Workload Analysis (fixed time period)" and →"Workload Analysis (steady state)", which is responsible for graphically representing the →task stacks of the different →performers in the →Working Environment model being simulated. (Do not mix up with →offline animation).

APF. **A**DONIS **P**rotocol **F**ile; file format in which the results of a simulation run can be saved. APF files serve as input for the →offline animation function.

Application Library. Combination of a →Business Preocess and a →Working Environment library. The two →libraries contain the →classes and the →relations for the →models of the →model types defined. The application libraries stored in the →ADONIS database are displayed in the →application library list. Exactly one application library is assigned to each →ADONIS user.

Application Library List. List displaying all application libraries that are stored in the →ADONIS database. From here you can import, export and delete application libraries.

Application Model. Combination of at least one →business process model and exactly one →Working Environment model. The definition of an application model is necessary for the →Simulation Component to carry out the simulation algorithms "→Capacity Analysis" and "→Workload Analysis".

AQL. **A**DONIS **Q**uery **L**anguage; query language which enables you to →assign performers and →allocate resources in the →Modelling Component. You can also enter selection criteria for the functions "global change" and "model search". In addition, you can run →queries in the →Analysis Component and generate →pre-defined queries and →pre-defined evaluation queries when you edit the →library attributes in the →Library Management.

Assessment. Functionality in the →Modelling Component in the →Business Process Management Toolkit. This functionality enables you to evaluate →business process models according to time and cost criteria.

Attribute. Property of a →library, a →model, a →class, an →object, a →relation or a →connector. Attributes are usually filled with values (=attribute values) in ADONIS Notebooks.

Attribute-dependent Graphical Representation. Representation of →objects and →visualised attribute values on the →drawing area depending on concrete attribute values. An attribute-dependent graphical representation can be defined while →customising in the →Administration Toolkit within the →Library Management component. A Resource is an example of an attribute from the ADONIS BPMS Application Library that has an attribute-dependent graphical representation.

Attribute Filter. Filter function for the →Documentation Component. Using the attribute filter, you can select the attributes you want to transfer to the documentation (e.g. HTML, XML, RTF files).

Attribute Mode. Presetting in the application library for the →Documentation Component. An attribute mode defines, which attributes of a class should be transferred to the documentation (e.g. HTML, XML, RTF files).

Attribute Profile. Concrete instance of an attribute profile class. An attribute profile represents one or more →attributes, which are reused for any →objects and can be maintained in the →Attribute Profile Management).

Attribute Profile Class. Template, according to which the →attribute profiles are generated by →instancing. An attribute profile class or its →attribute profiles are described by a quantity of →attributes, which are assigned with concrete values in an →attribute profile.

Attribute Profile Group. Grouping, in which →attribute profiles are summed up.

Attribute Profile Management. →Component in the →Administration Toolkit, in which you can instance, edit or delete →attribute profiles as well as create, rename and delete →attribute profile groups.

Attribute Profile Reference. ATTRPROFREF; attribute type for referencing an attribute profile.

Attribute Value. Value of an →attribute, being a concrete instance of a characteristic.

Automatic Random Generator. →Random generator, automatic.

- B -

Balanced Scorecard (BSC). Instrument to translate company's mission and strategy to a clear system in order to measure the performance. This system creates the framework for a strategic performance measurement and management system. The Balanced Scorecard focuses above all on the financial targets, but contains also the business drivers of these financial targets.

Bendpoint. Point where a →connector changes its direction.

BMP. Microsoft Bitmap; graphic file format, in which diagrams/graphics which have been generated from →models can be saved.

BPMS Paradigm. (=Business Process Management Systems-Paradigm). Framework for the management of →business processes. This framework consists of five subprocesses: "Strategic Decision Process", "Re-engineering Process", "Resource Allocation Process", "Workflow Process" and "Performance Evaluation Process". It was developed by the BPMS group of the Institute for Applied Computer Science and Information Systems, Department of Knowledge Engineering, University of Vienna.

Browser. →ADONIS browser

Business Process. Number of logically connected →activities whose execution should meet a business objective.

Business Process Library. →Classes for →modelling business processes. The →classes of a business process library are derived from the classes of a →business process meta model. A business process library is part of an →application library. A business process model always refers to exactly one business process library. Business process libraries are defined while ADONIS is →customised

Business Process Management Toolkit. →ADONIS component in which the →ADONIS user works. In its standard configuration it consists of the following →components: →Acquisition, →Modelling, →Analysis, →Simulation, →Evaluation and →Import/Export.

Business Process Meta Model. →Classes pre-defined in ADONIS for →modelling →business processes. The →classes of a →business process library are derived from the →classes contained in the business process meta model.

Business Process Model. **1.** →ADONIS model of a business process. Comes into existence by →instantiation of →classes of a →business process library.

2. Model type of the →ADONIS BPMS BP Library.

- C -

Calculation. →Analytical Evaluation.

Calendar. 1. →Agent Calendar, →Performer Calendar, →Process Calendar.

2. CALENDAR; attribute type for defining the period of presence of performers (→Performer Calendar) or the period of occurrence (frequency) of business processes (→Process Calendar).

Capacity Analysis. Simulation algorithm in the →Simulation Component of the →Business Process Management Toolkit which "plays through" →business processes and assigns →activities to the →performers. An →application model serves as input for capacity analysis. This analysis enables you to plan your personnel capacity demand. The results of the capacity analysis are displayed in the →ADONIS browser.

Capacity Management (component). Special definition of the →Dynamic Evaluation Modules. Used for period-related calculation of the organisational/personnel capacity requirements.

case/4/0. Data and function-oriented CASE tool (manufacturer: microTOOL GmbH)

CASE. **C**omputer **A**ided **S**oftware **E**ngineering.

Class. Static template according to which →objects are created by →instantiation. A class or its →objects, are described by a set of →attributes, which are filled with concrete values within an →object. The class "Performer", for example, has among others the →attributes "Name", "Availability" and "Hourly wages".

