

ActiveEntry
now you can.

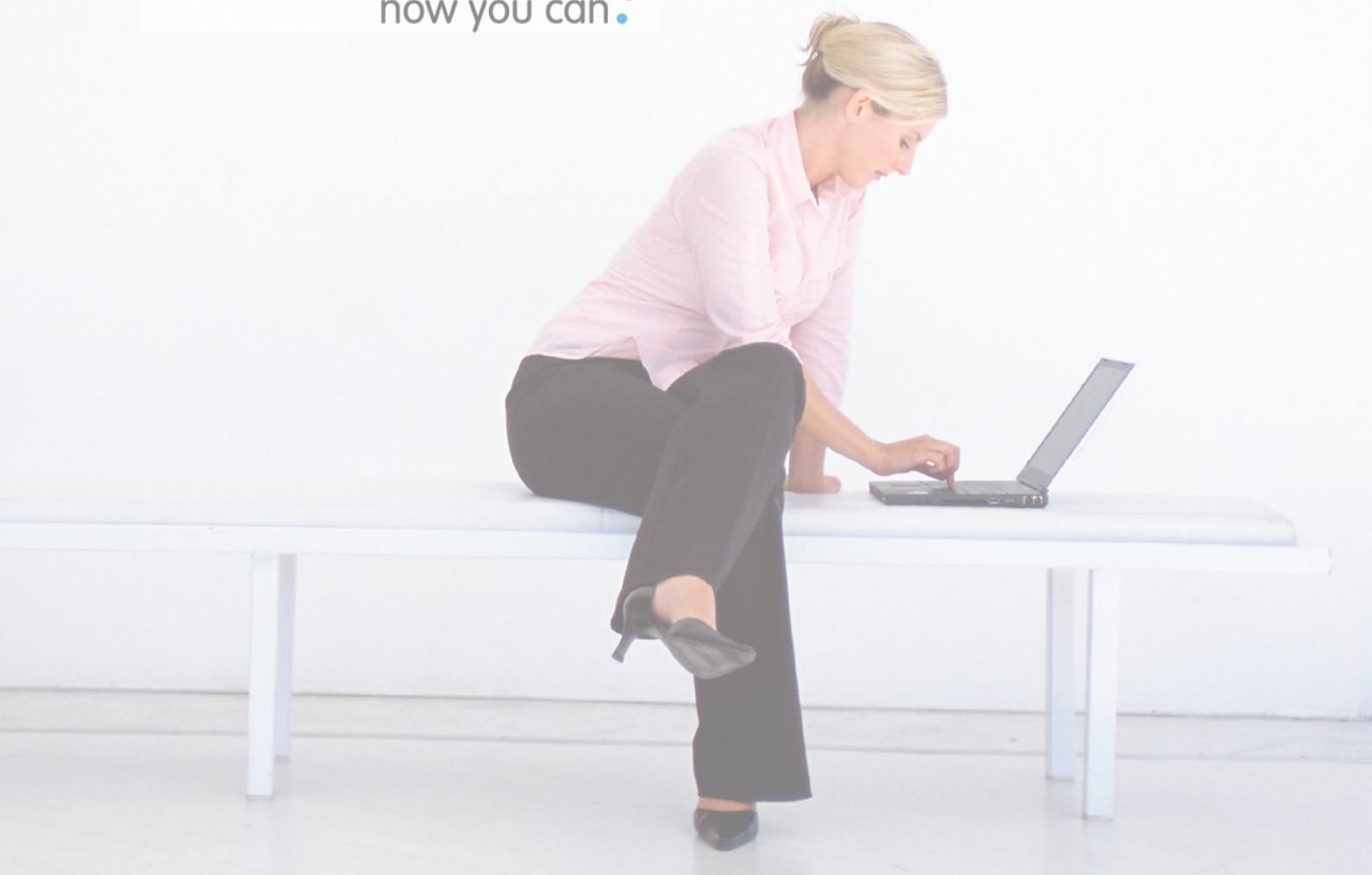


TABLE OF CONTENT - GETTING STARTED

Chapter 1	General Advice for ActiveEntry Documentation	9
Chapter 2	Use of Open Source Licenses	13
Part I	Installation	
Chapter 3	Installation Requirements	17
3.1	Installation Requirements for the ActiveEntry Database	17
3.1.1	Microsoft SQL Server Database System	17
3.1.2	Database User under Microsoft SQL Server	18
3.1.3	Oracle Database System	19
3.1.4	Database User under Oracle	19
3.2	Installation Requirements for ActiveEntry Tools on an Administrative Workstation 20	
3.3	Installation Requirements for ActiveEntry Service on a Server	21
Chapter 4	Installing ActiveEntry Tools	23
4.1	Updating ActiveEntry	23
4.2	ActiveEntry.Net Setup Wizard	24
4.2.1	Microsoft Windows 2000/2003 Terminal Server Installation	24
4.3	Setting Up an Administration Workstation	25
4.4	Migrating an ActiveEntry Database	27
4.4.1	Database Migration under Microsoft SQL Server	27
4.4.2	Preparing a Workstation for Migration	27
4.4.3	Running a Migration with the Program DBMigrator	29
	Database Login	30
	Database Migration	31
	Database Compilation	34
	Updating the Files in the Database	37
4.4.4	Setting Up Database Schedules	39
4.4.5	Posting and Sharing ActiveEntry IT Shop	40
4.5	ActiveEntry Licensing	41
4.5.1	Creating a License Request	41
4.5.2	Installing a License	45
4.6	Setting Up a Server for Database Access	48
4.6.1	ActiveEntry Service's Mode of Operation	48
	ActiveEntry Service Job Providers	49
	ActiveEntry Service Job Destinations	50
	ActiveEntry Service Job Gates	50

4.6.2	Configuring ActiveEntry Service	51
4.6.3	Working with ActiveEntry Service Configurator.....	52
	ActiveEntry Service Configuration Modules	53
	Selecting the Module Types	54
	Template for the Configuration File	55
	Configuration File Verification Test	55
4.6.4	Process Collection Module	55
	MSSQLJobProvider	56
	OracleJobProvider.....	57
	FileJobProvider	58
	FTPJobProvider.....	60
	HTTPJobProvider	61
	WebserviceJobProvider	61
4.6.5	Jobdestination Module	63
	JobServiceDestination	64
	FileJobDestination	66
	FTPJobDestination.....	67
	HTTPJobDestination	68
4.6.6	The Configuration Module	69
4.6.7	The Log Writer Module.....	70
	EventLogLogWriter.....	71
	FileLogWriter	72
4.6.8	The Dispatcher Module.....	73
4.6.9	The Connection Module.....	75
4.6.10	The Plug-in Module	76
	HTTPStatusPlugin	76
	HTTPLogPlugin	77
	StatisticsPlugin.....	78
	ScheduleCommandPlugin.....	79
	DBSchedulerWatchDogPlugin	80
	RequestWatchDogPlugin.....	81
	PerformanceCounterPlugin.....	82
4.6.11	ActiveEntry Service Installation and Startup	83
	Installing and Uninstalling ActiveEntry Service from the Command Line	83
4.6.12	ActiveEntry Service in a Cluster.....	84
	Registering ActiveEntry Service in a Cluster.....	85
	ActiveEntry Service Installation and Cluster Configuration	85
	Setting up the ActiveEntry Service Cluster Resource.....	87
	Storing the ActiveEntry Service Log file on a Shared Volume	90
4.7	Updating ActiveEntry Tools.....	94
4.7.1	Updating the Database	95
	Importing a Transport Package	97
4.7.2	Transferring Custom Data	100
	Creating a Customer Configuration Package	101
	Transport by Change Label	104

	Transport by Change Information	105
	Transporting Schema Extensions	107
	Transporting Selected Objects and their Dependencies.....	108
	System Configuration Transport	109
	System File Transport	110
4.7.3	Automatic Software Updating.....	110
	Automatic Updating of ActiveEntry Tools	111
	Automatic Job Server Updating	112
4.7.4	Updating Individual Job Servers.....	114
4.7.5	Importing New Files into the ActiveEntry Database	116
4.7.6	Exporting Files from the ActiveEntry Database	122
4.8	Compiling the ActiveEntry Database	126
4.9	Database Information Encrytion.....	135
4.10	Setting Up an ActiveEntry Database for Full-Text Search.....	141
4.10.1	Labeling Database Columns for Full-Text Search	142
4.10.2	Setting Up a Full-Text Catalog.....	143
4.11	Setting up a Reference Database	143

Part II Basics

Chapter 5	ActiveEntry Tools	151
5.1	ActiveEntry Identity Manager	151
5.2	ActiveEntry IT Shop	152
5.3	ActiveEntry Manager	152
5.4	DBMigrator	153
5.5	ActiveEntry Transporter	153
5.6	DBCompiler.....	153
5.7	Crypto Configuration	154
5.8	ActiveEntry Service Configurator	154
5.9	ActiveEntry Service Updater	154
5.10	Job Queue Info	154
5.11	Schema Extension	155
5.12	ActiveEntry Designer	155
5.13	License Wizard.....	155
5.14	Software Loader	156
5.15	ActiveEntry Data Import.....	156
5.16	Report Editor	156
5.17	Script Debugger	157
5.18	ActiveEntry Analyzer	157
5.19	Web Designer	157
5.20	HistoryDB Manager	157
Chapter 6	Logging into ActiveEntry Tools	159

6.1	Logging into the Database with a Database User.....	159
6.2	Logging into ActiveEntry Administration Tool as System User	161
6.2.1	Authentication Module.....	162
	System User	162
	Component Authenticator.....	162
	Employee	163
	Employee (dynamic).....	163
	Employee (Role Based).....	163
	ADS User	164
	Web ADS User.....	164
	ADS User (dynamic).....	164
	ADS User (manual input).....	165
	ADS User (role based).....	165
	ADS User (manual input/role based)	165
	LDAP User (dynamic).....	166
6.2.2	Login	166
Chapter 7	ActiveEntry Tools User Interface	169
7.1	The User Interface Layout	169
7.1.1	Title Bar	170
7.1.2	Status Bar	170
7.1.3	Menu Bar	171
7.1.4	Toolbars	175
7.1.5	Context Menus	176
7.1.6	Structure of the Edit Interface	176
7.1.7	Selecting the Display Mode	177
7.1.8	Searching for List Entries.....	180
7.1.9	Limiting the Number of Results	180
7.1.10	Changing the Password for the Current User.....	182
7.1.11	Modifying the Program Settings	183
	User Settings	183
	Special Settings for Individual Program Components.....	185
	General Settings	186
	Available Plugins	187
7.2	Functions in the Navigation View.....	188
7.3	Functions in the Result List	190
7.3.1	Special Consistency Icons	193
7.4	Functions in the Document View.....	193
7.4.1	Getting Form Help	194
7.4.2	Control Elements Used on Forms.....	195
	Form Elements	195
	Input Fields	196
	Script Input Field	197
	Permitted Value List Input Field.....	198

	Multiple Value Properties Input Field	198
	Language Dependent Input Field	198
	Pop-Up Menus	199
	Extended Pop-Up Menus	200
	Check Boxes	200
	Radio Buttons	200
	Tables.....	201
	Buttons	201
	Tabs	201
	Setting the Date.....	202
	Editing Login Times	203
	Displaying Object Relations	203
	Displaying Assignments in a Member Tree	205
	Entering Database Queries	206
	Wizard for Entering Database Queries.....	207
	Rule Editor for Entering Rule Conditions	208
	Overview of All Assignments	209
7.4.3	Multiple Object Edit	210
7.4.4	Reusing Value Templates	211
7.5	Functions in the Tasks View	212
7.6	Functions in the Favorites List	213
7.7	Logging Error Messages.....	214
7.7.1	Error Message Window	214
7.7.2	Displaying Error and System Logs	214
7.7.3	Error Log File.....	216
7.8	Evaluating the Process Monitoring Information.....	217
7.8.1	Standard Functions in the Process View	218
	Process Information Form Functions	218
	Functions in the Log	219
	Configuring the Process Display	221
7.8.2	Displaying Process Information	222
7.8.3	Displaying the Change History.....	224
7.9	Analysis of Historical Data in TimeTrace	225
7.9.1	Functions in the TimeTrace View.....	226
7.9.2	Displaying Change Information	226
7.10	Working in Simulation Mode.....	228
7.10.1	Simulation Data Overview	230
	Analyzing the DBQueue.....	230
	Analyzing Generated Processes	231
	Analyzing Trigger Changes.....	232
	Analyzing Modified Objects	233
	Analyzing the Rules.....	234
7.11	Planing the Execution Times of Operations.....	235
7.11.1	Specifying the Execution Time Schedule.....	235
7.11.2	Displaying Scheduled Operations	236

7.12	Implementing User Defined Filters for Database Searches.....	238
7.12.1	Setting Up User Defined Filters	238
	Searching with Wildcards	240
	Searching with Conditions	240
	Searching with Full-Text.....	240
7.13	Exporting Data	242
7.13.1	Creating an Export	242
	Selecting a Base Table and Columns for Export.....	242
	Changing the Column Order	243
	Limiting Export Data Sets	243
	Previewing and Exporting the Data.....	244
7.13.2	Saving and Loading Export Definitions	244
7.14	Information about DBScheduler Calculation Tasks	245

Chapter 1 General Advice for ActiveEntry Documentation

ActiveEntry documentation includes the following manuals as well as the Getting Started Manual. They can be found on the distribution CD in the directory ...\\ActiveEntry.NET\\Documentation.

Getting Started

Test changes The main components of the Getting Started Manual are:

- Installation prerequisites
- Installation and updates of ActiveEntry administration tools
- ActiveEntry database setup
- Configuration of administration workdesks
- Configuration of server for accessing the database
- Overview of ActiveEntry administration and configuration tools
- Interface for the main ActiveEntry tools

Identity Management

The main components of the Identity Management Manual are:

- Identity Management and User Provisioning with ActiveEntry
- Complying to and monitoring regulatory requirements using Identity Audit

Process Orchestration

The main components of the Process Orchestration Manual are:

- Monitoring process handling
- Controlling process handling
- Troubleshooting

Service Management

The main components of the Service Management Manual are:

- Administration of hardware, software and licenses within one ActiveEntry network
- Software distribution with the VI Client
- Monitoring software profile replication
- Service accounting for available resources
- Troubleshooting with the help desk module

Configuration

The main components of the Configuration manual are:

- ActiveEntry software architecture
- Configuration of ActiveEntry data models
- Access permissions configuration
- User interface configuration
- Script processing
- Creating reports
- Data transport
- System configuration parameters
- ActiveEntry inheritance mechanism
- Service provisioning using Service Provisioning Markup Language (SPML)
- Provider mode

IT Shop

The main components of the IT Shop manual are:

- IT Shop for authorized employees to supply themselves with company resources.
- Development of approval policies and workflows

Legal Disclaimer

This book and the program material are protected by copyright. Any usage outside the bounds of copyright is not permitted without the prior agreement of Völcker Informatik. This applies in particular to reproductions, translations, installation and processing in electronic systems. Völcker Informatik does not accept any responsibility, liability or guarantee for the contents of the text or the correctness of the program material. Völcker Informatik disclaims any liability that arises in any way from the use of the program material or the text.

The software and hardware products mentioned in this manual are, in most cases, registered products and as such are subject to legal regulations.

Produced by:
Völcker Informatik AG Berlin/Dresden | Quest Software
(now including the people and products of Völcker)

Fasanenstraße 33
10719 Berlin

Registered court authority: district court Charlottenburg, HRB 70508
Executive board: Eckhard Völcker (CEO), Gordon Patzschke
Supervisory board: Thomas Patterson (president)

copyright: 11.10.10. www.activeentry.com | www.quest.com

Chapter 2 Use of Open Source Licenses

Open source software is implemented in several ActiveEntry program functions. The following list provides you with an overview of these products and the current license agreements.

MochiKit

© 2005 Bob Ippolito

License Agreement: MochiKit License.txt (in install directory)

SharpZipLib

© 2001 Mike Krueger

License Agreement: <http://www.quest.com/legal/third-party-licenses.aspx>

Source Code: <http://rc.quest.com> und <http://www.icsharpcode.net/opensource/sharpziplib/>

plink

© 1997-2007 Simon Tatham

License Agreement: <http://rc.quest.com/topics/putty/>

Novell.directory.LDAP

© 2003 Novell Inc.

License Agreement:

The MIT License
Copyright (c) 2003 Novell Inc. www.novell.com

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the Software), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED AS IS, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Mono.Security

© Novell Inc.

License Agreement:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Part I **Installation**

This section of the Getting Started Manual describes the basics for working with ActiveEntry .

The chapters provide an overview of the functionality of ActiveEntry administration and configuration tools. Furthermore, you are given a description of how to install ActiveEntry, setup and migrate an ActiveEntry database and how you can update ActiveEntry tools.

Chapter 3 Installation Requirements

The installation prerequisites described in the following chapter provide the minimum requirements for putting ActiveEntry into operation and for unrestricted use of ActiveEntry. If other system requirements are necessary for individual ActiveEntry models, they will be listed in the corresponding chapter in this manual.

3.1 Installation Requirements for the ActiveEntry Database

The database is the nerve center of ActiveEntry. Information such as employee properties, information about user accounts and organizational data are stored in the database. ActiveEntry configuration data, for example, access permissions data, workflow definitions, parameters for influencing and controlling system behavior and data for customizing ActiveEntry administration tools to suit the user's needs are also stored in the database.

ActiveEntry works together with Microsoft SQL Server and Oracle database systems.

3.1.1 Microsoft SQL Server Database System

You need to guarantee the following system prerequisites in order to install the ActiveEntry database:

- An installed and configured database server with the default sort schema (case-insensitive)
Supported versions are:
 - Microsoft SQL Server 2005 with at least Service Pack 3 (Version 9.0.4035)
- SQL Server Management Studio (recommended)
- Always select the default language "English" for SQL server users, even for the German SQL server! You also need to select "English" as the language for the database users!
- Setting up the initial database
The database has to be set up with an SQL server database user. This means that the login on the SQL server has to use SQL server authentication (not with Windows authentication!).
- SQL Server Agent
The SQL Server Agent has to be started under a domain user with Windows authentication.