Class Attribute. →Attribute of a →class, which refers directly to the class and does therefore not occur in →objects of this class such as the attribute "Info text" in an ADONIS modelling class (See: Information icon in Notebooks).

Class Filter. Filter function for the →Documentation Component. Using the class filter, you can select the →classes, in which →objects should be transferred to the documentation (e.g. HTML, XML, RTF files).

Class Panel. Button panel, which contains exactly one button for each →class that can be modelled. The class panel represents the second part of the →modelling bar - between the →edit button and the →relation panel.

Client. A client is a software module which requests services from a →server, usually via a network.

Client/Server. Computer system consisting of a →client and a →server.

Company Map. 1. ADONIS model containing an overview of all →business processes occurring in a company. Comes into existence by →instantiating →classes of a →business process library.

2. Model type of the →ADONIS BPMS BP Library.

Comparison of Results. Functionality of the →Evaluation Component in the →Business Process Management Toolkit. Compares results of analysis and simulation in tabular and graphical form. The results are displayed in the →ADONIS browser.

Component. Individual module of the →Administration Toolkit or of the →Business Process Management Toolkit (e.g. Modelling, Analysis, Model Management).

Component Access. Configuration of the access to the →components of the →Business Process Management Toolkit. The component access can be defined individually by the →ADONIS administrator for →ADONIS user groups and for each →ADONIS user, whereby the →ADONIS configuration can be customised

Component Bar. Panel, by default below the →menu bar in the →ADONIS window. The smart-icons visible show the available →components. To the right of the component bar, usually the →quick-access bar can be found.

Component Configuration. Functionality of the Component Management in the →Administration Toolkit. Additional ADONIS components can be included here by entering a new licence number.

Component Management. Component of the →Administration Toolkit, in which the components of the →Business Process Management Toolkit can be configured.

Configuration, current. The current configuration (→ADONIS configuration) specifies which functions of the components of the →Business Process Management Toolkit are available.

Connector. →Instantiation of a →relation. Connection/link between two →objects (e.g. "Subsequent").

Connector Attribute. →Attribute of a →connector.

Connector Mark. Visualisation of connection joints for connectors that extend over page boundaries. (See →connectors, →page layout). Connector marks are very useful in model documentation.

Connector Numbering. Assigning numerical or alphabetical identifiers to →connector marks. The connector numbering is non-ambiguous within a →model.

Context Menu. Context-depending menu. Appears when you press the right mouse-button. Provides functionality suitable for the particular situation.

Cooperative Activity. An →activity, which is executed by more than one →performer (such as a meeting).

Cost Cutting Component. Additional →ADONIS component, which can be integrated into the →Evaluation Component of the →Business Process Management Toolkit. The Cost Cutting Component is not part of ADONIS standard configurations. The Cost Cutting Component supports cost optimisation, especially in the area of overhead costs.

Cost Driver. Establishes the relation between costs, processes and calculation objects. The cost driver is a measure of the costs caused by the business processes and subprocesses.

Cost Driver Quantity. The measurable output of a →cost driver. Each →output-induced (OI) process has a cost driver quantity.

CSV. **C**omma **S**eparated **V**alue; text file format in which contents from the →ADONIS browser can be stored and transferred without loss to a spreadsheet software.

Cursor Key. Arrow keys on the keyboard, using which you can navigate to the left, the right, upward and downward.

Customer Number. An unambiguous alphanumerical identifier for each customer. The customer number must be entered when installing ADONIS.

Customising. Adapting ADONIS to a customer's demands without programming effort. Via customising, definition of the →page layout, the →model types, →pre-defined analysis queries and →plans as well as →pre-defined evaluation queries and simulation mechanisms is performed.

Cycle Time. Time that (on average) passes between the start of a →business process and its end.

- D -

Database Administration. ADONIS component, which the →database administrator uses to administrate the →ADONIS databases. The database administration contains the programs to create, store (only for →DB2 databases) and restore (only for →DB2 databases) →ADONIS databases.

Database Administrator. Person responsible for maintaining, configuring and administering the database system.

Day Profile. Template, which describes a →performer's time of presence or the stochastic occurrence (frequency) of a →business process per day. Day profiles are used in the →Agent Calendar, the →Performer Calendar and the →Process Calendar.

DB2. →DBMS for Windows, Linux, OS/2 and other operating systems (owner: IBM).

DBMS. **D**ata**B**ase **M**anagement **S**ystem.

Default Language. →ADONIS default language.

Depth Search. Process to go through objects, i.e. to automatically process the objects and relations of a model in the RTF generation in the →Documentation Component. Starting from a start object, all objects, which are directly connected to this start object, will be processed. Unlike the →width search the process will not be carried out gradually, but the relations of the objects will be followed until an object, which has no further relation, is reached.

Distribution. DISTRIBUTION; Attribute type for statistic distribution used for assigning variables.

Document Model. Model type of the →ADONIS BPMS BP Library to illustrate the documents used in →Business Process Models.

Documentation (component). Function in the →Import/Export Component of the →Business Process Management Toolkit. Using the Documentation Component, you can transport →ADONIS models to specific target formats (e.g. →RTF, →HTML, →XML), whereby you can distribute the model contents including graphic representation integrated into documents or via the →Intranet.

Domain. A domain is a group of computers as part of a network using a common directory database. A domain is organised in different levels and is administered as one unit with common rules and procedures. Every domain has a unique name. (Definition from "Windows Help".)

Drawing Area. Rectangle in the →model editor of the →Business Process Management Toolkit. The drawing area is at least big enough so that it encloses all →objects and →connectors of a →model. The drawing area is represented in the →modelling window by a grey outline. It is part of the workspace.

Drawing Mode. Mode of the →model editor in which →objects and →connectors can be drawn. You can switch from →editing mode to drawing mode by clicking on a button in the →modelling bar.

Drawing Space. Area of a →model window, where objects and connectors can be placed. The →drawing area is a part of the drawing space.

DSSSL. **D**ocument **S**tyle **S**emantics and **S**pecification **L**anguage (DSL file); DSL files are used in the →Documentation Component, to convert →SGML files to any format (e.g. →HTML, →RTF, →XML).

DTD. **D**ocument **T**ype **D**efinition; DTD are used to define the structure of documents (e.g. in the formats →XML, →SGML) .

Dynamic Evaluation Modules. Additional →ADONIS Component, which can be integrated to the →Evaluation Component of the →Business Process Management Toolkit. The component "Dynamic Evaluation Modules" is not included in ADONIS standard configurations. The Dynamic Evaluation Modules enable period-related evaluations (e.g. Human Resources Management).

- E -

E-Business. E-Business includes all business activities, i.e. the internal and external business processes of an organisation, which are supported by Internet technology.

Edge Centre. For each →connector the →ADONIS user can move the "centre". By moving the edge centre, the transition condition of a connector of type "Subsequent" may be arranged along this connector as you like. Alternatively, the edge centre may automatically be placed in the geometrical centre of the connector. This is useful to position the text displayed along a connector at a position at which it can be clearly read.

Edit Button. Topmost button within the →modelling bar. With this button the cursor in the →model editor can switch from →drawing mode ("pen") to →editing mode ("arrow").

Editing Mode. Mode in the →model editor, in which it is possible to edit (i.e. select, move, copy, cut, paste, delete etc) →objects and →connectors. You can switch from →drawing mode to editing mode either by clicking on the →editing button in the modelling bar or by right-clicking on the →drawing area in the →modelling window.

EMF. Windows **E**nhanced **M**eta**F**iles; Graphic file format, in which graphics, which have been generated from →models can be stored. EMF is based on vector graphics. This way EMF graphics can be scaled (resized) freely.

Empty Space Tool. Tool in the →Model editor that helps creating empty spaces in crowded models by moving the →objects. Part of the →Modelling bar.

Enumeration. ENUMERATION; attribute type with predefined attribute values. The valid value is exactly one of the set of values supplied.

Enumeration List. ENUMERATIONLIST; attribute type with predefined attribute values. The value valid may be zero, one or more elements of the value range defined.

Evaluation Agent. →Agent

Evaluation (component). →Component of the →Business Process Management Toolkit. Supports both the core "evaluation" activity in re-organisation projects and the "Performance Evaluation Process" within the →BPMS Paradigm.

Evaluation Queries. Functionality of the →Evaluation Component in the →Business Process Management Toolkit. Evaluation queries on →models serve to extract the model content (especially with regard to the simulation results). These can then be displayed in a clearly structured way in the →ADONIS browser. Depending on your particular →library configuration various pre-defined evaluation queries may be available.

Evaluation Queries, predefined. Predefined evaluation queries are defined by the →ADONIS administrator for →business process libraries and →working environment libraries. They are made available to the →ADONIS user by special menu options in the Evaluation Component of the →Business Process Management Toolkit. The results of predefined evaluation queries are displayed in the →ADONIS browser.

Execution Time. Time during which an →activity belonging to a →business process is executed.

Exit Button. Button in the top left of the →window panel, which closes the window when you double-click on it.

Export. →ADL-Export, →XML-Export.

Export Migration Assistant. Functionality in the →Administration Toolkit. The export migration assistant supports you in exporting →application libraries, →ADONIS users, →ADONIS user groups, →ADONIS models and →ADONIS model groups.

Expression. 1. EXPRESSION; attribute type which determines the value of an →attribute using a flexible calculation rule, whereas the attribute value is defined by other attributes.

2. →AQL expression.

Express Message. Specific form of a message (→ADONIS mail) to a logged in →ADONIS user. An express message will be immediately delivered to the receiver and shown in a popup window.

External Coupling. Coupling of an external tool to ADONIS.

- F -

FDL. **F**lowmark **D**efinition **L**anguage; language for describing →FlowMark models.

FDL Generation. Functionality of the →Import/Export Component in the →Business Process Management Toolkit. This functionality generates FDL files from →models (in order to then import them into →FlowMark).

Floating-Point Number. DOUBLE; attribute type for decimals (up to six decimal places).

FlowMark. →MQSeries Workflow.

Flowmark Audit Trail. Audit trail of →business processes carried out by →FlowMark.

- G -

Global Change. Simultaneous change of attribute values of several objects in one or more models. The attributes to be changed are selected by entering standardised or user-specific search criteria.

Global End. Terminates a →business process including those processes that call it (i.e. may appear within a subprocess). A global end represents the termination of the whole processing.

Groupware System. Computer system supporting teamwork by providing services like e-mail, document management, discussion forum and so on.

- H -

HOMER. →Acquisition Tables.

HTML. **H**yper**T**ext **M**arkup **L**anguage; file format, in which the contents of the →ADONIS browser or model content generated using the →Documentation Component can be saved. HTML files can be displayed in web browsers.

Human Resource Management (component). Specific definition of the →dynamical evaluation modules. Used for the period-related calculation (→time-related versioning) of the organisational/personnel capacity requirements.

- I -

Import Migration Assistant. Functionality in the →Administration Toolkit. The import migration assistant supports you in importing →application libraries, →ADONIS users, →ADONIS user groups, →ADONIS models and →ADONIS model groups.

Import/Export (component). →Component of the →Business Process Management Toolkit in which →ADL Import, →ADL Export and →FDL Transformation can be called.

Informix. →DBMS for Windows (owner: IBM).

Input Parameter Combination. Parameter for →simulation and analytical evaluation (calculation). When a Simulation or →Analytical Evaluation is started, an input parameter combination must be selected. This specifies on which →attributes the results should be calculated. Depending on the input

parameter combination different results will be calculated. Additional input parameter combinations can be defined in the →Administration Toolkit (→Library Management, →library attribute →"Simmapping").