3.1.2 Database User under Microsoft SQL Server

The database should be setup and used under the SQL server systems administrator account (sa account). If the 'sa' account cannot be used, the database users have to fulfill the following minimum requirements:

- Default language "English"

Permissions to set up a database:

- Server role "dbcreator"

Permissions to migrate the database:

- Database role "db_owner" for the ActiveEntry database
- Database role "SQLAgentUserRole" and "db_owner" for the msdb database
- Database role "db_owner" for the tempdb database

Permissions for live operations:

- Database roles "db_owner" and "basegroup" for the ActiveEntry database
The database role "basegroup" is added by default during initial migration of the ActiveEntry database.
- Database roles "db_Datareader", "db_Datawriter" and "SQLAgentOperatorRole" for the msdb database
- Database role "db_Datareader" for the master database
- Database role "db_owner" for the tempdb database
- Server role "processadmin"
- Server Permissions "VIEW SERVER STATE"

NOTE!

If the database user's account is not changed until after migration, then the new database user has to be entered as owner of the schedule at a later date. Otherwise, it can result in errors when running the database schedule.

3.1.3 Oracle Database System

The following system prerequisites need to be guaranteed to install the ActiveEntry database:

- Installed and configured database server with Oracle Database 10g Enterprise Edition from Version 10.2.0.3 on (at least patch 6867056) with the option "Oracle Text"
- Oracle Client Tools from version 10.2.0.3 on (recommended)
Please note the advice about unicode support in section [Installation Requirements for ActiveEntry Tools on an Administrative Workstation](#).
- Oracle client tools (recommended)
- Initially set up database
The table space is created using the Oracle Enterprise Manager. The login has to take place as user with dba rights (e.g. "system").
- DBScheduler directory for log files
The directory must exist physically and must be known to the database. This can be done with the following statement using the appropriate query tool:

```
Create directory VI_LOG_DIR as '<path input>';
```

3.1.4 Database User under Oracle

A new database user should be set up in order to use the database. The set up can be done with the appropriate query tool and the following statement:

```
create user <username> identified by <password>  
default tablespace <tablespace_name>  
temporary tablespace <temptablespace_name>  
account unlock;
```

For unrestricted use of ActiveEntry, the database users have to fulfill the following minimum requirements:

- Member in role "dba"

Permissions:

- Permission to load Oracle internal views for configuration and for activities on the database ("select any dictionary").
- Permission to execute Oracle packages for writing directly to file ("execute on sys.utl_file")
- Permission to use unlimited table space ("unlimited tablespace")
- Permission to add and change triggers ("create any trigger", "alter any trigger")

- Permission to add and change procedures ("create any procedure", "alter any procedure")
- Permission to add views ("create any view")
- Database user permissions to write to this directory ("write on directory VI_LOG_DIR")
The local path must be entered as a directory first, assuming the entry for the instance does not yet exist.

3.2 Installation Requirements for ActiveEntry Tools on an Administrative Workstation

ActiveEntry Administration and configuration tools are installed on an administrative workstation for displaying and processing data. The following system prerequisites have to be guaranteed for installing tools on an administrative workstation.

Microsoft Windows Operating System

- Windows XP
- Microsoft .NET Framework with at least Version 2.0
- Microsoft Software Installation (MSI) service
- If an Oracle database system is used, Oracle Client Tools from version 10.2.0.3 onwards also have to be installed.

Advice for Unicode Support when using an Oracle Database System

There are other adjustments that need to be made for unicode support:

- The following environment variable has to be set for applications that use Oracle Call Interface (OCI) for connecting to the database (i.e. SQL*Plus).

```
ORA_NCHAR_LITERAL_REPLACE=TRUE
```

- The following options has to be added to the configuration file for Java application (i.e. SQL developer):

```
AddVMOption -Doracle.jdbc.convertNcharLiterals=true
```

User account

The user who should work with the ActiveEntry administration and configuration tools has to be a local administrator on the workstation.

3.3 Installation Requirements for ActiveEntry Service on a Server

The ActiveEntry program "ActiveEntry Service" ensures that data managed by ActiveEntry is distributed within the network. ActiveEntry Service performs data synchronization between the database and any connected target systems and executes actions at database and file level. The following system prerequisites have to be guaranteed for installing the service on the servers.

Microsoft Windows Operating System

- Windows 2000 Server or Advanced Server with at least Service Pack 2 for Windows 2000 or Windows 2003 Server or Windows 2008 Server
- Microsoft .NET Framework with at least Version 2.0
- Microsoft Software Installation (MSI) service
- If an Oracle database system is used, Oracle client tools from version 10.2.0.3 onwards also have to be installed for database access. Please note the advice about unicode support in section [Installation Requirements for ActiveEntry Tools on an Administrative Workstation](#)

Linux Operating System

- SuSE Enterprise Server 10
- Mono 1.1.13.2
- If an Oracle database system is used, Oracle client tools also have to be installed for database access.

User Account for ActiveEntry Service

ActiveEntry Service configures users, groups and workstations as well as creating directories such as the home and profile directories and creating files. ActiveEntry Service is responsible for the distribution of profile files, machine definitions and logon scripts on the appropriate login server in an ActiveEntry network. The user account for ActiveEntry Service requires the necessary permissions to carry out operations at file level (issuing permissions, adding directories and file to be edited). The ActiveEntry Service user account must be a local administrator on the server. In addition, the user account must belong to the groups "Domain Users", "Domain Admins" and "Account Operators" and be configured with the extended user permissions "Log on as a service" and "Add workstation to domain". Other target system specific permissions may be required for synchronizing ActiveEntry with each target system. These are explained in the corresponding chapters in this Manual.

Chapter 4 Installing ActiveEntry Tools

Before ActiveEntry can be put into operation, the network needs to be setup. The number of servers, their configuration and their functions, all this needs to be determined on an individual basis. For example, the location structure, the desired capacity, the number of users, are all items that need taking into account. The following sections explain the procedure for manually setting up an ActiveEntry network. You can find system requirements in the section [Installation Requirements](#)

Set up the initial database **before** you install ActiveEntry for the first time. You can find the prerequisites for it in the section [Installation Requirements for the ActiveEntry Database](#).

NOTE!

Before you start to install ActiveEntry, close all programs and service components otherwise the setup cannot start.

4.1 Updating ActiveEntry

NOTE!

Older versions of ActiveEntry can only be upgraded to the newest version 4 by uninstalling the old version.

Please take into account in the following installation instructions that updating ActiveEntry from a version older than version 4 to the current version neither be carried out with a manual update nor with help from auto-update. This is a result of the new improvements in ActiveEntry. Certain changes have been made which include the file format, the user interface and the registry structure. The amount of files that ActiveEntry uses, however, does not correspond to the file volume of older versions. Due to this, you need to uninstall all programs and service components of the old ActiveEntry version on this workstation or server before you can start with the installation.

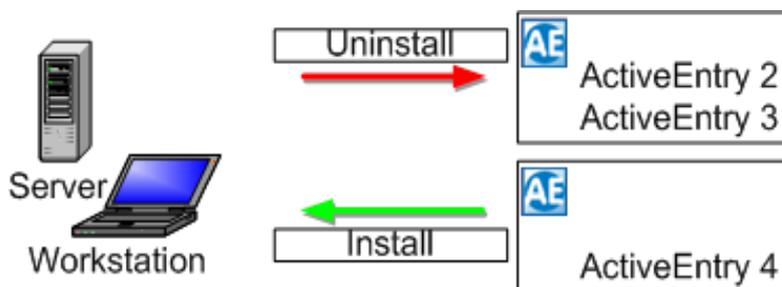


Figure 1: Upgrading ActiveEntry if Older Versions Exist.

4.2 ActiveEntry.Net Setup Wizard

Use the ActiveEntry.Net Setup Wizard for the initial installation of ActiveEntry tools on workstations and servers in the ActiveEntry network. The automatic software update is used to bring an existing installation up-to-date. See section [Updating ActiveEntry Tools](#).

You can install, update and remove all the setup tools provided with the ActiveEntry.Net Setup Wizard. Start the ActiveEntry.Net Setup Wizard from the setup CD by running "setup.exe"

On the ActiveEntry.Net Setup Wizard startup screen, you choose the role for the ActiveEntry tools that need to be installed. The content of the installation depends on which role is selected. The following roles are available:

- **Server**
The installation package ensures the functionality of ActiveEntry Service. It contains the Job provider function as well as the process components.
- **System Configurator**
This installation package contains all the tools for the standard user plus additional programs that are required for system configuration, such as ActiveEntry Transporter, DB-Compiler, Crypto Configuration, ActiveEntry Designer and configuration programs for ActiveEntry Service.
- **Standard User**
This installs all the tools that are necessary for an ActiveEntry user to be able to complete his tasks. Along with the tools that ensure basic functionality for working with ActiveEntry, there is also the main administration tool, the Manager, as well as monitoring programs such as Job Queue Info and Replication Info.
- **All Components**
All ActiveEntry administration and configuration tools are included in this installation package.
- **Profile Editor**
The components in this installation package are the Profile Editor, the Profile Scanner and the MSI Setup Editor.

4.2.1 Microsoft Windows 2000/2003 Terminal Server Installation

To install ActiveEntry tools on a Microsoft Windows 2000/2003 terminal server you need to ensure that the terminal server has been fully installed and configured. This includes profile handling in particular as well as permissions for terminal server use. Take extra care, in an Active Directory Domain, to ensure that the user also has relevant permissions to use the terminal server himself.

In order to install the tools on the terminal server, log in with a user account which has administrative access rights for the terminal server. We recommend that you log in over a console window. Start the console with:

```
Start/run: mstsc /console /v:<servername>
```

where <servername> has to be replaced by the server name of the Terminal Server (without “\” in front).

After that, open the command line console (CMD.exe) and switch the terminal server into software installation mode with help of the command “CHANGE USER /INSTALL”. Start the ActiveEntry.Net Setup Wizard and install the tools as described. Using the command “CHANGE USER /EXECUTE” in the command console, you can end the software installation on the terminal server. After the installation is complete, anyone who is an authorized terminal server user can start the ActiveEntry tools and use them.

For further information about the Microsoft Windows 2000/2003 Terminal Server software installation please refer to the operating system documentation.

As prerequisite for the installation of ActiveEntry tools on a Microsoft Windows 2000/2003 Terminal Server is that the system is completely installed and configured. This includes, in particular, the treatment of profiles and authorization to use the terminal server. Please note that in an Active Directory domain, the user also needs to be authorized to use the terminal server.

4.3 Setting Up an Administration Workstation

System prerequisites for the installation of ActiveEntry administration tools on an administration workstation and the necessary authorization are listed in [Installation Requirements for ActiveEntry Tools on an Administrative Workstation](#).

The initial installation of ActiveEntry tools is done with the ActiveEntry.Net Setup Wizard. The installation is described in section [ActiveEntry.Net Setup Wizard](#). To update the existing installation, use the automatic software update. See section [Updating ActiveEntry Tools](#).

A minimum of the following ActiveEntry tools should be installed on a administration workstation:

- ActiveEntry Manager
- ActiveEntry Identity Manager
- Job Queue Info

Depending on which roles are selected, you can also install:

- DBMigrator
- ActiveEntry Transporter
- DBCompiler
- Crypto Configuration
- Profile Scanner, Profile Editor, Replication Info
- Report Editor
- ActiveEntry Designer
- Object Browser, Script Debugger, Schema Extension, Namespace Mapping Editor
- Software Loader
- License Wizard
- ActiveEntry Service Configurator
- ActiveEntry Service Updater
- ActiveEntry Data Import
- ActiveEntry Analyzer

A general description of the functionality of the individual tools as well as how to start them from the start menu can be found in Chapter [ActiveEntry Tools](#).

4.4 Migrating an ActiveEntry Database

In order to set up a database on a database server for use in the ActiveEntry environment a migration has to be run. The necessary system prerequisites are listed in section [Installation Requirements for the ActiveEntry Database](#).

A migration causes all the necessary tables, data types, database procedures to be copied to the database. The database function "basegroup" is set up and is given full access rights to the database objects. In ActiveEntry, automatic version control is integrated that maintains consistency of ActiveEntry objects in relation to each other and in relation to the database. A database migration is necessary if program changes are made that alter structures, for example, if tables are extended. Depending on the current state of the database, the program "DBMigrator" runs the migration and enters the new state into the version control.

4.4.1 Database Migration under Microsoft SQL Server

A database migration is basically only possible in single user mode. For this reason, all existing database connections need to be closed before the migration starts. Verify in the SQL Server Management Studio whether a process is accessing the migration database. Check if the SQL Server Agent is already running and start it if necessary. You will find the start options for the SQL Server Agent in the Services control panel on the SQL Server.

When an already existing ActiveEntry database is migrated, the database schedules should be disabled to be on the safe side. The database scheduled tasks need to be checked after the migration with help of the SQL Server Management Studios under <SQL Server Agent>/<Tasks> and enabled if necessary before the they start to run again automatically at the preset intervals.

4.4.2 Preparing a Workstation for Migration

The system prerequisites for the installation of ActiveEntry tools onto an administration Workstation are listed in section [Installation Requirements for ActiveEntry Tools on an Administrative Workstation](#).

On the workstation, from which the migration should be started, the following prerequisites have to be implemented:

- Installing "DBMigrator"
Use ActiveEntry.Net Setup Wizard to install the program. We recommend using the installation package for the role "system configurator". Updating the ActiveEntry tools should be done with the ActiveEntry.Net Setup Wizard and not by the automatic software update feature. Read the section [ActiveEntry.Net Setup Wizard](#) for details of installing with ActiveEntry.Net Setup Wizard.

- Access to the directories "Migration" and "Binaries" on the ActiveEntry Setup CD. The "Migration" directory contains the current migration package. The files that are uploaded and distributed by the automatic software update feature are found in the directory "Binaries". Should you back up this directory, make sure that the directory structure remains intact.

4.4.3 Running a Migration with the Program DBMigrator

NOTE!

When you start the program "DBMigrator" from a server, the migration may abort with errors. Always start the program "DBMigrator" on one of the administration workstations!

The process of migration is similar under Microsoft SQL Server and Oracle. In the following section, a migration under Microsoft SQL Server is described. Any differences to the Oracle migration are noted.

The migration is carried out by the programm "DBMigrator". The program takes you through step-by-step. Use the button <Next> to proceed to the next step. The button <Back> takes you to the previous step. You can stop the program with <Cancel>. In this case, all the changes are discarded.

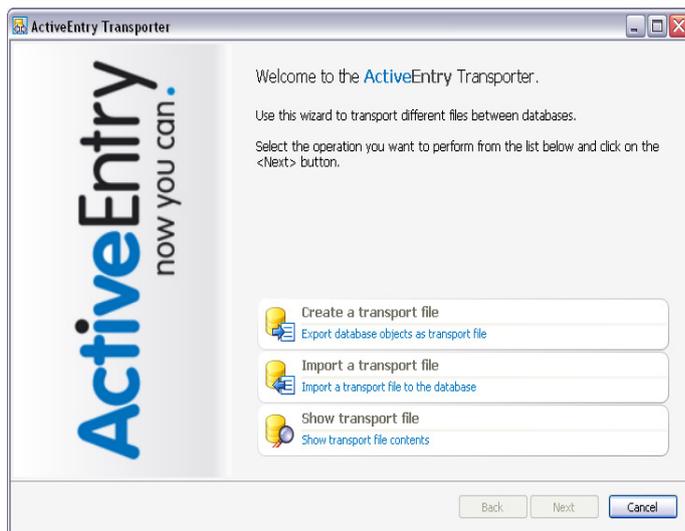


Figure 2: Startup Screen for "DBMigrator"

The main stages of the program are:

- Database login
- Database migration
- Database compilation
- Update database files

Database Login

After the program has started, enter valid connect data for the database. Choose <Select> to open the connection dialog.

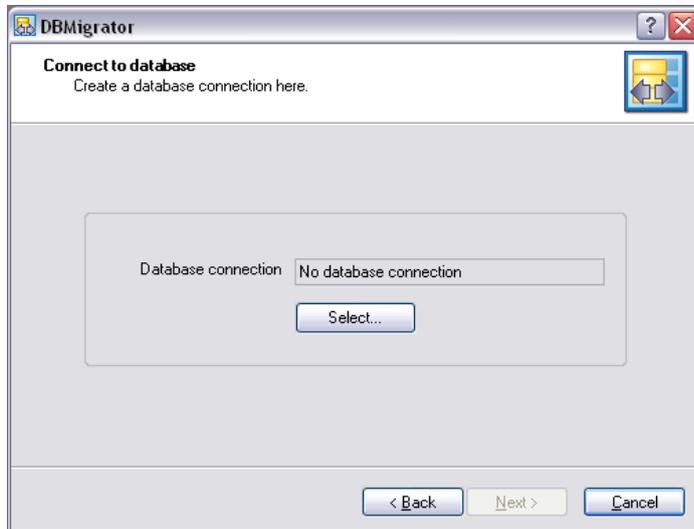


Figure 3: Creating the Database Connection

Log in as described in the section [Logging into the Database with a Database User](#). Enter the database user (usually "sa") and the password of the database user for the database login.

The connection data is passed to the respective database entry during migration. This information is accessed by ActiveEntry Service when tasks are generated. Confirm with <OK> and continue with <Next>.



Figure 4: Connection Data for the Database Login.

Database Migration

After successfully logging onto the database, select the newest migration file (AEDatabaseMSSQL2K.zip). If the migration is under Oracle, select the file AEDatabase-ORACLE9I.zip. Confirm the selection of the migration package with <Next>.

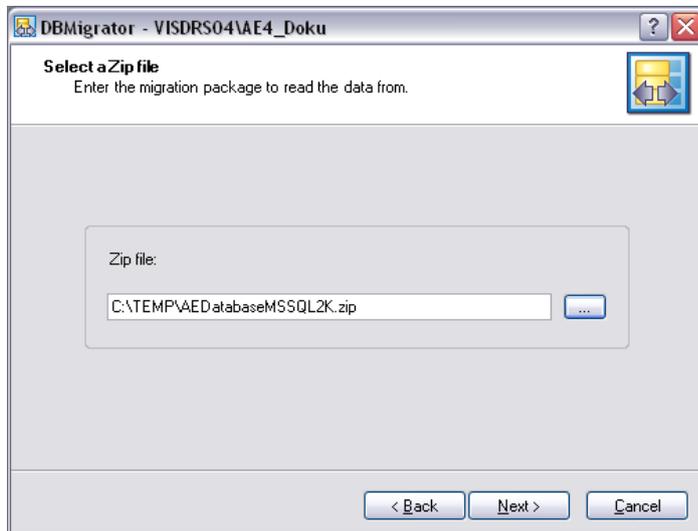


Figure 5: Selecting the Migration Files

The program determines what the current version of the database is and displays the migration steps. If an service pack or an update needs to be installed, you can select a step-by-step procedure with the option <Only perform first step>.

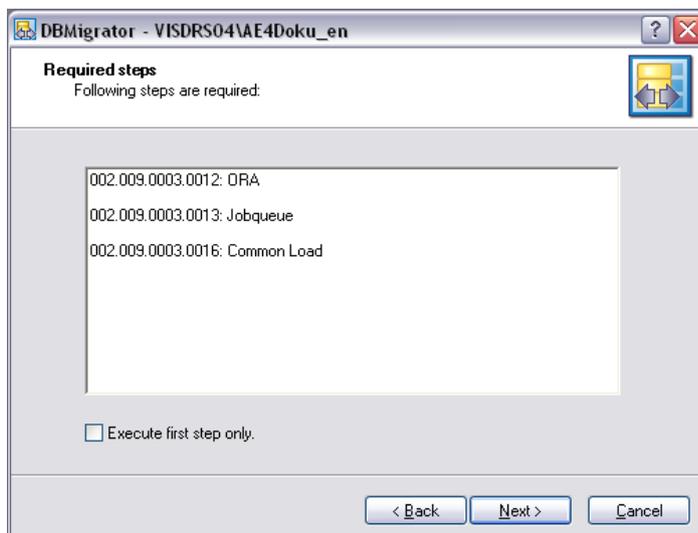


Figure 6: Displaying the Migration Steps

Start the migration by selecting <Next>. A security alert is opened.

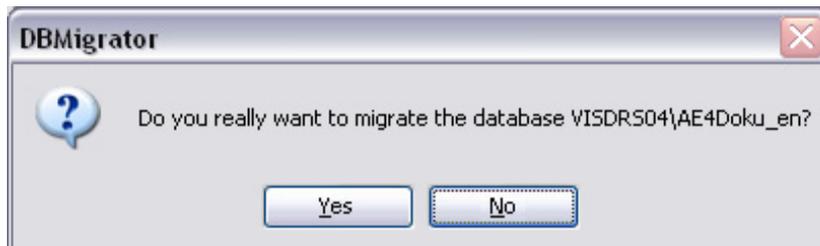


Figure 7: Alert Box

Next, you are asked if a database backup should be created. If you confirm with <Yes>, the migration is continued, if you select <No> it is cancelled. This message is not shown on initial migration. For subsequent work, you should always create a backup before starting the migration in order to reinstate to an original state of the database if necessary.

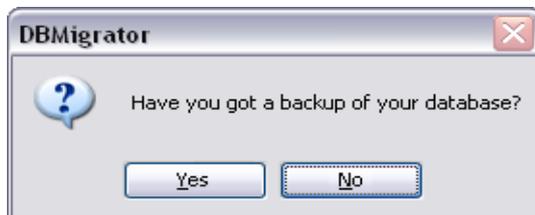


Figure 8: Database Backup Alert Box

During the migration the steps that are being carried out are displayed in a window. The user cannot interrupt the process which can take a number of minutes.

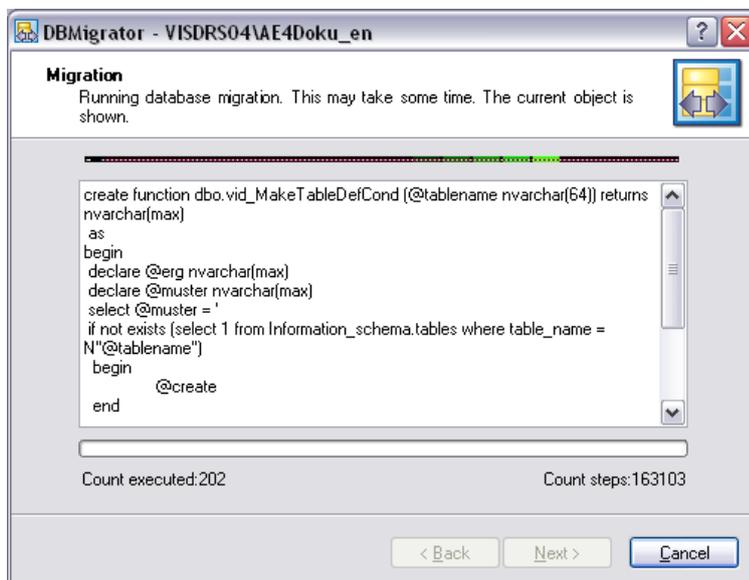


Figure 9: Migration

If the database is not in single user mode, you receive an overview of any active users and processes after starting the migration tool. Select the connections and send users a message using the button <Notify>. Likewise, you can end the selected connections using the button <Kill session>.

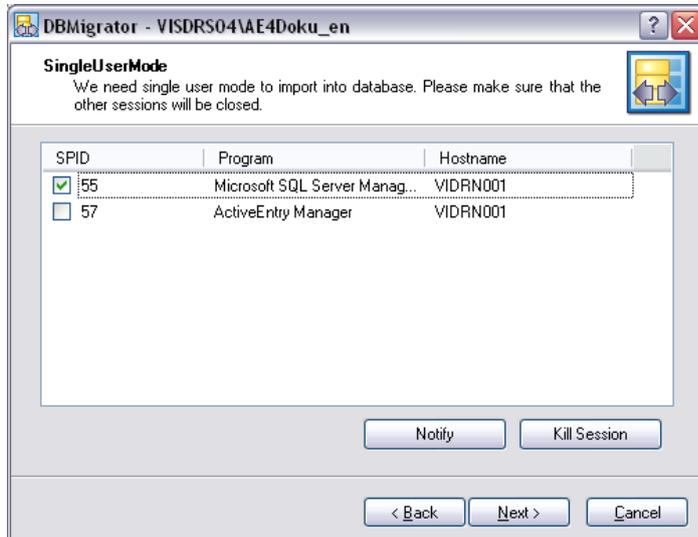


Figure 10: Display Current User

After the migration has been completed without errors, a message appears which you confirm with <Next>. If an error occurs during migration, the migration step in which the error occurred is rolled back. The database is returned to its previous state.

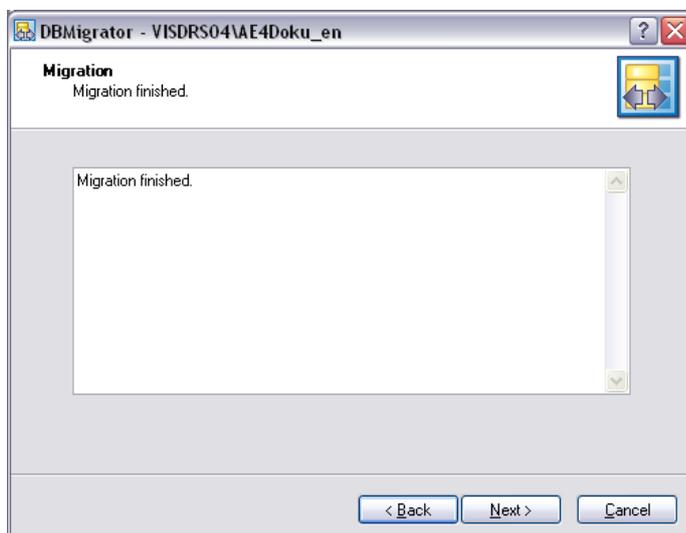


Figure 11: Migration Completed

The program logs the processes and configuration parameters that have been added or deleted by the migration.

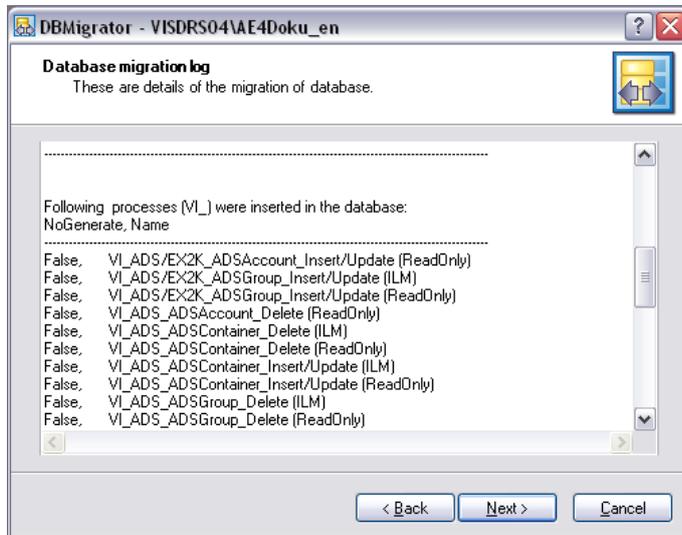


Figure 12: Logging Processes Changes

During migration, computational tasks are set up for the DBScheduler. These are displayed in the next step. The others steps in the program are blocked up to DBScheduler task processing.

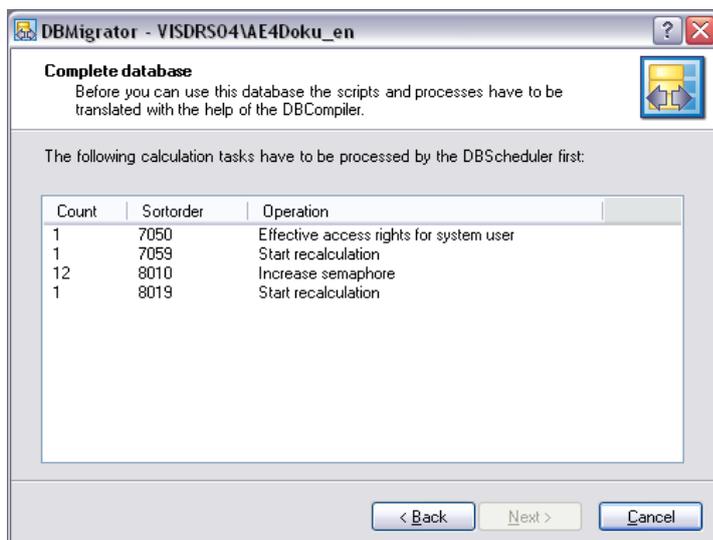


Figure 13: Task Set Up for the DBScheduler

Database Compilation

In order to post processes and script changes to the system, the database needs to be recompiled after the tasks have been completed.

In order to compile the database, log into the program as system user. Select the button <Login as...> and the login window opens.



Figure 14: Creating a Dialog Login

After an initial migration you automatically use the authentication module "System user". By default, the system user "viadmin" (without password) is available after the initial migration. You can use this system user to compile the initial ActiveEntry database and for the first administration tool login. The user interface and permissions structure for "viadmin" should not be used in a productive environment or changed, as it is a template system user from Völcker Informatik and is overwritten by each migration. For further productive use of ActiveEntry, a password should be given to the system user.

The login is described in the section [Logging into ActiveEntry Administration Tool as System User](#). Further authentication modules can be made available via the program "ActiveEntry Designer", after the initial migration.

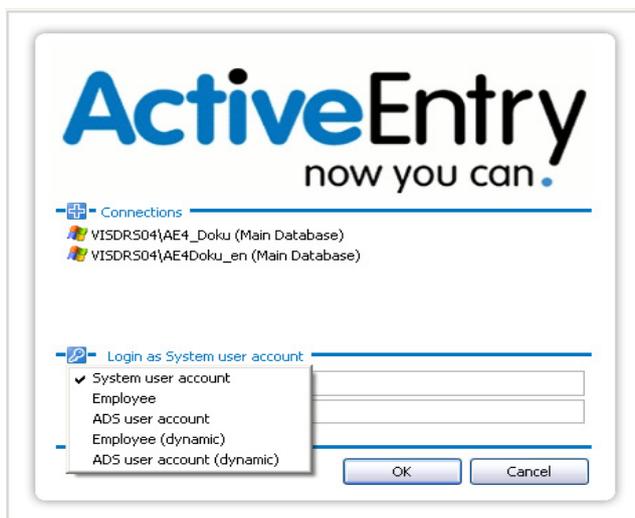


Figure 15: Connection data for the Login

The license is checked during login. If ActiveEntry could not find a valid license, an alert message is displayed accordingly. You can request licenses as soon as the database migration with the DBMigrator has been completed. Use the License Wizard to make a license request. Read section [ActiveEntry Licensing](#) for more information.



Figure 16: License Status Advice

You need to fill out the database connection data for the first compilation.

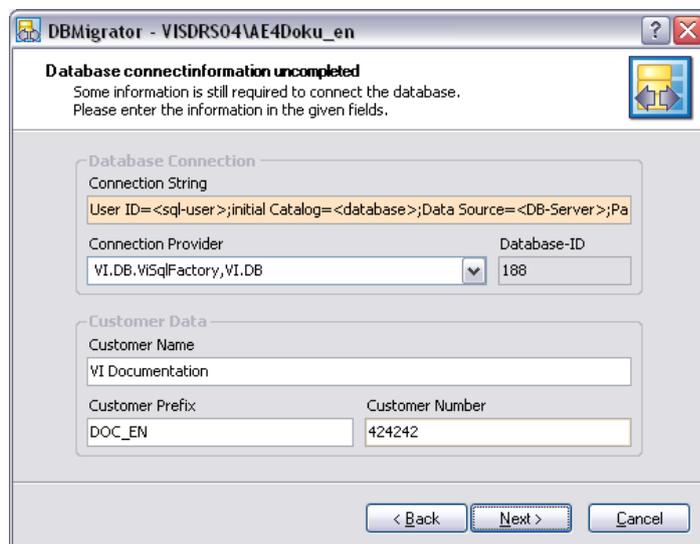


Figure 17: Completing the Database Connection Data

The following data needs to be checked for a ActiveEntry database under Microsoft SQL and corrected if necessary:

- **ConnectionString**
`User ID=<satabase user>;initial Catalog=<database>;
 Data Source=<Server>;Password=<database user password>;
 pooling='false'`

By using a known instance of the database server as data origin, the notation to input the ConnectString is:

```
User ID=<database user>;initial Catalog=<database>;
Data Source=<Server>\<Instance>;Password=<database user password>;
pooling='false'
```

- **ConnectionProvider**
`VI.DB.ViSqlFactory,VI.DB`

For an ActiveEntry schema under Oracle the following data needs to be checked and corrected if necessary:

- **ConnectionString**
Data Source=<data source from TNSNames.ora>;User ID=<database user>;
Password=<database user password>
- **ConnectionProvider**
VI.DB.Oracle.ViOracleFactory, VI.DB.Oracle

Change the connection string using the button next to the input field. Select your database connection data at this point. See section [Logging into the Database with a Database User](#) for more detailed information about connection data. Furthermore, you need to enter the full customer name, prefix and the number.

When you have entered the data, the database connection is tested. After that, compiling the database can begin. Compiling can take some time. The objects being processed are displayed in the dialog window.

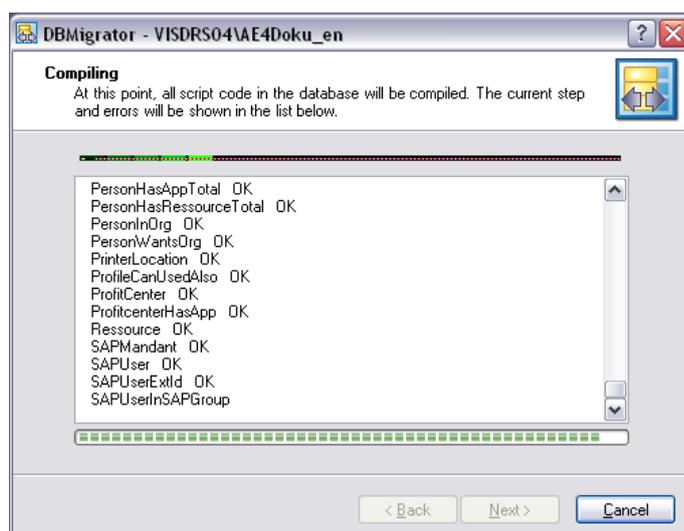


Figure 18: Compiling the Database

Updating the Files in the Database

In order to distribute ActiveEntry tools files that have been added or updated using the automatic software update, they are loaded into the ActiveEntry database in the following sequence.

Next, an alert is displayed asking whether the files need to be updated or not. Confirm this request with <Yes>.

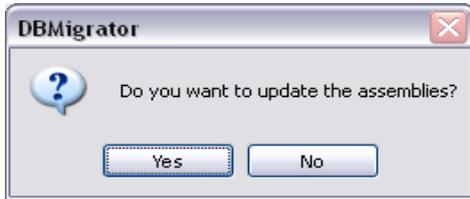


Figure 19: Software Update Alert

In the next step, the database files are loaded. All the files that are in the directory "Binaries" on the setup CD are imported.

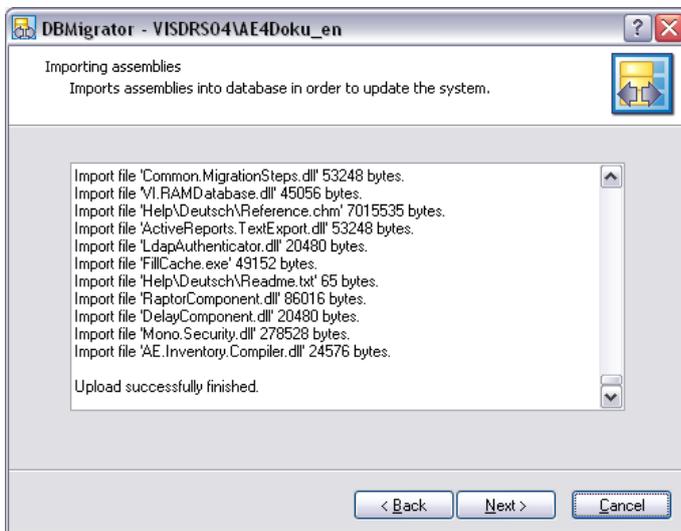


Figure 20: Importing Files to the Database

If the configuration parameter for automatic software update support is not enabled yet, you can do this in the next step.