Inspector. →ADONIS Inspector.

InstallShield. Software, using which you can install ADONIS via Windows (manufacturer: Macrovision Corp.).

Instance. 1. →Object

2. →Attribute Profile

3. →Process Instance.

Instantiation. 1. Creating an →object of a →class or a →connector of a →relation

2. Creating an →attribute profile of an →attribute profile class.

Integer. INTEGER; attribute type for an integer (whole number).

Inter-Model Reference. INTERREF; attribute type used to create a reference to another →ADONIS model or to an →object in a different →ADONIS model.

Internet. Worldwide computer network.

Intranet. Computer network, which is restricted to one organisation (company, department) - in contrast to the →Internet.

ISO9000. This umbrella term is commonly used for the series of norms ISO 9000 ff. These regulations of the International Standards Organisation (ISO) aim at creating a quality assurance system. They include a maximum number of twenty core elements. Depending on the type and size of a company's business operations, the company may acquire a certificate according to the norms ISO 9001, 9002 or 9003. An accredited certifier (= one recognised by the commission) awards this certificate. It certifies that the enterprise quality assurance system conforms with the core elements of the respective norm. The certificate according to ISO 9001-9003 is usually regarded as guarantee for high-quality products and/or services. It is a commonly applied criterion when selecting suppliers especially in the public services and industrial sector.

- J -

JPG. JPEG; graphic file format of the Joint Photographic Experts Group, in which graphics generated from →models can be saved.

- K -

Knowledge Management. Knowledge Management covers all activities of an organisation which deal with supplying and maintaining knowledge.

- L -

LEO. Meta-language for customising and specifying →attributes. Shows the contents of data structures in text form (→AdoScript).

LEOgram. Sequence of →LEO elements.

Library. Number of user-defined →classes for →modelling. Both →ADONIS BPMS BP Library and →ADONIS BPMS WE Library can be defined in ADONIS. Libraries are defined during ADONIS →customising.

Library Attributes. →Attributes of libraries.

Library Configuration. Functionality of the →Library Management in the →Administration Toolkit. Using the library configuration, you can edit and check class and library attributes (→customising).

Library Export. Export of an →application library into an →ABL file to transfer data into a different →ADONIS database or for backup purposes. This functionality of the →Library Management is located in the →Administration Toolkit.

Library Import. Import of an →application library from an →ABL file and then save it in the →ADONIS database. Functionality of the →Library Management in the →Administration Toolkit.

Library List. List that shows all the →business process and →Working Environment libraries stored in the →ADONIS database. From the library list you can import, export and delete libraries.

Library Management (component). →Component of the →Administration Toolkit, which is used to import and export →application libraries and to configure library attributes (= library configuration). The Library Management also contains an import and export migration assistant with the help of which the complete contents of the →ADONIS database (models, model groups, application libraries, users and user groups) can be exported (back-up) and imported into a different ADONIS database (data transfer).

Licence Number. Customer-specific number for configuring ADONIS and creating →ADONIS databases. A licence number consists of the encoded →customer number and the key for the available →components of the →Business Process Management Toolkit (depending on the customer-specific configuration of ADONIS).

Longstring. LONGSTRING; attribute type for text of up to 32.000 arbitrary symbols.

Lotus Notes. →Groupware and document management system (manufacturer: IBM).

LOVEM. **L**ine **o**f **V**isibility **E**ngineering **M**ethodology; business process modelling method of the IBM Corp.

- M -

Main Reference. Classification of →references to determine the referenced models (for instance when opening models including referenced models) outgoing from a model. Main referenced models will be processed like the start model, i.e. all references contained in this model will be followed, if the restriction of depth allows it. (see also →side reference).

Management Office. Software tool suite of BOC Asset Management GmbH. The Management Office consists of the Strategy and Performance Toolkit →ADOscore, the Business Process and Knowledge Management Toolkit →ADONIS, the Supply Chain Management Toolkit →ADOlog and the IT Service and Architecture Management Toolkit →ADOit.

Manual Random Generator. →Random generator, automatic.

Max. Length of Loops. Parameter of the →Analytical Evaluation. Determines how many →objects a loop (in a →model) may contain at most, before the evaluation is assumed as a never-ending loop.

Max. Number of Paths. Parameter of the →Analytical Evaluation. Determines how many different paths a model may contain before the evaluation is cancelled.

Max. Start Time Period. →Attribute of →objects of the →class "Activity". The maximum start time period specifies, for how long a →cooperative activity will at most be delayed until sufficient →performers are available for executing the cooperative →activity. This attribute is only evaluated by the simulation algorithm "→Workload Analysis".

Menu Bar. Panel between the →window panel and the →component bar/→quick-access bar in the →ADONIS window which contains the menu options for the activated →ADONIS component.

Messages. →ADONIS mail.

Meta Model. Describes the modelling classes of a model type and their dependences.

Method Library. →Application library.

Microsoft SQL Server. →DBMS for Windows (owner: Microsoft Corp.).

Migration. Transfer of an →application library, →ADONIS database or ADONIS version to a new or changed ADONIS application library, database or version.

Model. →ADONIS model.

Model Access. →Access right.

Model Attribute. →Attribute of a model.

Model Comparison. Comparison of two →models. Differences found will be displayed in the →ADONIS browser.

Model Editor. Graphically oriented editor in the →Modelling Component, used for creating →models.

Model Group. →ADONIS model group.

Model Group Management. Functionality of the →Model Management in the →Administration Toolkit. With the help of the model group management you can create, rename, move or delete →ADONIS model groups, create, copy, move or delete →model references as well as define →access rights of →ADONIS user groups to ADONIS model groups.

Model History. Supports the navigation between →models. For instance, you may switch between a main model and its sub model.