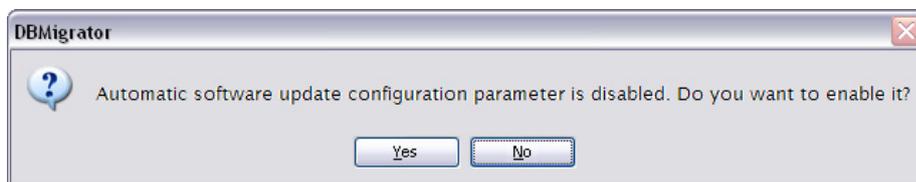


Figure 21: Enable Automatic Software Update

After the files have been successfully loaded into the database, the semaphore value "SoftwareRevision" is update by DBScheduler in the database. When the next semaphore test takes place, the files are added to the list of updated files and therefore distributed to the workstations and Job servers by automatic software update. Read section [Updating ActiveEntry Tools](#) for more information.

By selecting the <Next> button you reach the next dialog window where you can end the program using the button <Finish>. The migration date, the last migration step and the database version are recorded in the database history.

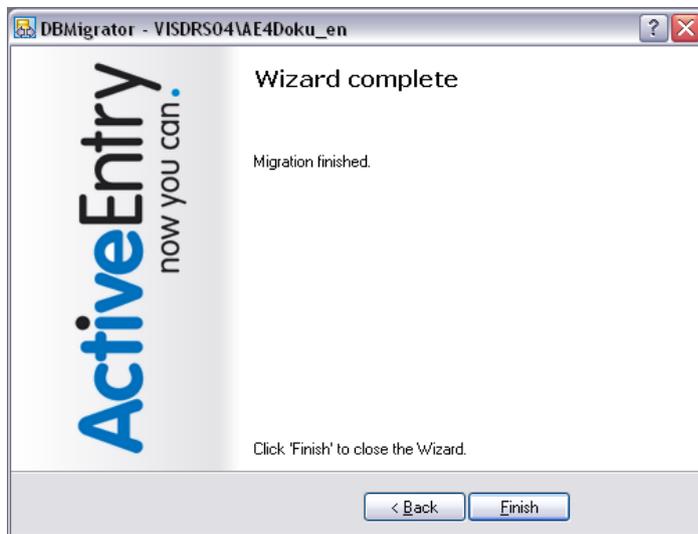


Figure 22: Ending the Program

Before the program finally quits, the following message is displayed.

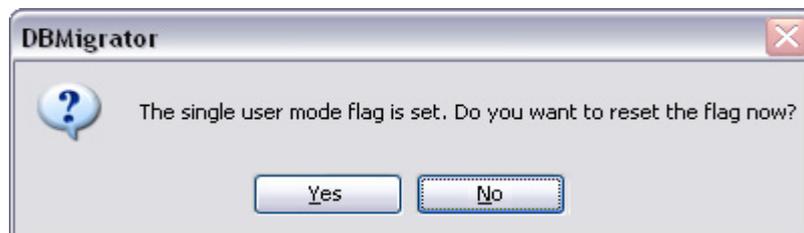


Figure 23: Reset System Update Option

During the migration the database is in single-user mode. The system update option prevents other users connecting to the database. If the migration has completed successfully, you can close the message with the "Yes" button. Other users then have unrestricted access to the database again.

Close the message with "No" if you want to continue using the database in single user mode. To set the database to multi-user at a later point in time, you have to manually reset the option. See section [Database Connection Data](#) in the Configuration Manual.

4.4.4 Setting Up Database Schedules

In order for ActiveEntry to work with the correct functionality and high-performance it is necessary to process several database procedures held by the ActiveEntry database on a cyclical basis. The affected procedures are delivered with the migration and set up as database schedules. The following database schedules are dealt with:

- **vid_DBScheduler**
The DBScheduler assumes the task of calculating processing tasks from the DBQueue. The DBScheduler is called from the database ask "vid_DBScheduler".
- **vid_DialogSchedule**
The database schedule "vid_DialogSchedule" checks the tasks in the system part of the ActiveEntry database and runs the tasks at regular intervals.
- **vi_PayLoadSchedule**
The database schedule "vi_PayLoadSchedule" checks the tasks in the user part of the ActiveEntry database and runs the tasks at regular intervals.
- **vid_CompressJobQueueStats**
When the configuration parameter "Common\JobQueueStats" is set, for any action that alters the Job queue (such as changeing or deleting a process), a new entry is created in the table "Jobqueuestats". The procedure vid_CompressJobQueueStats compresses the entries on an hourly basis and re-enters them with a new UID. The compression takes place for every hour not including the current one.
- **vid_ClearConnects**
The database schedule "vid_ClearConnects" tests database connections for activity and deletes those that have bben inactive for more than 4 hours..

If necessary you can change the scheduled tasks in the database schedules in SQL Server Management Studio under <SQL Server Agent>\<Jobs>.

The time zone on the database server dictates the timing of the database tasks. This avoids unnecessary shifts due to winter and summer time. Daily maintenance jobs that are queued by the DBScheduler, for example, calculating statistics or indexing the database, are run at midnight server time. In case this is during main working hours, you can specify another time using the parameter "Common\DBScheduler\MaintHourLocal".

4.4.5 Posting and Sharing ActiveEntry IT Shop

If you use ActiveEntry IT Shop, you need to post the web application after you have migrated your database for the first time. Read the section "Posting ActiveEntry Web Applications" in teh Web Designer Reference for more details.

If you update your database with a migration package, you need to reshare the project that your web application is based on. Read the section "Dialog 'Project Sharing'" in the Web Designer Reference.

4.5 ActiveEntry Licensing

ActiveEntry has to be licensed in order to work with it without restrictions. There is no write access to the ActiveEntry database without a valid license. Read access is however available without a license.

A new license is required after initial ActiveEntry database migration, when a version is updated or when the license runs out. Licenses are valid for a specific period of time and for a specific number of units.

Use the License Wizard to create a license request and send it to the address "licenceorder@voelcker.com". To order a license you need your customer details such as the ActiveEntry database ID, the database server name, customer name, number and prefix. Use the License Wizard to install the license one you have received it.

4.5.1 Creating a License Request

Use the License Wizard to create your license request. The wizard guides you through each step. Use the <Next> button to move on to the next step. Select <Back> to return to the previous step. Use the <Cancel> button to discard the changes and exit the wizard.



Figure 24: License Wizard Startup Screen

Licenses are stored in the ActiveEntry database. First, you decide whether to create the license request for an existing ActiveEntry database or for an ActiveEntry database, which still has to be installed.



Figure 25: Preparing a License Request

To request a license for an existing database, enter the ActiveEntry database connection data by creating a connection with the <Select> button in the next step.

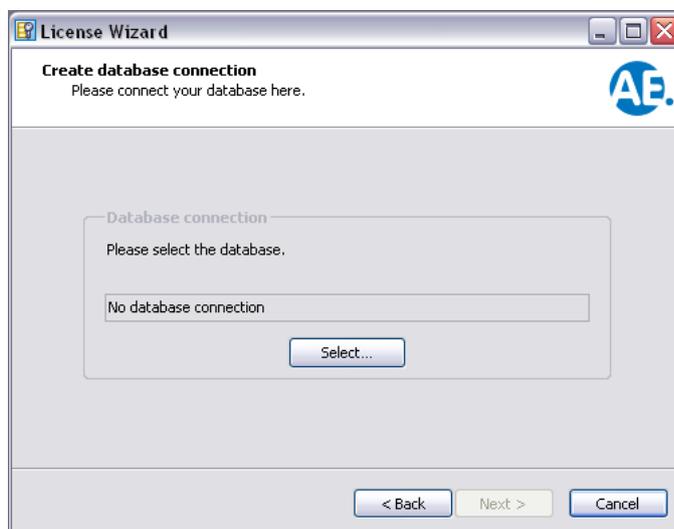


Figure 26: Creating a Database Connection

The login takes place as described in the section [Logging into ActiveEntry Tools](#). After entering the required data, confirm with <Next>.



Figure 27: Login

Select which product to license and the license type in the next step. There are different options available depending on whether you want to create a request for an existing or for a new ActiveEntry database.

If you create a license request for an existing ActiveEntry database, the product is already selected. You can choose between a request for a new license or for an extension to an existing license as license type. You need a new license type for an initial ActiveEntry installation or when changing to a new ActiveEntry version. When the license runs out or the number of license units is exceeded you can extend the existing license type. To do this, select the existing license type in the pop-up menu. You create a license request for a new ActiveEntry database by selecting the product "ActiveEntry" and requesting a new license type.

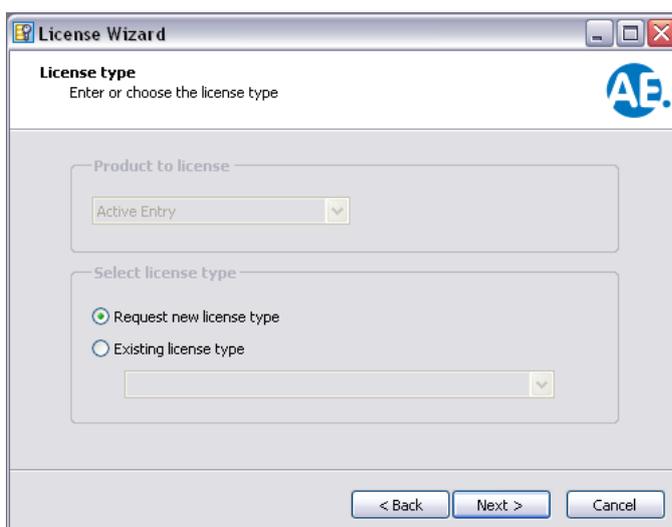


Figure 28: Selecting Product and License Type

To order a license you need your customer details such as the ActiveEntry database ID, the database server name, customer name, number and prefix. The customer details are taken from the database if it already exists. Enter the customer details for a new database in the next step.

In order for a license to be issued, enter a number of units and validity period to be licensed according to your authorized number of licenses. If the license type only needs to be extended because the units have been exceeded, enter the addition number of units required. The actual number of license units is calculated from the existing license type and the extension.

Figure 29: License Data Input

Initialize the details for sending the license request in the next step. Enter the email address that the license should be delivered to.

Figure 30: Saving and Sending the License Request

If you do not want to send the license request until a later date, you can save it in a license file (*.lic). To do this you enable the option <Save in file> and enter the path and file name. To send the license request immediately, enable the option <Send as mail>. This creates a new email in the default mail program, which contains the destination address, license file and customer details. The email with the license request is sent to the address "licenceorder@voelcker.com".

Then you can then exit the License Wizard.

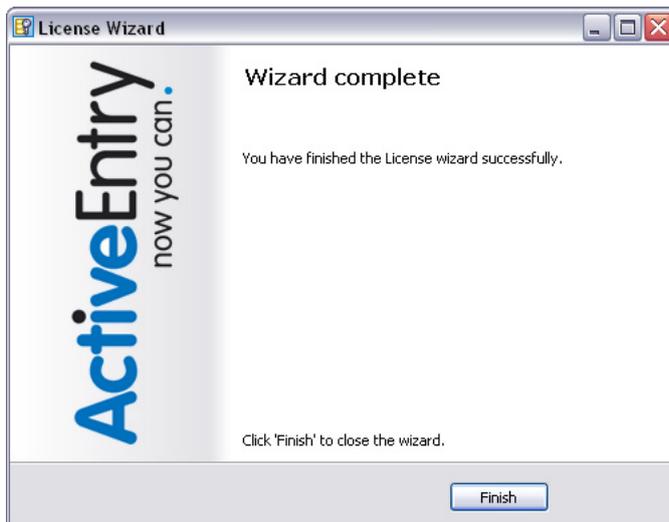


Figure 31: Exiting the License Wizard

4.5.2 Installing a License

After receiving your license you have to install it. Use the License Wizard to do this. The wizard guides you through each step. Use the <Next> button to move on to the next step. Select <Back> to return to the previous step. Use the <Cancel> button to discard the changes and exit the wizard.



Figure 32: License Wizard Startup Screen

Select the option <Install license> in the next step.



Figure 33: Preparing the License Installation

To install a license for an existing database, enter the ActiveEntry database connection data by creating a connection with the <Select> button in the next step.



Figure 34: Connecting to the Database

The login takes place as described in the section [Logging into ActiveEntry Tools](#). After entering the required data, confirm with <Next>.



Figure 35: Login

Enter the license file to be installed. The license is installed after you confirm with the <Next> button.

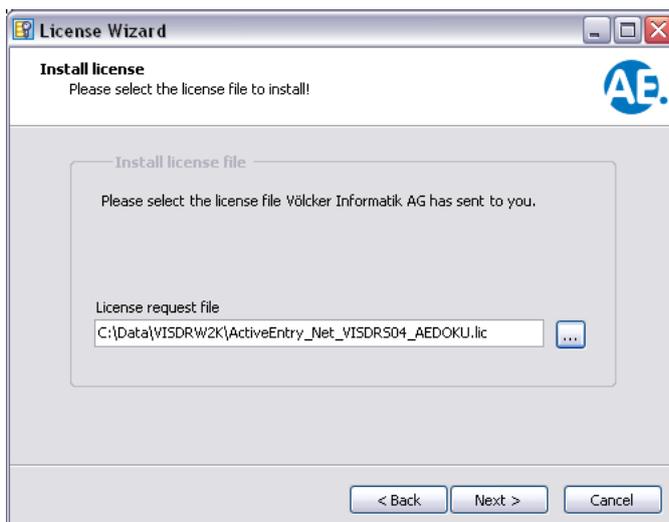


Figure 36: Installing the License

Then you can close the License Wizard.

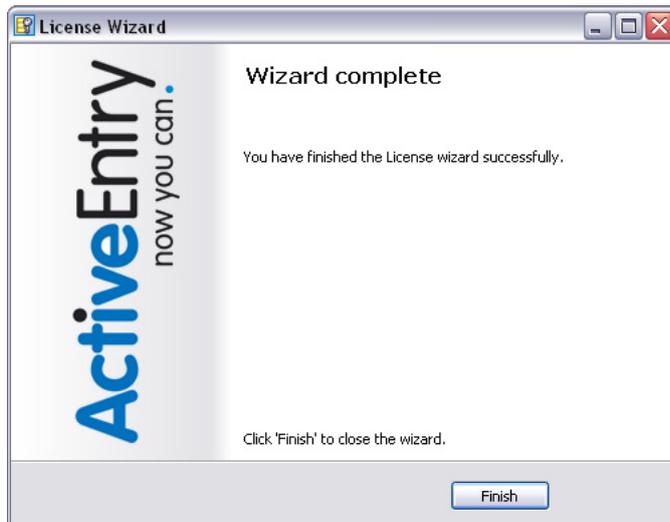


Figure 37: Completing the License Wizard

4.6 Setting Up a Server for Database Access

To install ActiveEntry Service the system prerequisites need to be guaranteed on the servers, as described in section [Installation Requirements for the ActiveEntry Database](#). The first installation of ActiveEntry Service is done with the ActiveEntry.Net Setup Wizard. The installation is described in section [ActiveEntry.Net Setup Wizard](#). Also note the Terminal Server installation advice (see [Microsoft Windows 2000/2003 Terminal Server Installation](#)). In order to update existing installations, the automatic software update is used. See section [Updating ActiveEntry Tools](#).

4.6.1 ActiveEntry Service's Mode of Operation

The server support program "ActiveEntry Service" is responsible for propagating ActiveEntry administration information within the network. ActiveEntry Service includes "process components" that require no further system prerequisites, for example, "process components" to carry out actions at database and file level. Furthermore, there are target system specific process components integrated that synchronize the ActiveEntry database with the individual target system. These require additional system requirements.

A "Job" provider function makes a "Job" destination process step available within ActiveEntry. The Job destination function handles the process steps and returns a result to the Job provider. The Job provider evaluates the result.

The combination of a Job provider on one server and a Job destination on another server is called a "Job gate". The Job provider and Job destination are configured within the Jobgate such that they can communicate with each other.

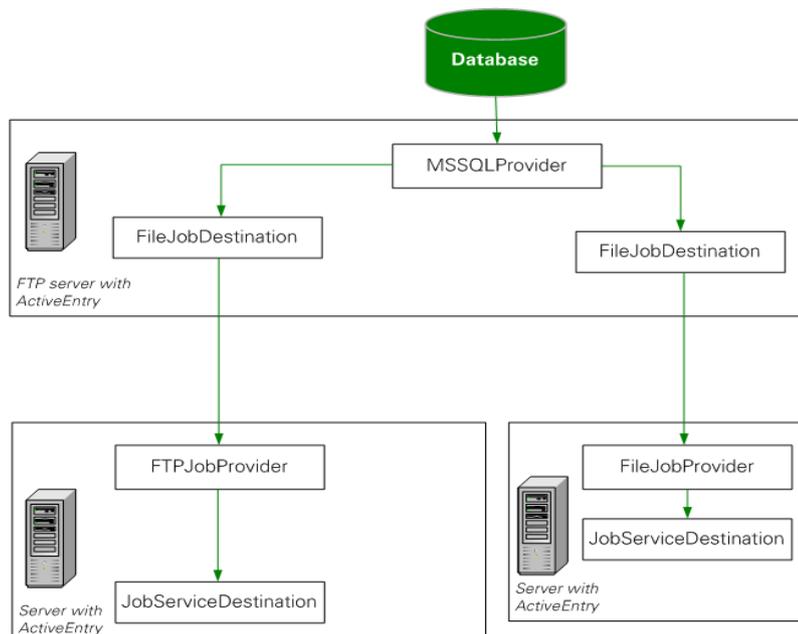


Figure 38: ActiveEntry Service Operating Mode

ActiveEntry Service Job Providers

ActiveEntry Service makes the following Job providers available:

- **MSSQLJobProvider**
The MSSQLJobProvider collects process steps from the database on the Microsoft SQL Server and sends them to a Job destination.
- **OracleJobProvider**
The OracleJobProvider collects process steps from the database on the Oracle Server and sends them to a Job destination.
- **FileJobProvider**
The FileJobProvider reads process requests and results from files and writes them to file. These files can be processed by FileJobGate (File or FTP Job destination). The data is transfer via these files.
- **FTPJobProvider**
The FTPJobProvider is based on the functionality of the FileJobProvider. The FTPJobProvider reads process requests and results from files and writes them to file. After the files have been created in local directories, the FTPJobProviders connects to the FTP Server and transfer the files to the server.
A connection is also made to the FTP Server when it gets a signal and the data is collected.

- **HTTPJobProvider**
The HTTPJobProvider receives process steps from a parent server. The data transfer is done by HyperText Transfer Protocol.
- **WebServiceJobProvider**
There is a web service on the Web Server that accesses the process switching directories for a FileJobDestination and makes an external interface available to post requests and to poll for responses. The WebServiceJobProvider processes the process steps that are set up by the web service.

ActiveEntry Service Job Destinations

The following Job destinations are available:

- **JobServiceDestination**
The JobServiceDestination is the ActiveEntry Service tool that actually deals with processing the process steps. It request the process steps from the Job provider, processes them with the process component and returns the result.
- **FileJobDestination**
The FileJobDestination processes the process steps that are made available by the FileJobGate (FileJobProvider or FTPJobProvider) and returns the results to the Job provider.
- **FTPJobDestination**
The FTPJobDestination processes the process steps that are made available by the FileJobGate (FileJobProvider or FTPJobProvider) and returns the results to the Job provider.
- **HTTPJobDestination**
The HTTPJobDestination sends process steps to a child Job server. The data transfer is in HyperText Transfer Protocol.

ActiveEntry Service Job Gates

The following Job gates are integrated in ActiveEntry Service:

- HTTPJobGate consists of HTTPJobProvider and HTTPJobDestination.
- FileJobGate consists of FileJobProvider, FileJobDestination, FTPJobProvider and FTPJobDestination. FileJobProvider, FileJobDestination, FTPJobProvider, FTPJobDestination can be combined with each other.

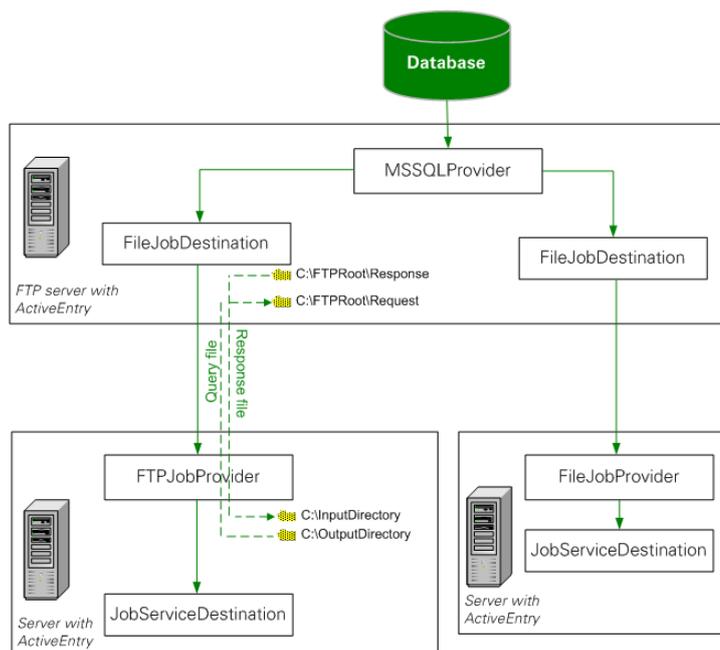


Figure 39: Example Configuration for FileJobGate

4.6.2 Configuring ActiveEntry Service

The configuration of ActiveEntry Service is contained in a configuration file. The file has to reside in the same directory as viNetworkService. The service and its plug-ins are configured using this file. There is one unique section in the file for each of the different modules in ActiveEntry Service. The configuration file is necessary both for ActiveEntry Service on a windows based operating system and for the Linux daemon.

Two configuration files are supported:

- **Jobservice.cfg**
Jobservice.cfg is an XML configurationfile in VI's own format. The advantage of this file is that reloading during working hours is supported.
- **viNetworkService.exe.config**
viNetworkService.exe.config is the standard configuration file for .NET executables and therefore has a pre-defined format.

A value is initially searched for in the configuration file "Jobservice.cfg" in order to determine the setups. If the value is not found, the file "viNetworkService.exe is automatically used. ActiveEntry Service can thereafter, only use work with this file. An example of the configuration files is found in the Configuration Manual in section [ActiveEntry Service Configuration Files](#).

4.6.3 Working with ActiveEntry Service Configurator

You can adapt the configuration files to suit your requirements with the program "ActiveEntry Service Configurator". When the program starts, the configuration file (Jobservice.cfg) is loaded that is found in the application directory. The path of the file that is loaded is displayed in the program's title bar.

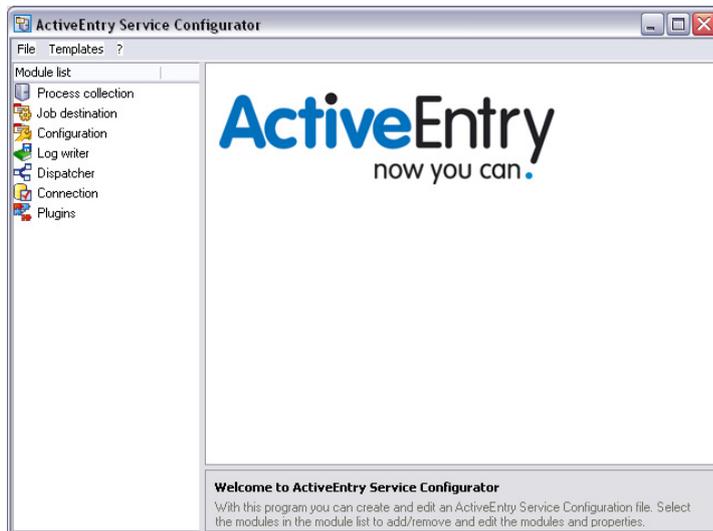


Figure 40: Startup Window for the Program "ActiveEntry Service Configurator"

ActiveEntry Service Configurator functions are also available in the program "ActiveEntry Designer" via the Job Server Editor. Read section [Job Server Declaration](#) in the Configuration Manual for more information.

In the following table, the meaning of the program's menu commands is explained:

Table 1: ActiveEntry Service Configurator - Menu Commands and Key Combinations

Menu	Menu Command	Meaning	Key Combination
File	New	Create a new configuration	Ctl + N
	Open...	Load an existing configuration file	Ctl + O
	Save	Save the configuration file	Ctl + S
	Save as..	Save the configuration file with different file type (*.config, *.cfg or both)	Ctl + Shift + N
	Verify	Test the configuration settings	Ctl + Shift + V
	Quit	Quit the program	Ctl+ Q
Templates	SQL Server direct	Template for a configuration file with a direct connection to an SQL Server	
?	Help	Program help	

ActiveEntry Service Configuration Modules

The separate configuration sections are listed in the module list. The following modules need to be configured:

- **Process collection**
Specify the Job provider in this module. The configuration is described in section [Process Collection Module](#).
- **Jobdestination**
Specify the Job destination is defined in this module. The configuration is described in section [Jobdestination Module](#).
- **Configuration**
The standard configuration settings for ActiveEntry Service are in this module. The configuration is described in section [The Configuration Module](#).
- **Log Writer**
This module writes ActiveEntry Service messages to a log file. The configuration is described in section [The Log Writer Module](#).
- **Dispatcher**
ActiveEntry Service is configured as a dispatcher in this module. The process requests from the child Job server are buffered, processed and forwarded. The configuration is described in section [The Dispatcher Module](#).
- **Connection**
With this module you make special configuration settings for ActiveEntry Service administration. The configuration is described in section [The Connection Module](#).
- **Plug-ins**
In this module you specify, which plug-ins should be installed. The configuration is described in section [The Plug-in Module](#).

Selecting the Module Types

A selection of module types is available for certain modules. You open the selection list (<Select module> or <Insert> button), select a module type and add it to a module list (<OK> button). Use the <Cancel> button to discard changes and close the dialog box

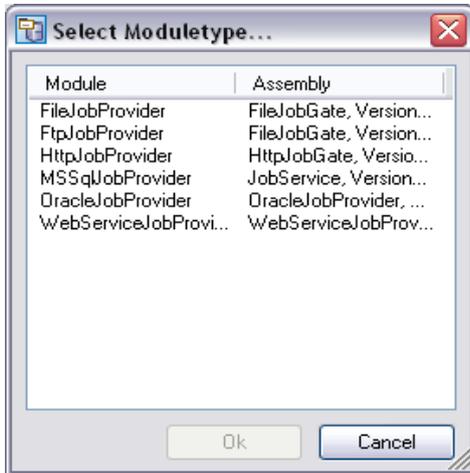


Figure 41: Selecting a Module Type

Depending on the module type, the selection can be renamed (<Rename>) or removed again from the configuration (<Delete>).

When an entry in the module list is selected, the possible properties for the module are displayed. The following icons are used:

Table 2: Meaning of the Icons for the Module Properties

Icon	Meaning
	The property is preset with a standard value. The value is passed as a string.
	Compulsory input. The property must be altered as required. The value is passed as a string.
	The property is preset with a standard numeric value. The value is passed as an integer.
	The property can be activated and deactivated.

Template for the Configuration File

A template for an ActiveEntry Service Configurator configuration is supplied with the ActiveEntry Service Configurator. This template already contains the most important modules with settings for a simple ActiveEntry Service configuration with a direct connection to an SQL Server. You can load the template using the menu item <Templates>\<SQL server direct>. After loading, the configuration the template needs to be modified as required.

Configuration File Verification Test

This program can also carry out a verification of the configuration file. The test ensures that the minimum requirements for a configuration file are met. Errors are output to a message window..

Table 3: Verification Test Error Output

Error	Output
No Job provider found	error
No Logwriter found	error
No input in compulsory field	error
No Job destination found	warning
No Plug-ins found	warning

4.6.4 Process Collection Module

In this module you define the Job providers. Read more in section [ActiveEntry Service Job Providers](#).

The following module types are available:

- MSSQLJobProvider
- OracleJobprovider
- FileJobProvider
- FTPJobProvider
- HTTPJobProvider
- WebserviceJobProvider

MSSQLJobProvider

The MSSQLJobProvider handles process requests made to an ActiveEntry database on a Microsoft SQL Server.

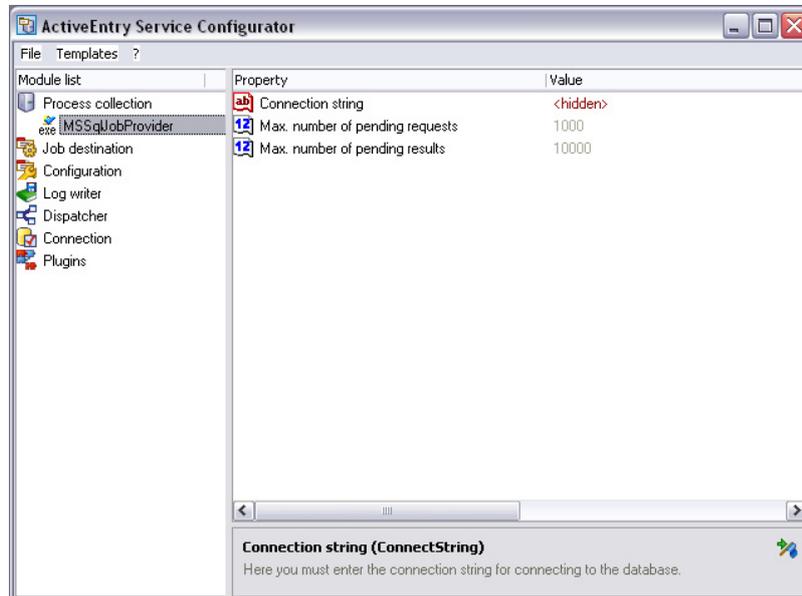


Figure 42: SQLJobProvider Configuration Data

Following parameters are available:

- **Connection Parameter (ConnectionString)**
This parameter supplies the access data for the database server and the database that is to be used. If you double-click on the parameter, the connection dialog is opened and you can enter the database server, database user with password and the database to connect to.
- **Max. number of requests (RequestQueueLimit)**
The MSSQLJobProvider internally caches process requests that are queried by the database. This value defines the maximum number of each entries. The default value is 1000.
- **Max. number of depended results (ResultQueueLimit)**
The MSSQLJobProvider internally caches the process results that are written to the database. This value defines the maximum number if each entries. The default is 10000.

OracleJobProvider

The OracleJobProvider handles ActiveEntry database process requests on an Oracle server.

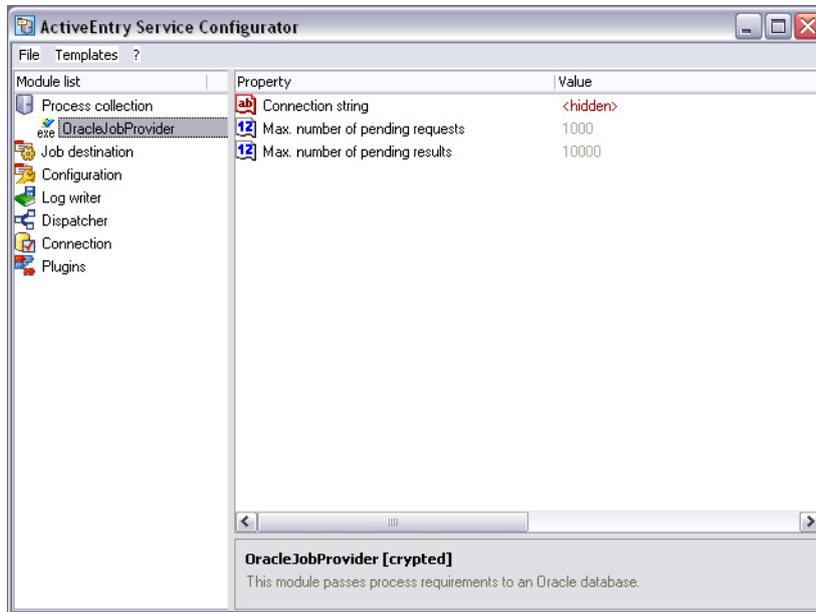


Figure 43: OracleJobProvider Configuration

The following parameters are available:

- ConnectionString**
 The ConnectString defines the access data for the database server and the database that is to be used. By double-clicking on the parameter, the connection dialog is opened and you can enter the data source and database user with password.
- Max. number of requests (RequestQueueLimit)**
 The OracleJobProvider internally caches process requests that are queried by the database. This value defines the maximum number of each entries. The default value is 1000.
- Max. number of depended results (ResultQueueLimit)**
 The OracleJobProvider internally caches the process results that are written to the database. Use this parameter to define the maximum number if each entries. The default is 10000.

FileJobProvider

In the FileJobProvider the process requests and results are written to file or read from and written to file. These files can be processed by FileJobDestination. The data transfer takes place via the files.

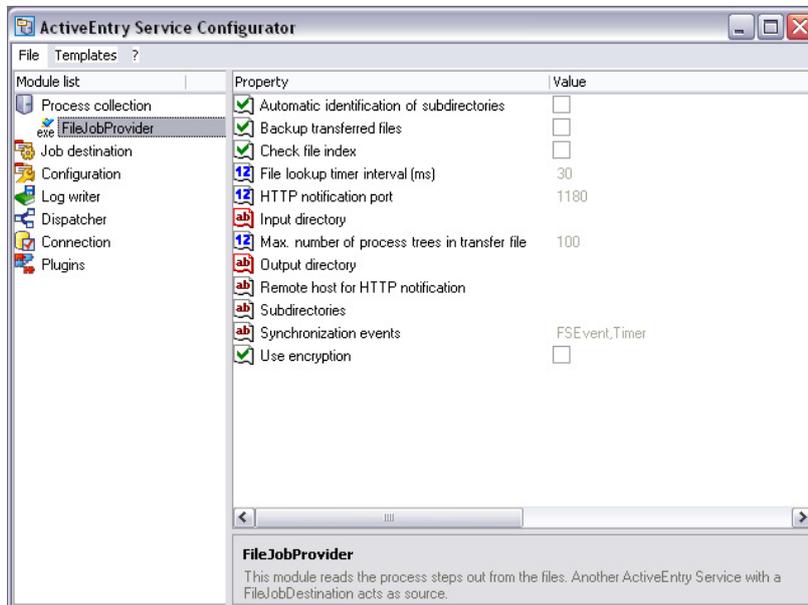


Figure 44: FileJobProviders Configuration Data

The following parameters are available:

- Backup transferred files (BackupFiles)
If this option is set, all the file are moved to a directoy "Backup" irrespective of errors. In the default case (not set) only files with errors are saved.
- Check file index (CheckInputIndex)
If this option is set, the file name index is checked to see if has increased in size. Files with the same or a lower index are not processed. This option is not set by default.
- Max. number of process trees in one transfer file (MaxListCount)
This setting specifies the maximum number of process steps that can be grouped together in one file. This allows limiting of the file size.
- Use encoding (UseEncryption)
The data is encoded when written to file. The setting has to be configured in Job provider and Job destination.
- Notification procedure (EventTypes)
The FileJobProvider supports three different methods of acquiring information about new data. The different results can be combined when separated by commas.

E.g.:

TIMER, FSEVENT

Table 4: Permitted Notification Methods

Result Type	Description
Timer	Request for newly added data takes place at timed intervals
HTTP	The FileJobProvider queries the parent job server via HTTP and processes newly added data after its receiving a reply.
FSEvent	Request for newly added data takes place after an event from the file system

- HTTP notification destination computer (HostName)
Enter the name of the target computer here that will receive the queries if "HTTP" event types are used for the notification method.
- Port for HTTP notification (Port)
Enter the transfer port here if "HTTP" event types are used for the notification method.
- Monitoring interval for input directory (TimerInterval)
Enter the time interval in milliseconds if events of type "timer" are used for the notification method.
- Directory for receiving input (InputDirectory)
The module reads and processes the process files (*.fjg) in this directory. It is necessary to ensure that the Job provider and associated Job destination use the same directory. The input and output directories are correspondingly reversed.
- Destination directory for processed files (OutputDirectory)
The processed files are written to this directory.
- List of subdirectories (SubDirectories)
A list of directory names separated by a pipe character "|" can be entered here. All the directories are then monitored and processed correspondingly. The following directory structure is expected:

```
SubDirectories = "ServerA|ServerB"
...
Request
  ServerA
  ServerB
Response
  ServerA
  ServerB
```

where "Request" and "Response" are directories enter in the parameters "InputDirectory" and "OutputDirectory".