Model Management (component). →Component of the →Administration Toolkit, in which the →model access can be defined.

Model Pool. Quantity of all →ADONIS models saved in the →ADONIS database which are not assigned to any →ADONIS model group. Can be accessed via →model group management.

Model Reference. 1. Reference of an →ADONIS model group to an →ADONIS model in the →model group management. ADONIS models may be referenced in any number of ADONIS model groups.
2. →Inter-model reference (INTERREF) to an →ADONIS model, created in the →Modelling Component.

Model Search. Functionality of the →Modelling Component of the →Business Process Management Toolkit. Using the model search, you can search for models that are stored in the →ADONIS database. You can either search according to →model attributes or within the →process hierarchy.

Model Type. In ADONIS version 3.9 one or several model types can be defined in a →ADONIS BPMS BP Library and in a →ADONIS BPMS WE Library. In the →ADONIS BPMS Application Library we distinguish between →Business Process Models, →company maps (business process library) and →working environment models (working environment model).

Model Window. Window in which a →model is displayed and edited.

Modelling (component). →Component of the →Business Process Management Toolkit, in which →Business Process Models and →Working Environment models are created, using the →model editor.

Modelling, bottom-up. Approach to creating →models. When modelling bottom-up you start with detailed small models ("micro perspective") and gradually combine them to a bird's-eye view on the process in question by reducing their complexity (→modelling, top-down).

Modelling, top-down. Approach to creating →models. When modelling top-down you start with a bird's-eye view on a given process and gradually go into more detail ("micro perspective") (→Modelling, Bottom-Up).

Modelling Bar. Button panel, by default on the left-hand side of the →ADONIS window. The modelling bar consists of the →edit button, the →empty space tool, the →class panel and the →relation panel.

Mode. A mode is a (sub)set of the →objects and →connectors contained within an →ADONIS model. By selecting different modes (menu "View"), some extra objects may be made available and some existing objects may be hidden.

Mouse Access. It is possible to block mouse access for each →class and →relation during →modelling separately. This means that none of the →objects of that particular class or none of the →connectors of the particular relation can be selected.

MQ Series Workflow. →WebSphere MQ.

MSDE. **M**icro**s**oft **D**atabase **E**ngine; →DBMS for Windows (owner: Microsoft Corp.).

MS SQL Server. →DBMS for Windows (owner: Microsoft Corp.).

- N -

Navigator. →ADONIS Navigator.

Notebook. →ADONIS Notebook.

Notes. →Lotus Notes.

- O -

Object. Instantiation of a →class as for example "Specialist" could be an instantiation of the →class "Role".

Object Attribute. →Attribute of an →object.

Object Reference. Specific form of an →inter-model reference to an →object in the same or in a different →ADONIS model.

objectiF. Object-oriented →UML →CASE tool (owner: microTOOL GmbH).

Offline Animation. The offline animation enables you to exactly replay an earlier simulation run based on a →simulation protocol. During this, the performers' →task stacks are →animated. →Agents may also be employed during the offline animation. When generating the →simulation protocol the format "short version" must be specified in order to create a file (*.apf) that can be used by the offline animation function.

Online Manual. Electronic ADONIS manual, i.e. a manual that can be displayed on the screen by the help system. In ADONIS there are online manuals for the →Business Process Management-Toolkit, the →Administration Toolkit and the →ADONIS Database Administration.

Output-induced (OI) Process. Process causing costs that vary depending on the amount of work that must be executed by the cost centre. A →cost driver exists.

Output-neutral (ON) Process. Process causing costs which are independent of the amount of work that must be executed by the cost centre. No →cost driver exists.

Oracle. →DBMS for Unix, Windows and other operating systems (owner: Oracle Corp.).

- P -

Page Area. Visualisation of the →page layout in the →model editor of the →Business Process Management Toolkit. The page area is marked by a broken line. The page area is most important when graphic files are generated.

Page Layout. Description of the size of a printed page. The section available for the graphic model information and the headers and footers per page layout can be defined in the →library configuration of the →Administration Toolkit. The size of a printed page is especially taken into account when graphic files and →connector marks are generated.

Path. Sequence of objects (→activities, decisions etc), which describe one possible way in which a process can be executed.

Path Analysis. Simulation algorithm in the →Simulation Component of the →Business Process Management Toolkit, which "plays" through →Business Process Models. Serves to calculate process-specific key figures such as "Expected Cycle time" and "Expected Costs" and to determine which paths are critical with regard to various criteria selected by the →ADONIS user.

PCX. ZSoft Paintbrush graphic file format, in which graphics generated from →models can be saved.

PDF. **P**ortable **D**ocument **F**ormat; file format, using which documents can be shown independently of the hardware, the operating system and the application software used for the creation (owner: Adobe Systems Incorporated). The complete ADONIS user documentation is contained in PDF on the ADONIS installation CD.

Performer. Actor within a →business process. Performers are represented in ADONIS as →objects of the →class "Performer" in →Working Environment models. It is possible to specify the performer (or set of performers) who can execute a certain →activity using the "Performer" →attribute in an activity's notebook.

Performer Calendar. Description when a →performer is present and available for executing →activities. A performer's availability can be described in detail down to seconds and refers to one year. Performer calendars are evaluated in the simulation algorithm "→Workload Analysis".

Performer Assignment. In ADONIS, →business processes and →Working Environments are modelled separately. To assign →activities to performers, →AQL expressions are entered in the attribute "Performer" in objects of the class "Activity". These are interpreted during simulation to assign an actual →performer to an →activity.

Plans, pre-defined. Pre-defined plans are a specific type of result representation for →pre-defined queries. If a →query's result are all →objects of a →class within a →model - such as all →activities of a →business process model - the lines of the result table will only show the model name. The columns of the result table will be empty. Pre-defined plans may e.g. be used for generating task plans. They are defined by the →ADONIS administrator (during →library configuration) for →Working Environment libraries and →business process libraries) and are then available to the →ADONIS user, together with the →pre-defined queries in the →Analysis Component of the →Business Process Management Toolkit.