NOTE!

Only events of type "Timer" can be used as notification methods (EventTypes). The event types "HTTP" and "FSEvent" are not possible!

- Automatic identification of subdirectories (AutoSubDirectories)
If this option is enabled, the module automatically processes all the files in the subdirectories. Processing is not recursive.

FTPJobProvider

After the files have been created in the local directory, the FTPJobProvider connects to the FTP server and transfers all the files. After a signal, a connection is set up to the FTP Server and the data is transferred. The directories "Request" and "Response" are expected to be found on the FTP Server. The names of these directories are fixed and cannot be changed. The software components (Job provider/Job destination) deposit or collect the files from here. The FTP user requires the necessary access rights to create, rename and delete files.

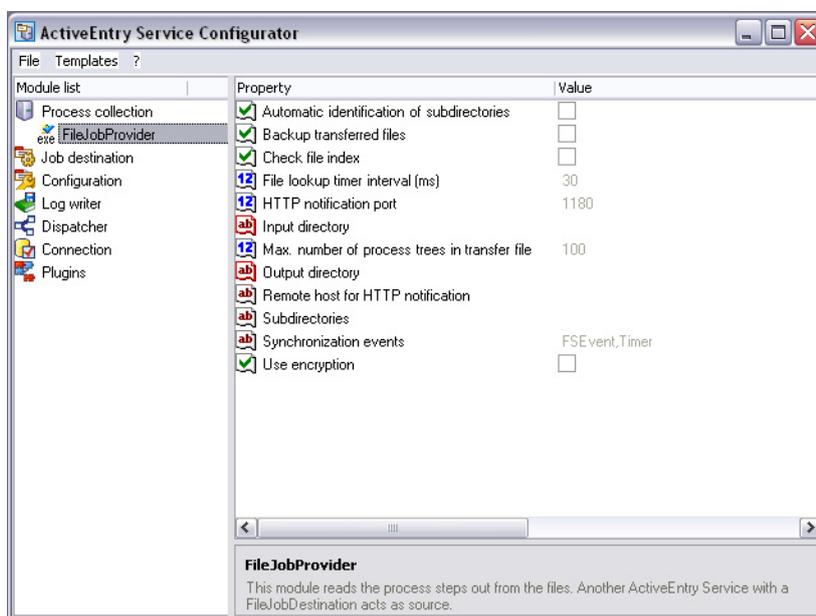


Figure 45: FTPJobProvider Configuration Data

The FTPJobProvider is based on the functionality of the FileJobProvider. Therefore, a large number of the functions are the same (see [FTPJobProvider](#)). Only the extra parameters are described here:

- FTP server (FTPServer)
Enter the name or the IP address of the FTP Server.
- FTP port (FTPPort)
If the FTP Server does not use the default port 21 for FTP transfer, the required port may be entered here.
- FTP user name (FTPUser)
Enter the user name with which the FTPJobProvider logs on onto the FTP Server.
- FTP password (FTPPassword)
Enter the password for the user account for the FTP login here.

HTTPJobProvider

The HTTPJobProvider receives process steps from a parent Job server. The file transfer takes place via HTTP.



Figure 46: HTTPJobProviders Configuration

The following parameters are available:

- Receiver port (ParentPort)
Enter the HTTP port of the parent Job server.
- Receiver server (ParentServer)
Enter the DNS name or the IP adress of the parent Job server.
- Number of retries (Retries)
This value defines how many time the module retries the data transfer if it fails.
- Time interval between retries (RetryDelay)
This time delay defines how long a module waits after a failed process transfer before re-trying.
Time delay format:
`day.hour:minutes:seconds`

WebserviceJobProvider

On the Web Server, in addition to ActiveEntry Service, there is a web service installed that accesses the ActiveEntry Service FileJobDestination transfer directory. It sets up an external interface to post requests and poll responses. The directories "Request" and "Response" are expected to be found on the Web Server. These transfer directories are entered in the file JobTransfer-WebService.cfg. The directories need to be transferred to the FileJobDestination parameters

“Inputdirectory” and “OutputDirectory.” The web service deposits the files in these directories or the posted requests are stored as files in the respective directory. When feedback is requested the files are sent and deleted locally.

In order to access the web service as a process source, a WebServiceJobprovider has to be configured. The WebServiceJobprovider processes the process steps that are made available over the web service. The web server user account requires the necessary access rights to create, rename and delete files.

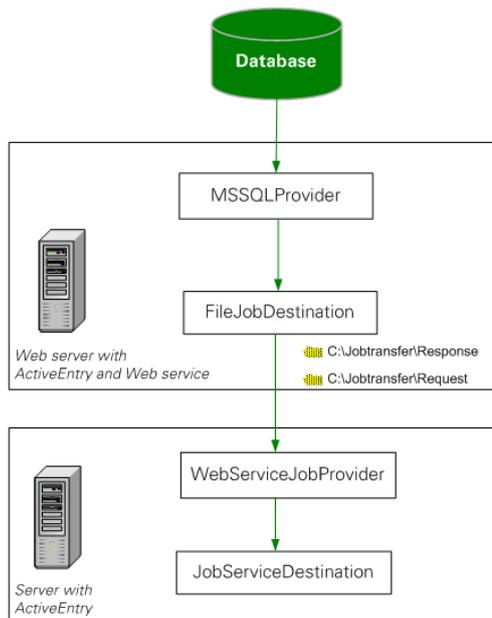


Figure 47: Example Configuration for Web Service Job Processing

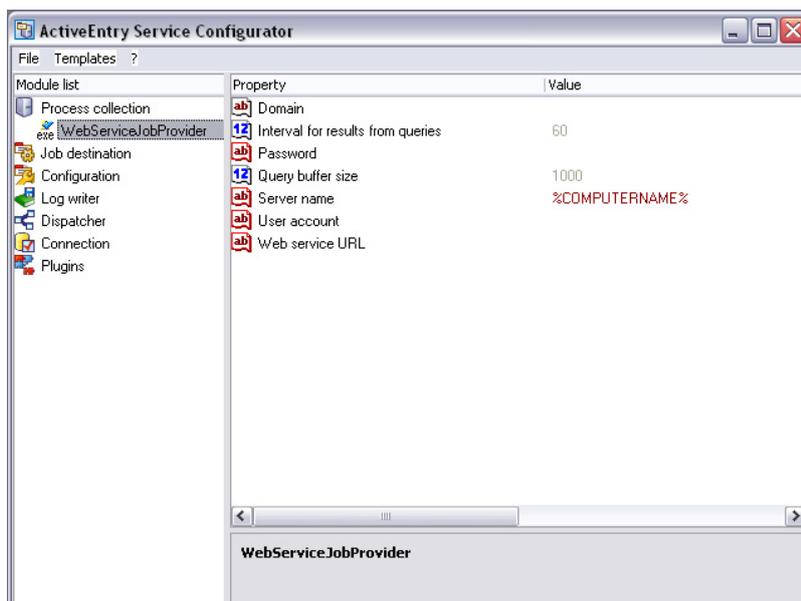


Figure 48: WebServiceJobProvider Configuration Data

The following parameters are available for configuring the WebServiceJobProvider:

- **Servername**
This parameter contains the name of the server that will process the Job provider process steps. Since several Job providers can work in parallel with a web service, each of the web service transfer directories have a subdirectory with the name of the server entered here. The process steps are processed in these directories.
- **Web service URL (URL)**
This parameter contains the web service URL.
- **Domain (Domain)**
Enter the user account domain that the web service runs under.
- **User account (User)**
The user account that the web service runs under.
- **Password (Password)**
Enter the password for the web service user account.
- **Size of request buffer (RequestQueueLimit)**
The WebserviceJobProvider internally caches the process requests that are queried. This value defines the maximum number of each entries. The default value is 1000.
- **Interval for checking responses (CheckResponseInterval)**
This parameter specifies the interval between testing for new tasks. The input is in seconds.

4.6.5 Jobdestination Module

In this module you can define the Job destinations with ActiveEntry Service. The following module types may be selected:

- `JobServiceDestination`
- `FileJobDestination`
- `FTPJobDestination`
- `HTTPJobDestination`

Within a configuration file you can configure as many Job destinations as you wish. The associated configuration sections are determined by name. Therefore the Job destinations that are added can be renamed.

JobServiceDestination

The JobServiceDestination is the ActiveEntry Service module that deals with the actual processing of the process steps. A JobServiceDestination requests the process steps from the Job provider, processes them using process components and returns the results.

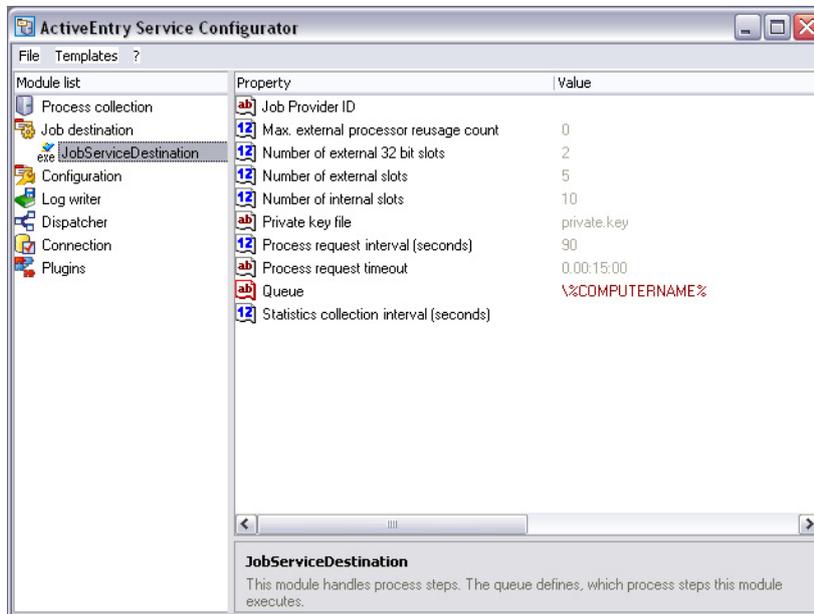


Figure 49: JobServiceDestination Configuration Data

The following parameters are available:

- External slot count (ExternalSlots)
This parameter specifies how many external processes (StudioProcessor.exe) ActiveEntry Service opens to handle process components.
- Internal slot count (InternalSlots)
This parameter specifies how many internal slots ActiveEntry Service makes available for internal process components processing.
- File with private key (PrivateKey)
Enter the file with the encryption information. The default file is "privat.key". The encryption file has to be in the installation directory of all servers with ActiveEntry Service. Use the program "Crypto Configuration" in order to create an encryption file and to encrypt the database information. You can read more in the section [Database Information Encryption](#).

NOTE!

If ActiveEntry Service finds a private key at the beginning, it stores it in a user related key container and deletes the file from the drive. If the ActiveEntry Service user account is change, you must add it again to the service's installation directory the key file.

- **ID of Job provider (ProviderID)**
Enter the name of the Job provider that will be used if more than one Job provider is being processed. If nothing is entered, the first Job provider is used.
- **Queue**
Specify the queue for processing the process steps. Each ActiveEntry Service within the network needs to have a unique queue name. Only process steps that have this exact queue name are requested from the Job queue. The queue name is defined when a Job server is added to the database. See section [Configuring a Job Server](#) for more information.
- **Timeout for process queries (RequestTimeout)**
This input specifies a time after which a process request can be said to have failed and is sent again.
Timeout format:
`day.hour:minutes:seconds`
- **Process query interval (StartInterval)**
This property defines a time interval in which ActiveEntry Service can request new process steps. The input is in seconds. Suggestions for configuring the time interval are calculated from Job server statistical data. Read the section [Job Server Statistic Information](#) in the Configuration Manual for more information.
- **Interval of time allowed for statical calculations (StatisticInterval)**
This property defines the time interval (in seconds) in which ActiveEntry Service's processing speed statistics are supplied to the database. Suggestions for configuring the interval are determined from the Job server statistical data. Read the section [Job Server Statistic Information](#) in the Configuration Manual for more information.
- **Max. reuse of external processors (MaxExternalSlotReuse)**
This value specifies how many times an external processor can be reused before the process is unloaded and restarted. The default value "0" means that the process is not unloaded until it is no longer in use.

FileJobDestination

The FileJobDestination processes the process steps that are queued by the FileJobGate (FileJobProvider or FTPJobProvider) and returns the results to the Job provider.

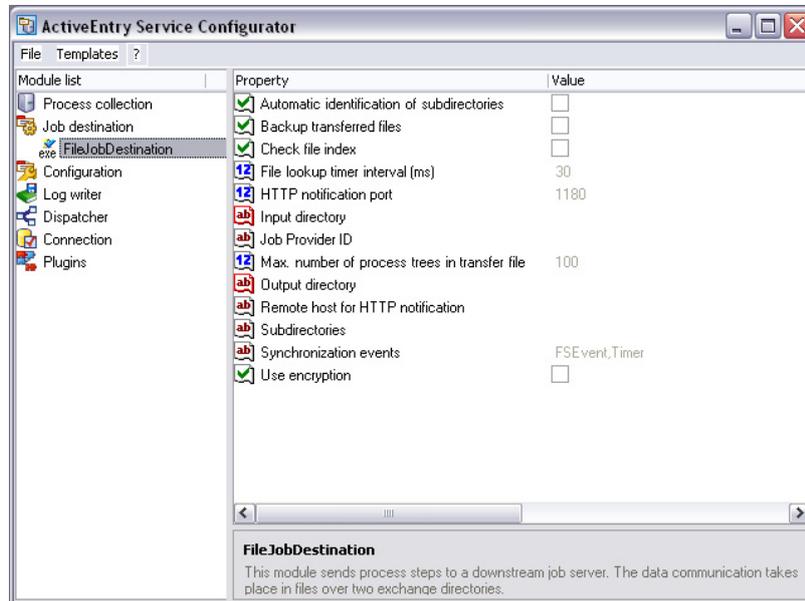


Figure 50: FileJobDestination Configuration

The FileJobDestination configuration properties correspond to those of the FileJobProvider ([Process Collection Module](#), [FileJobProvider](#)). Please note that the parameters "InputDirectory" and "OutputDirectory" need to be reversed.

There is one further parameter to be entered, ProviderID. If more than one Job provider is being processed, enter the name of the Job provider that is going to be used. If the input is empty, the first Job provider is used.

FTPJobDestination

FTPJobDestination processes the process steps that are queued in the FileJobGate (FileJobProvider or FTPJobProvider) and returns the results to the Job provider.

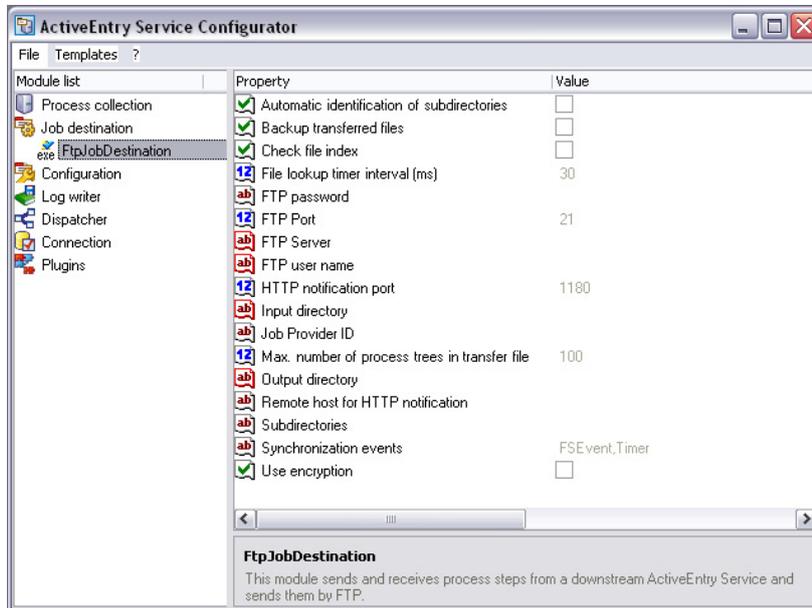


Figure 51: FTPJobDestination Configuration Data

The FTPJobDestination configuration properties correspond exactly to those of the FTPJobProvider ([Process Collection Module](#), [FTPJobProvider](#)). Please note that the parameters "InputDirectory" and "OutputDirectory" need to be reversed.

There is one further parameter to be entered, ProviderID. If more than one Job provider is being processed, enter the name of the Job provider that is going to be used. If the input is empty, the first Job provider is used.

HTTPJobDestination

An HTTPJobDestination sends process steps to a child Job server. The data transfer is carried out by HTTP.

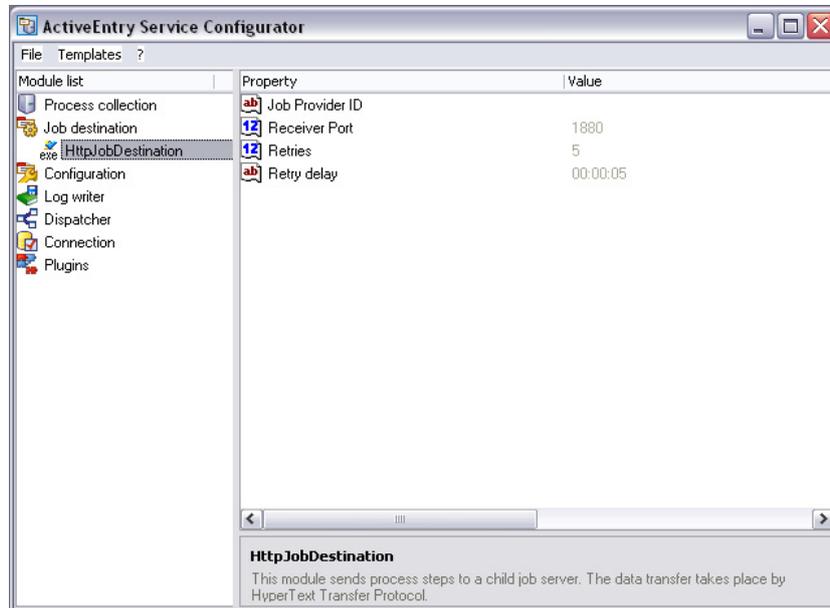


Figure 52: HTTPJobDestination Configuration Data

Configure the following parameters:

- Recipient port (ChildPort)
Enter the HTTP-Port of the child server.
- ID of the Job provider (ProviderID)
Enter the name of the Job provider that will be used if more than one Job provider is being processed.
- Number of retries (Retries)
This value defines how many times the module retries the data transfer if it fails.
- Time interval between retries (RetryDelay)
This time delay defines how long a module waits after a failed process transfer before re-trying.
Time delay format:
`day.hour:minutes:seconds`

4.6.6 The Configuration Module

The standard ActiveEntry Service configuration settings are specified in the module.