PNG. **P**ortable **N**etwork **G**raphics; Graphic file format, in which graphics generated from →models can be stored.

Pre-defined Evaluation Queries. →Queries, pre-defined.

Pre-defined Plans. →Plans, pre-defined.

Pre-defined Queries. →Queries, pre-defined.

Prioritisation. Process, using which the →ADONIS user can select the next →activity to be edited during the →simulation.

Process Calendar. Describes the stochastic occurrence of →business processes in time intervals. An example of stochastic occurrence is seasonal fluctuation. Process calendars are evaluated by the simulation algorithm "→Workload Analysis".

Process Hierarchy. Call hierarchy of →Business Process Models. A hierarchy may be structured according to →business processes calling or to →business processes called. Within a process hierarchy, models can be searched (→model search). The process hierarchy can be displayed graphically or in tabular form.

Process Instance. Specific instance of a →business process during the →simulation.

Program Call. PROGRAM CALL; attribute type for launching an external program (such as Microsoft Excel).

Program Icon. A program icon can be found in the →ADONIS program folder for every →ADONIS component and for every program of the →ADONIS database administration installed. A program is started by clicking on the particular program icon.

- Q -

Quality Assurance. The basic objective of quality assurance measures (or QA systems) is to guarantee a constant quality level with regard to a company's products and/or services. There is a fundamental difference between two different approaches of quality assurance: active vs. reactive methods. The latter attempts to assure the product quality by introducing control mechanisms. These, however, just fight the consequences of faulty processes, i.e. the poor/defective products. Newer approaches have tried to develop optimised procedures to actively affect the products' quality. These active quality assurance approaches are known as →Quality Management and find their expression in a great number of methods, concepts and canons of rules. These include Total Quality Management (TQM) in particular or the series of norms →ISO 9000 ff.

Quality Management. Quality Management is an umbrella term for a number of active quality assurance methods. Active QM approaches - in contrast to approaches of reactive →Quality Assurance - are those the purpose of which is to avoid "quality gaps" in the products and/or services of a company. An essential part of the quality management is thus the documentation and (re-)structuring of →business processes).

Quantity. →Attribute of →objects of the →class "Process start". Specifies how often the →business process occurs. The quantity refers to the value of the →attribute "Time Period". Both attributes are used in the →Capacity Analysis simulation algorithm.

Queries. Functionality of the →Analysis Component in the →Business Process Management Toolkit. By running queries on →models, the model contents can be extracted and be displayed in a clearly structured way in the →ADONIS browser. The following types of queries are available: →user-defined, →standardised and (depending on the particular →library configuration) →predefined ones.

Queries, predefined. The →ADONIS administrator defines predefined queries for →business process libraries and →Working Environment libraries in the →Library Management component of the →Administration Toolkit. The predefined queries are made available to the →ADONIS user in the →Analysis Component of the →Business Process Management Toolkit by specific menu options. The results of the predefined queries are displayed in the →ADONIS browser.

Queries, standardised. Standardised queries are available to the →ADONIS user in the →Analysis Component in the →Business Process Management Toolkit. Standardised queries may be used to create →user-defined queries. The results of the standardised queries are displayed in the →ADONIS browser.

Queries, user-defined. The →ADONIS user defines user-defined queries in the →Analysis Component of the →Business Process Management Toolkit by using →AQL expressions. The results of the user-defined queries are displayed in the →ADONIS browser.

Quick-access Bar. Panel, by default below the →menu bar in the →ADONIS window. The buttons (smart-icons) visible depend on the →component selected. They provide quick access to certain functions of the →component. To the left of the quick-access bar, usually the →component bar can be found.

- R -

R/3. Standard software suite (manufacturer: SAP).

Random Generator, automatic. The assignment of variables during the →simulation run is automatically using the defined →distributions.

Random Generator, manual. The ADONIS user can select the variable assignment during the →simulation run.

Rational Rose. Object-oriented →CASE tool (owner: IBM).

RDBMS. Relational database management system. →DBMS, which is based on the relational data model.

Read Access. →Access right to an →ADONIS model group and the →ADONIS models referenced within this group. With this type of access right, you may view models **locally** but do not have access to make any changes.

Record. RECORD; attribute type which enables a flexible list/record administration of assembled file types (→attributes).

Reference Model. Consists of standardised reference processes (templates) that represent operational activities generally valid or necessary. Reference models focus on branch or software-specific aspects and can be adapted (customised) to company-specific circumstances.

Reference Process Model. Process from a →reference model.

Relation. General connection between two →classes. →Objects of a →class can be linked by concrete →instantiations) of this →connection), for example, two organisational units may be linked by the relation "Is subordinated".

Relation Attribute. →Attribute of a →relation.

Relation Panel. Button panel that contains exactly one button for each relation that can be modelled in the active context. It is the lowest part of the →modelling bar, following the →edit button and the →class panel.

Relation Table. Functionality available in the →Analysis Component of the →Business Process Management Toolkit. A relation table represents →connections (of a relation) inside a →model. The →objects (start and target objects), which are linked by the →connectors, are displayed in the row or column headers of the table. Which relation tables are available depends on the respective →library configuration.

Repository. Central database for the administration of all →ADONIS users, →ADONIS application libraries, →ADONIS models etc.

Representation, graphical. Possibility of visualising an ADONIS model in graphical form.

Representation, tabular. Possibility of visualising an ADONIS model in tabular form.

Resource. Part of a →Working Environment which is needed for executing →activities, such as PC, printer or fax.

Resource Assignment. In ADONIS →business processes and →Working Environments are modelled separately from each other. For the →Simulation to take account of →resources, →AQL expression(s) must be entered in the "Selection" attribute of →objects of the →class "Resource" in business processes (support available for this). These are interpreted during the →Capacity and →Workload Analysis s. In this way, the resources in the →Working Environment which are needed to execute an →activity are determined.

Resting Time. Time during which an →activity of a →business process has been executed but has not yet moved on to be transported to the next stage in the process.

Result Function. Function which is assigned to an →agent. By this, the agent is instructed to calculate a certain simulation result.

RTF. **R**ich **T**ext **F**ormat; file format, in which the contents of the →ADONIS browser or model contents which were generated using the →Documentation Component can be saved. RTF files can be further edited in word-processing programs.

- S -

SAP. **S**ystems, **A**pplications, **P**roducts in data processing. Owner of the standard software →R/3.

Server. A server is a software module which puts its services at the disposal of other software modules, the →clients.

SGML. **S**tandard **G**eneralized **M**arkup **L**anguage; SGML files are used in the →Documentation Component to transfer the documented model contents. In a second phase, the SGML files will be then converted to any target format (e.g. →HTML, →RTF, →XML) using a →DSL file.

Side Reference. Classification of →references to determine the referenced models (for example when opening models including referenced models) starting from a model. Outgoing references from side-referenced models will only be followed, if these references start from the same attribute and if no depth restrictions prevent it.

Sim Mapping. Attribute of a →business process library. In the sim mapping the →input parameter combinations are defined. In addition, the sim mapping defines which →classes should be to totalled for aggregated results, for the →assignment of performers and for the reference objects of the →agents.

Simulation (component). →Component in the →Business Process Management Toolkit, which simulates →business processes and →Working Environments. ADONIS contains the simulation algorithms "→Path Analysis", "→Capacity Analysis", "→Workload aAnalysis (steady state)" and "→Workload Analysis (fixed time period)".

Simulation Cache. The simulation cache stores the simulated →ADONIS models in the internal representation necessary for →simulation. Thus, the models need not be transferred into the internal representation again, if you start the simulation anew.

Simulation Log. Report (protocol) of a simulation run in the Simulation Component.

Simulation Log, short version. Compressed form of the →simulation log; used by the →offline animation function.

Simulation Log, long version. Detailed form of the →simulation log for replaying each step of the →simulation.

Simulation Run. Processing of a →process instance during the →Simulation.

Single Sign-on (SSO). Functionality in ADONIS which enables that the Windows system user can be imported into the ADONIS user management and therefore allow →system users working and logged in to the →domain to log into ADONIS without providing user name and password again.

Smart-Icon. Graphical buttons in the →component bar, the →quick-access bar etc.

Snap Grid. Grid according to which →objects and →bendpoints can be aligned on the →drawing area.

SQL. **S**tructured **Q**uery **L**anguage. Query language for relational databases.

SQL Server. →Microsoft SQL Server.

Staffware. →Workflow Management System (owner: TIBCO, Inc.).

Stand-alone. A single computer system.

Standard Model Group. →ADONIS model group, which is created automatically with the →ADONIS database. The standard model group has the name "Models".

Standard User. →ADONIS user who is created automatically together with the →ADONIS database. The standard user has the user name "Admin" (default password is "password") and cannot be deleted.

Standard User Group. →ADONIS user group, which is created automatically with an →ADONIS database. The standard user group has the name "ADONIS" and cannot be deleted.

Standardised Queries. →Queries, standardised.

Swimlane. A swimlane is a modelling construct to describe roles, organisational units etc. A swimlane is visualised by vertical and horizontal areas on the drawing area. The characteristic of a swim lane is that when changing its size you will also move the →objects and →relations placed on this swimlane.

System Administrator. Person responsible for the IT infrastructure (hardware, network, software installations and so on).

System User. User of the operating system (Windows) who is authenticated by logging into a →domain. In general the system has to be created by a →system administrator within the system user administration in order to be able to work with and log in to the system. System users can log in to ADONIS via →single sign-on (SSO), if this feature is provided.

System User Administration. Part of an operating system used to create, delete or change the settings of →system users, create, rename and delete →system user groups, as well as assign system users to system user groups.

System User Group. A group containing →system users. A system user group has specific access and execution rights to components of the operating system which are defined by the →system administrator.

- T -

Tabular Representation of a Model. →Representation, tabular.

Task Stack. Stack of →activities a →performer has to execute. The size of the task stack is determined in the simulation algorithm →"Workload Analysis".

Task stack, central. The activities in the central task stack will be executed by the first performer who has time for it (→prioritisation).

Task Stack, personal. The activities in the personal task stack of a performer are exclusively executed by this performer.

Text. STRING; attribute type for text containing up to 3.700 arbitrary symbols (exception: attribute "Name" which is restricted to at most 240 symbols).

Time. TIME; attribute type for the ADONIS time format YY:DDD:HH:MM:SS (Years:Days:Hours:Minutes:Seconds).

Time, company. Division of time referring to the presence of a →performer in the company. A (performer's) company time may for example consists of 170 days a year and 8 hours a day.

Time, real. Conventional division of time, i.e. a year has 365 days and a day has 24 hours.

Time Period. →Attribute of →objects of the →class "Process start". The time period specifies for which period of time the value of the →attribute "→Quantity" is valid. (Used in capacity analysis).

Tolerance Waiting Time. Maximum waiting time before a →process instance is aborted unedited during the →simulation.

Transformation. →FDL-Generation.

Transport Time. Time needed for transporting files, documents etc.

TXT. ASCII-TEXT; file format in which ordinary text files are saved.

- U -

UDL. **U**ser **D**efinition **L**anguage; file format in which →ADONIS users and →ADONIS user groups are saved (exported) from the →Administration Toolkit.

UDL Export. Functionality of the →User Management in the →Administration Toolkit. With UDL export, →ADONIS users and →ADONIS user groups can be exported into →UDL files.