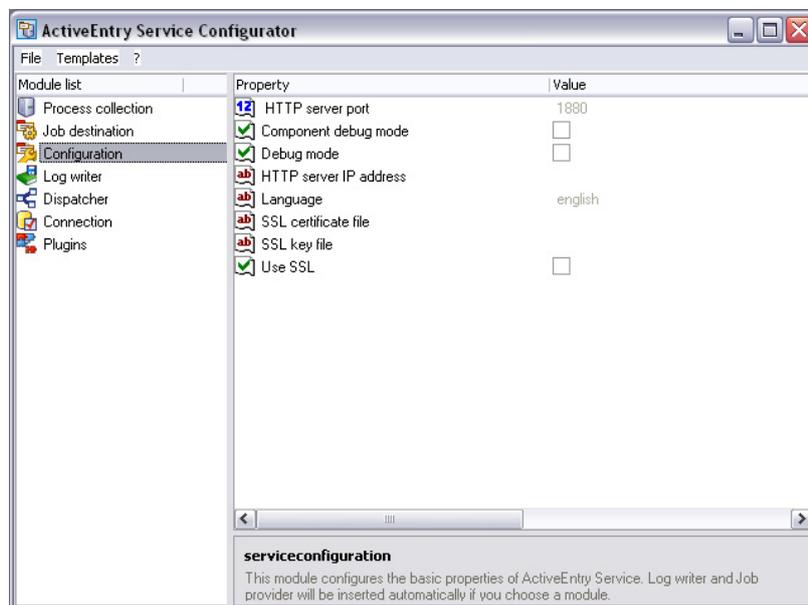


Figure 53: General Configuration Settings

The following parameters are available:

- **Debugging mode (DebugMode)**
In DebugMode ActiveEntry Service writes additional information to the log file. For example, all the parameters and results that are passed to a component are written to the log file. Read the section [ActiveEntry Service Extended Debugging](#) in the Process Orchestration Manual for more information about DebugMode.
- **Component debugging mode (ComponentDebugMode)**
When set, individual ActiveEntry Service process components write additional process information to a log file. The ComponentDebugMode localizes errors and it is not recommended during normal work hours because system performance is affected. Read the section [ActiveEntry Service Extended Debugging](#) in the Process Orchestration Manual for more information about using ComponentDebugMode.
- **IP address of the HTTP server (HTTPAddress)**
If ActiveEntry Service is running on a computer with several network cards, you can use this parameter to define which service should work over which IP address. If no IP address is entered, then all of them are used.
- **HTTP server port (HTTPPort)**
Each ActiveEntry Service automatically acts as an HTTP server. This parameter specifies the port that ActiveEntry Service works with. The port 1880 is the default value. The HTTP Server communicates via:
`http://server name:port number/service`

E.g.:

`http://Servername:1880/log`

The service that is available to ActiveEntry Service depends on the configuration of the plug-ins. The HTTPStatusPlugin provides, for example, the status display and the log display (see [The Plug-in Module](#), [HTTPStatusPlugin](#)).

- **Language**
This parameter specifies the language for ActiveEntry Service error messages and output. Permitted input is "deutsch" or "english". The default is "English".
- **Use SSL (UseSSL)**
Set this option if an HTTP Server secure connection is available. The server is accessed over HTTPS in the browser.
- **SSL certificate file (SSLCertificate)**
If the server communicates over SSL (Secure Sockets Layer) the name of the certificate needs to be entered here.
- **SSL key file (SSLKey)**
Enter the name of the encryption key when communicating via SSL.

4.6.7 The Log Writer Module

This module writes the ActiveEntry Service messages. The following module types are available:

- EventLogLogWriter
- FileLogWriter

EventLogLogWriter

This module writes ActiveEntry Service log events to the log file (EventLog)

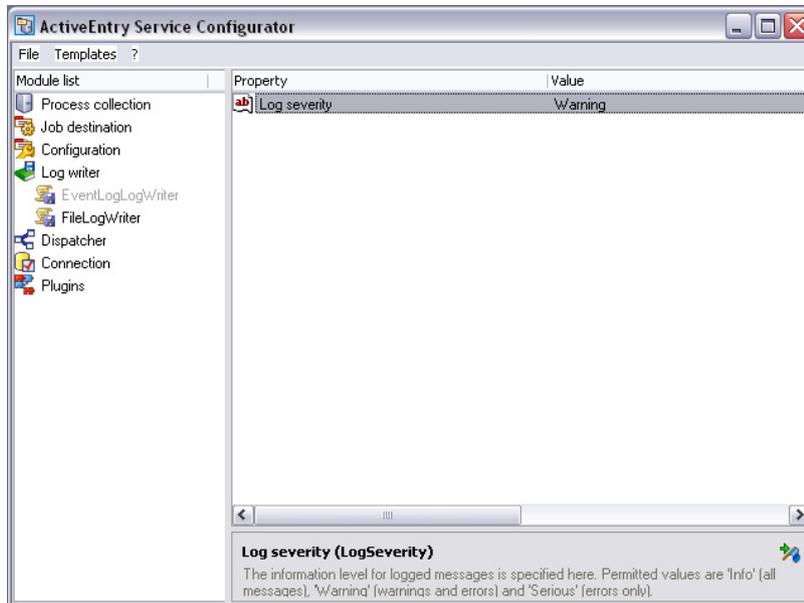


Figure 54: Log Writer Configuration Data

The following parameters available in this module are:

- Severity level (LogSeverity)
Specifies the warning level for logging messages.

Table 5: Message Types

Warning Level	Description
Info	All messages are written to the log file. The log file quickly becomes large and confusing.
Warning	Only warning and serious errors appear in the log (default).
Serious	Only serious errors are written to the log file (exceptions).

FileLogWriter

The FileLogWriter writes ActiveEntry Service messages into a log file. The log file can be displayed in a browser. Prerequisite is configuration of the HTTPStatusPlugins (see [HTTPStatusPlugin](#)).

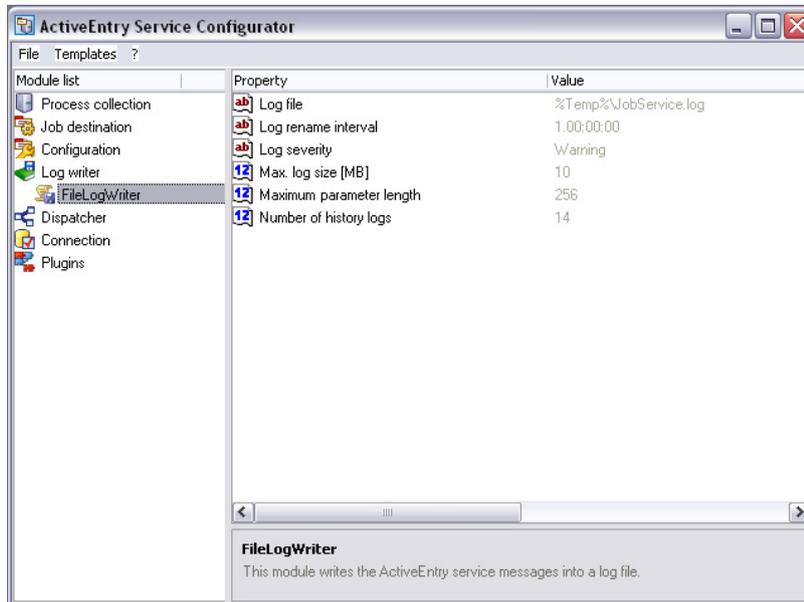


Figure 55: FileLogWriter Configuration File

The following parameters are available:

- OutPutFile**
 The log information for ActiveEntry Service is written to this file. Ensure that the given directory exists. If the file cannot be created, no error output is possible. In this case, the error messages appear in the windows event log or, under Linux, in /var/log/messages.
- LogLifeTime**
 In order to avoid unnecessarily large log files, the module supports the functionality of exchanging the log file with a history list. The LogLifeTime specifies the maximum life of a log file before it is renamed as backup. If the log file has reached its maximum age, the file is renamed (i.e. as "JobService.log_20040819-083554") and a new log file is started. When ActiveEntry Service is stopped or started the LogLifeTime is reset.
 Time format:
`day.hours:minutes:seconds`
- Max. number of archived log files (HistorySize)**
 This attribute limits the number of log files. If several log files exist, the oldest backup file is deleted when a new log file is created so that the limit is not exceeded.
- Max. log file size (MB) (MaxLogSize)**
 Use this parameter to specify the maximum size for the log file. Once the log file has reached the limit, it is renamed into a backup file and a new log file is created.

- Max. length of the parameter (ParamMaxLength)
This parameter defines how many character can be in a job so that it is still written to the log file.
- Severity level (LogSeverity)
Specifies the warning level for logging messages.

Table 6: Message Types

Warning Level	Description
Info	All messages are written to the log file. The log file quickly becomes large and confusing (default when using templates).
Warning	Only warnings and serious errors appear in the log (initial default).
Serious	Only serious errors are written to the log file (exceptions).

4.6.8 The Dispatcher Module

In a hierarchical server structure a server can be used as a proxy server for other servers. The proxy server makes requests at set time intervals for process steps to be processed on a server and sends them to the next server. If the request load needs to be minimized, a proxy server is recommended.

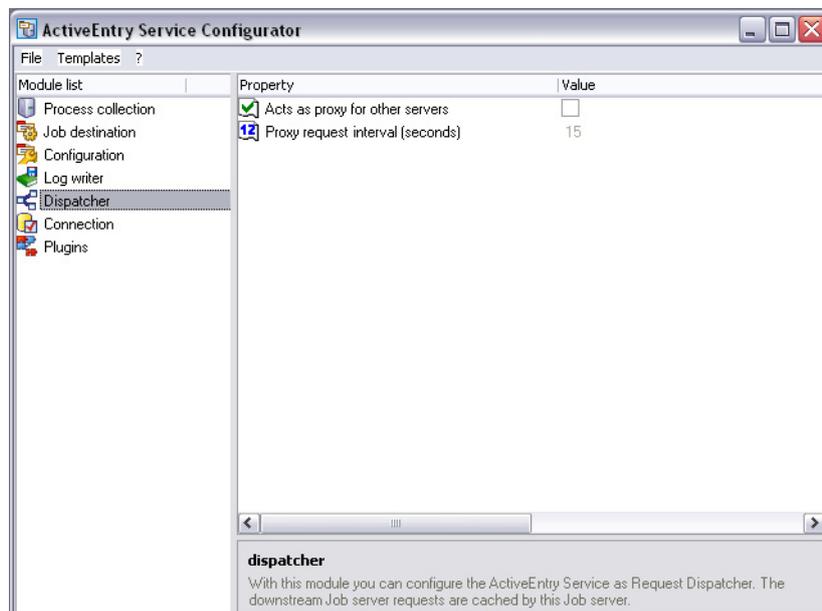


Figure 56: Dispatcher Configuration Data

This module has the parameters:

- Acts as proxy for other servers (IsProxy)
This parameter specifies if a server is acting as a proxy server. Set this option if the server should be a proxy server.

- Proxy request interval (ProxyInterval)
The ProxyIntervall sets the time interval in seconds, after which the proxy server acting as deputy for another server, should renew a request to the database.

The following guidelines can be used as orientation for the configuration of ActiveEntry Service polling intervals in a cascading environment:

Table 7: Polling Interval Guidelines for ActiveEntry Service

Parameter	Root Server (direct connection to database)	Leaf Server (connected via HTTP/File/etc.)
JobServiceDestination.Startinterval	15 seconds	600 seconds
JobServiceDestination.Statisticinterval	60 seconds	600 seconds
Dispatcher.ProxyInterval	30 seconds	
Dispatcher.IsProxy	True	False

The proxy mode of a root server ensures that, acting on behalf of the leaf server, process steps are queried in shorter proxy intervals. When a root server is restarted it can take a while until all the leaf servers have send their first requests (in this case max. 6000 seconds), but then the system takes over.

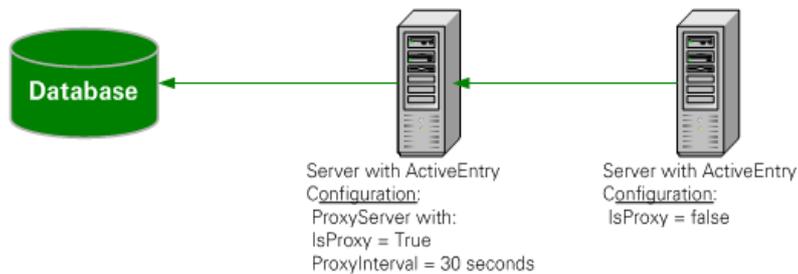


Figure 57: Dispatcher Configuration Example

4.6.9 The Connection Module

This module configures special behavior settings for ActiveEntry Service.

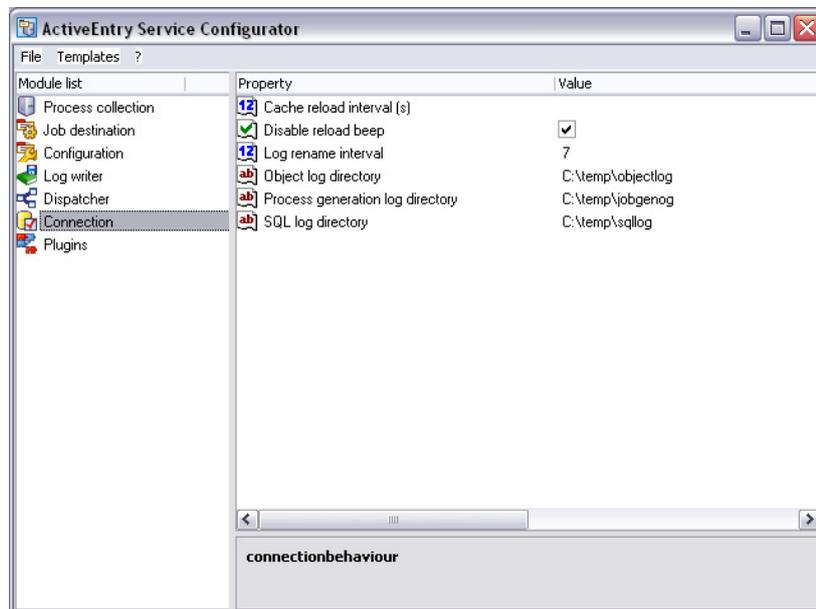


Figure 58: Connection Configuration Data

The parameters in this module are:

- Directory for generating logging (JobGenLogDir)
Log files are created in this directory that record process generation instructions from ActiveEntry Service. Read more in the section [Process Generation Logging](#) in the Process Orchestration Manual.
- Directory for object logging (ObjectLogDir)
Log files are created in this directory that record the object actions from ActiveEntry Service. Read more in the section [Object Action Logging](#) in the Process Orchestration Manual.
- Directory for the SQL log (SQLLogDir)
Log files are created in this directory that record the database requests from ActiveEntry Service. Read more in the section [Database Query Logging](#) in the Process Orchestration Manual.
- Renaming interval for the log file (LogLifeTime)
Use this parameter to specify how many many days the log files (generating log, object log, SQL log) should be stored. During database connection, ActiveEntry Service checks whether old log files still exist and deletes them from the given directories. This prevents an unnecessarily large number of log files.
- Suppress reload beep (NoReloadBeep)
When this parameter is set the beep is switched off that is made when buffered dialog data is loaded.

4.6.10 The Plug-in Module

Plug-ins are program classes that ActiveEntry Service loads and that extend the functionality of the service. The following plug-ins are available:

- HTTPStatusPlugin
- HTTPLogPlugin
- ShareInfoPlugin
- StatisticsPlugin
- ScheduleCommandPlugin
- DBSchedulerWatchDogPlugin
- RequestWatchDogPlugin
- PerformanceCounterPlugin

HTTPStatusPlugin

The HTTPStatusPlugin extends ActiveEntry Service with a different services. This plug-in needs no further parameters.

Calling syntax:

```
http://servername:1880/Assemblies  
http://servername:1880/Cache  
http://servername:1880/Comp  
http://servername:1880/Log  
http://servername:1880/Statistics  
http://servername:1880/Status
```

HTTPLogPlugin

The HTTPLogPlugin writes a log file that records the ActiveEntry Service HTTP requests.

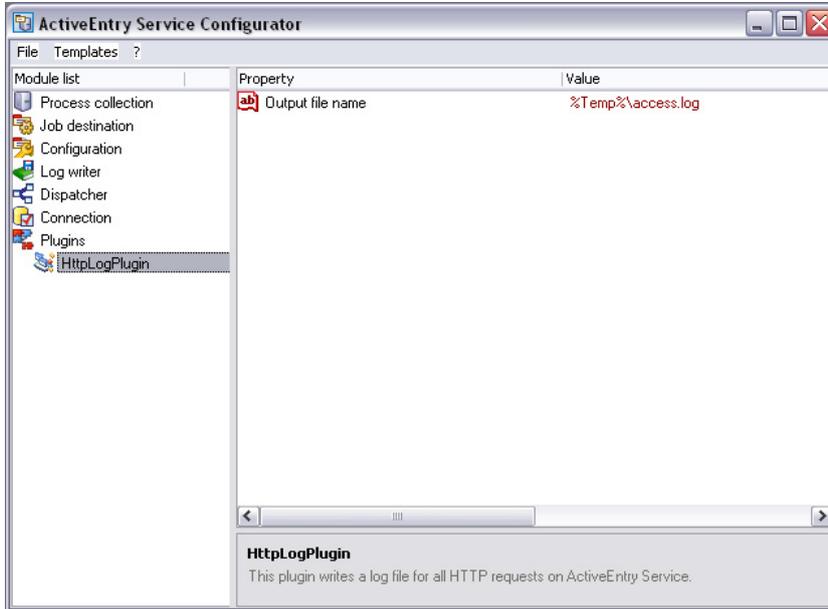


Figure 59: HTTPLogPlugins Configuration Data

Enter the following parameter:

- Output file (LogFile)
Enter the name of the file that is to record the messages. The file is written in Apache HTTP Server Combined Log Format.