UDL Import. Functionality of the →User Management in the →Administration Toolkit. With UDL import →UDL files (→ADONIS users and →ADONIS user groups) can be imported into ADONIS and then stored in the →ADONIS database.

UML. **U**nified **M**odeling **L**anguage; method of object-oriented modelling of complex software systems.

Use Case Diagram. 1. →UML diagram type. Interactions between a system and external people involved are described in Use Case Diagrams.

2. Model type of the →ADONIS BPMS BP Library.

User. →ADONIS user.

User Account. User name and password for logging into the →DBMS or the operating system.

User Assignment. Assignment of an →ADONIS user to one or more →ADONIS user group(s) in the →User Management of the →Administration Toolkit. This assignment makes it possible for the user to access →ADONIS model groups and →ADONIS models within the Business Process Management Toolkit.

User Export. Export of →ADONIS users and/or →ADONIS user groups into a →UDL file in order to transfer the data into a different →ADONIS database or make a backup. Functionality of the →User Management in the →Administration Toolkit.

User Group. →ADONIS user group.

User Group List. List of all →ADONIS user groups stored in the →ADONIS database as well as all →ADONIS users referenced in them. With the help of the user list, ADONIS user groups can be added, renamed and deleted. →user assignment can also be edited.

User Import. Import of →ADONIS users and/or →ADONIS user groups from a →UDL file into the →ADONIS database. Functionality of the →User Management in the →Administration Toolkit.

User List. List of all →ADONIS users stored in the →ADONIS database. With the help of the user list, you can add and delete →ADONIS users as well as edit their settings.

User Management (component). →ADONIS User Mangement.

User-defined Queries. →Queries, user-defined.

- V -

Validity Date. Specifies the time from which a model or attribute profile version becomes valid. The validity of a version ends with the validity start of the next version of the same model or attribute profile.

Versioning. Enables the administration of different versions of a →model. in ADONIS, either a →model-related or a →time-related versioning is possible.

Versioning, model-related. A version number is assigned manually to each version of a →model.

Versioning, time-related. A →validity date is assigned to each version of a →model.

Visio. Graphical modelling tool (Owner: Microsoft Corp.).

Visualised Attribute Value. The →attribute value of an →object or a →connector which is displayed on the →drawing area (g.g. Name of an "Activity", transition condition of a "Subsequent" relation).

- W -

Waiting Time. Time during which an →activity of a →business process cannot yet be executed, as the person due to execute it is still busy working on something else; i.e. the activity is ready to be executed but must first queue for a while.

WebSphere MQ. →Workflow Management System, successor of FlowMark and MQ Series Workflow (owner: IBM).

Width Search. Process to go through objects, i.e. to automatically process the objects and relations of a model in the RTF generation in the →Documentation Component. Starting from a "Process start" object, all objects which are directly connected to this start object, will be processed gradually. When all objects which are directly connected to this start object have been processed, one of these directly connected objects will be selected as a new start object and the process will recur. This will be carried out for all objects. (see also →depth search)

Window Panel. Title bar of an →ADONIS component's window (→ADONIS window).

Windows System User. →System user.

Workflow Management System (WMS). Computer system that controls the execution of →business processes. The most prominent task of a Workflow management system is to delegate business process activities to the organisation's employees.

Working Environment. (= organisational structures); structures (e.g. departments and roles), →performers and →resources of an organisation. The working environment of an organisation is represented in ADONIS in →Working Environment models.

Working Environment Library. Contains →classes and →relations for →modelling the →Working Environment as well as description, layout, analysis and simulation definitions. The →classes of a Working Environment library are derived from the classes of the Working Environment meta model. A →Working Environment model always refers to exactly one Working Environment library. Working Environment libraries are defined during ADONIS →customising.

Working Environment Meta Model. Pre-defined →classes for →modelling Working Environments in ADONIS. The classes in a Working Environment library are derived from the classes contained in the Working Environment meta model.

Working Environment Model. 1. →Model of a Working Environment. Results from instantiating →classes of a →Working Environment library.

2. Model type of the →ADONIS BPMS WE Library.

Workspace (Working Area). Main part of the ADONIS window, usually situated below the →component bar/→quick-access bar. The →modelling windows are displayed in the workspace.

Workload Analysis. Simulation algorithm in the →Simulation Component of the →Business Process Management Toolkit, based on queuing theory. The simulation runs "on the time axis". The results are the number of →cycle times and →waiting times depending on the capacity/amount of work of →performers and →resources. The input for the Workload Analysis is an →application model. A Workload Analysis can be run based on a →steady state or a →fixed time period basis. The results of the Workload Analysis are displayed in the →ADONIS browser.

Workload Analysis, fixed-time period. →Workload Analysis, in which the length of the initialisation phase and the calculation (result measuring) phase and the types of results calculated (input parameters) are defined by the user.

Workload Analysis, steady-state view. →Workload Analysis, in which the length of the oscillation phases and the calculation period depend on the number of simulation runs.

Write Access. →Access right granted to an ADONIS user group to an →ADONIS model group and the →ADONIS models referenced within it. With this access right, you can locally change as well as save a model's content.

- X -

XML. E**X**tensible **M**arkup **L**anguage; file format in which →ADONIS models can be stored (exported).

XML Export. Functionality in the →Import/Export Component of the →Business Process Management Toolkit. Using the XML export, →ADONIS models can be exported to →XML files.

XML Import. Functionality in the →Import/Export Component of the →Business Process Management Toolkit. Using the XML import,→XML files (→ADONIS models) can be imported to ADONIS and stored in the →ADONIS database.

23. Index

B

ADONIS Administration Toolkit

ADONIS Administration Toolkit

ADONIS Administration Toolkit

                                                    ADONIS Administration Toolkit

ADONIS Administration Toolkit

Q

R

V

ADONIS®
Business Process Management

ADOscore®
Strategy Management

ADOlog®
Supply Chain Management

ADOit®
IT Management

www.boc-eu.com