StatisticsPlugin

This plug-in monitors the properties of ActiveEntry Service over a longer time period and supplies statistics data.

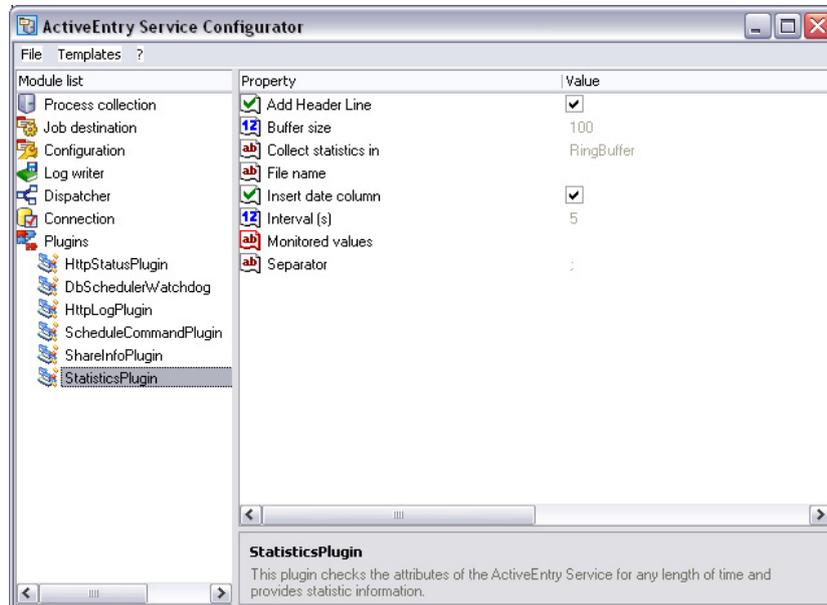


Figure 60: StatisticsPlugins Configuration

Following parameters are available:

- Values to be monitored (Columns)
Specifies which ActiveEntry Service attributes the plug-in should monitor. The input is in list form, separated by commas.
Example for a request for buffered information:
`Destinations\abarbeiter\Cache\PrognostedDuration, Destinations\abarbeiter\Cache\CacheLimit`
- Interval (Interval)
This is where the time interval for reading out and saving the monitored ActiveEntry Service attributes is specified. The input is given in seconds.
- Collect statistics to (CollectTo)
Specifies in which for the measurements are saved. Permitted input is "CVS" or "RingBuffer"
- Size of buffer (BufferSize)
Specifies how many time measurements can be saved to the circular buffer..
- Name of file (FileName)
If the measurements are written in a CSV file, the output file (with directory path) is entered here.
- Delimiter (Separator)
Specifies the character used to separate measurement values in CSV format.

- show date column (ShowDates)
If this option is set, the measurement times are added to the CSV output.
- Insert header (ShowHeaders)
If this option is set, The column titles and added to the CSV output.

ScheduleCommandPlugin

This plug-in calls up an external program in regular intervals. This is useful, for example, when process steps need to be routed over their own transfer methods (e.g. SSH).

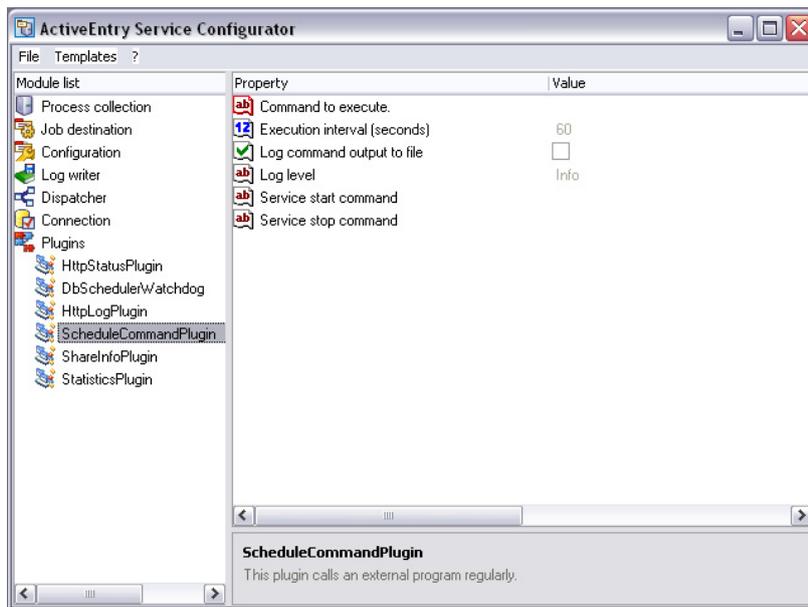


Figure 61: ScheduleCommandPlugin Configuration Data

The following parameters are necessary:

- Run command (Command)
This parameter defines the command that is to be carried out, including the command line options. This will be executed as a cmd, and therefore built-in commands are possible.
- Service start Command (StartCommand)
This command is run when ActiveEntry Services is started.
- Service stop command (StopCommand)
This command is run when ActiveEntry Services finishes.
- Interval (Interval)
This parameter specifies how often the command should be called. The input is in seconds. While the command is running, the timer is stopped so that the calls do not overlap.

- Command output to log file (OutputToLog)
If this parameter is set the all command output is written to file. Otherwise only errors are written to the log file.
- Severity level (LogSeverity)
This is where the warning level is given with which the warnings appear in the log file. Permitted are "Info", "Warning" and "Serious".

DBSchedulerWatchDogPlugin

This plug-in checks, at defined intervals, if a database schedule for the DBScheduler is enabled and starts it if necessary. The plug-in should only be running on one Job server in the network. We recommend running it on the database server.

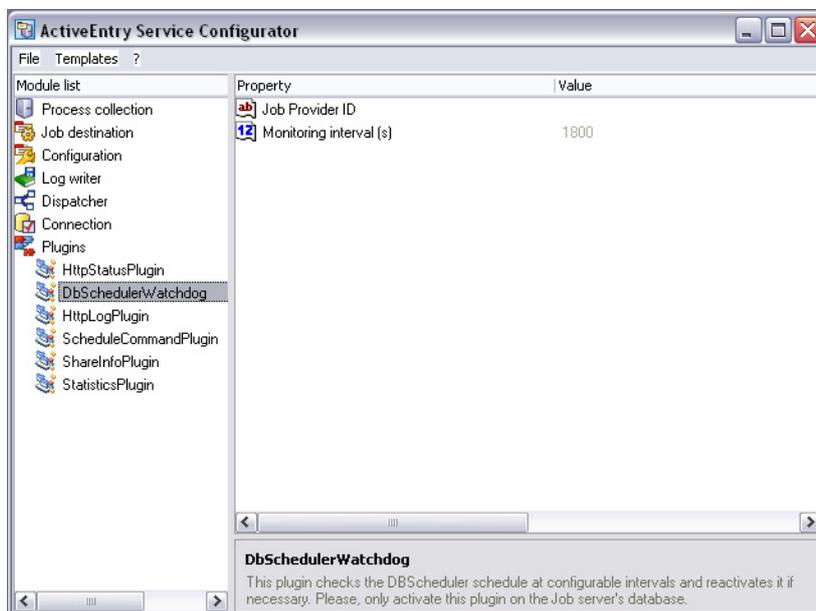


Figure 62: DBSchedulerWatchDogPlugins Configuration

The following parameters are required:

- Monitoring interval (Interval)
This parameter specifies how often, in time intervals, the DBScheduler is checked. The input is given in seconds.
- ID of the Job provider (ProviderID)
This parameter contains the ID of the Job provider that will be in use. If there is no entry, the first Job provider is taken.

RequestWatchDogPlugin

This plug-in restarts ActiveEntry Service when less than a defined number of requests are made within a specified interval.

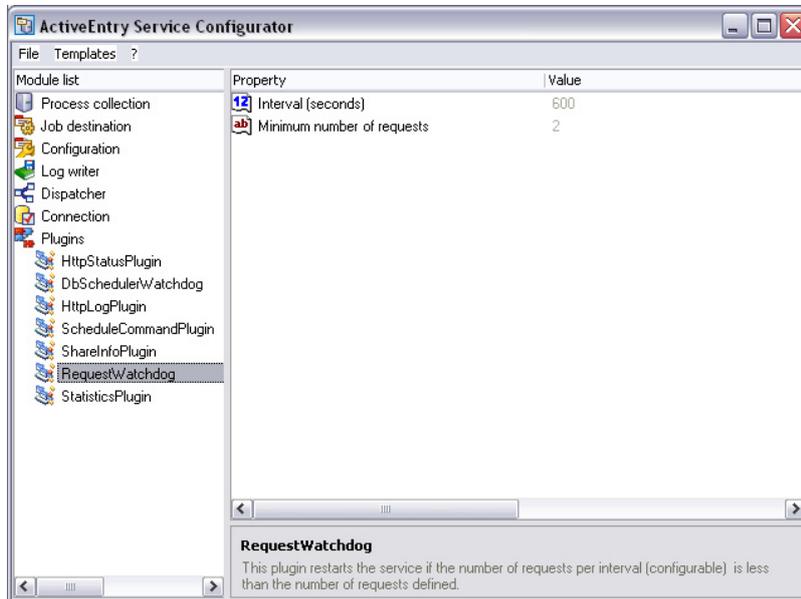


Figure 63: RequestWatchDogPlugins Configuration Data

Enter the following parameters:

- **Monitoring interval (Interval)**
This parameter sets the monitoring interval. The input is in seconds.
- **Min. number of requests (MinRequests)**
This parameter contains the minimum number of requests that need to be made within the interval. When setting this value, take into account that a DBSchedulerWatchDogPlugin is possibly in use that also posts requests.

PerformanceCounterPlugin

This plugin exports the ActiveEntry Service status values as performance counter. This makes monitoring via a system monitor (e.g. Perfmon) possible.

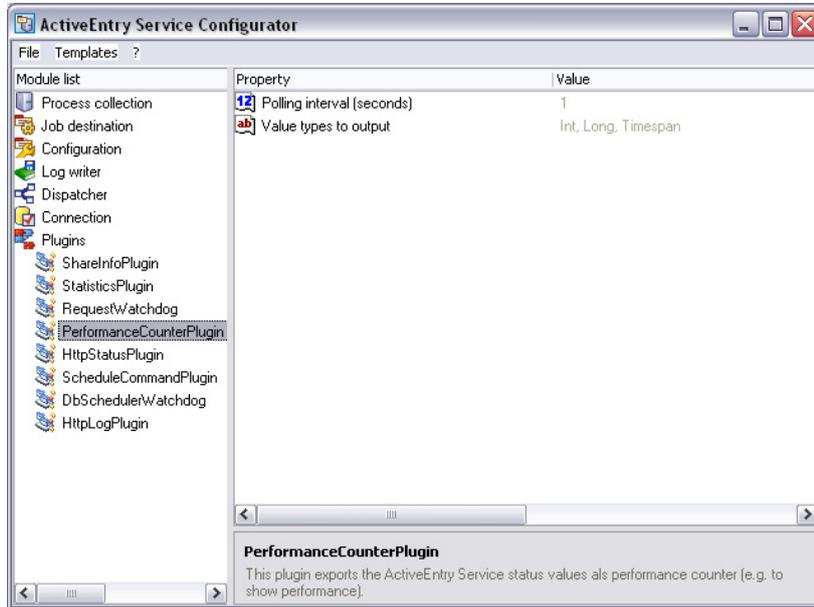


Figure 64: PerformanceCounterPlugins Configuration Data

Parameters to enter are:

- Value types to specify (CounterType)
Use this parameter to specify which value types are made available as performance counters. Int and long values should be entered directly, time values should be entered as long values (number of milliseconds).
- Polling intervall (PollingInterval)
This parameter to specify the interval for exporting the performance counter. Input is in seconds.

TIP

If the error "At least one service could not be started" occurs after re-starting ActiveEntry Service, then make the WMI Performance Adapter service a dependency of ActiveEntry Service.

4.6.11 ActiveEntry Service Installation and Startup

When ActiveEntry Service is installed with the ActiveEntry.Net Setup Wizard the service is already entered into the "Services" system administration tool.

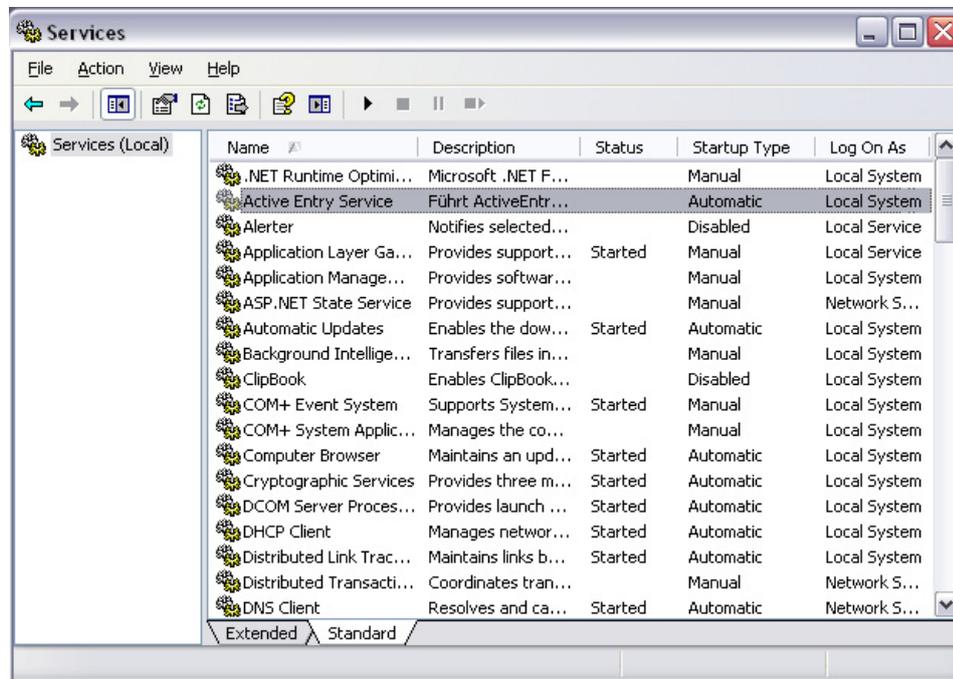


Figure 65: ActiveEntry Service Entry

You can change the login data and if necessary the startup method. Then you can start ActiveEntry Service. An appropriate message is written in the event log if you cannot start ActiveEntry Service.

Installing and Uninstalling ActiveEntry Service from the Command Line

Manual installation and deinstallation of ActiveEntry Service can be done with the program "installutil.exe". You start the program over the command line in the program directory. All parameters, possibilities and the calling syntax for the program "installutil.exe" are displayed if you use the help parameter "/" or "/help".

In the following, only the parameters for starting ActiveEntry Service are described.

Table 8: Calling Parameters for the Service Startup

Parameter	Meaning
/username	User account name (with domain)
/password	User account password
/unattended	Unattended Installation (without prompting for user and password)

If the parameters /username and /password are not given, then a login mask appears for the installation service, in which the user account and its password must be entered.

Calling example:

```
installutil.exe vinetworkservice.exe
```

ActiveEntry Service is added to the service administration of the computer and needs to be restarted manually.

4.6.12 ActiveEntry Service in a Cluster

The idea of a cluster solution is to make the system highly available. The aim is to limit system failure to only a few seconds if a hardware or software component fails. This can be attained with the installation of a Microsoft Windows cluster solution (only possible with Enterprise Servers). The following diagram shows such a solution.

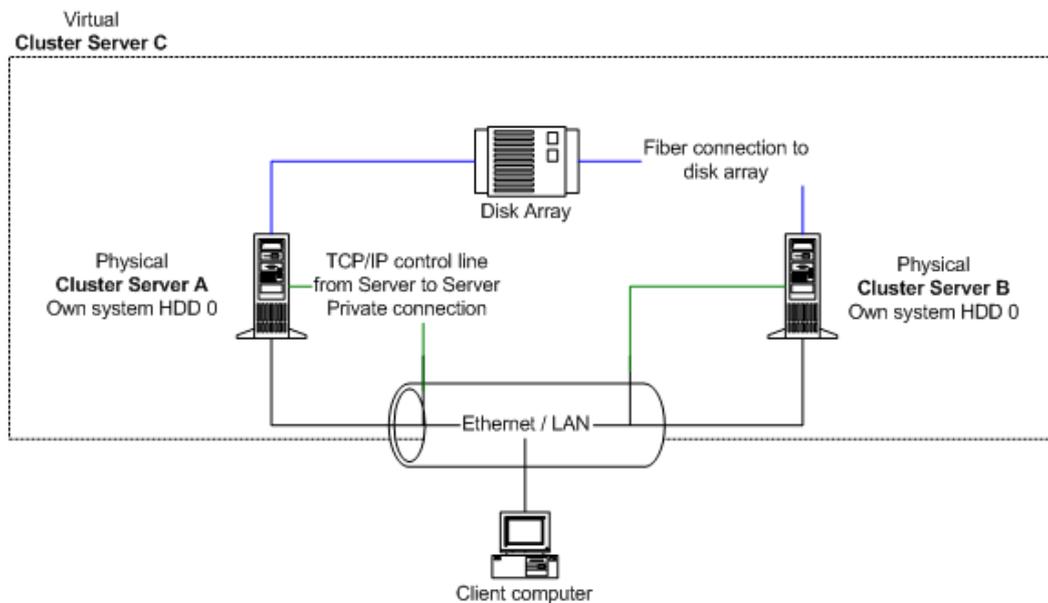


Figure 66: Example of a Cluster Solution

This cluster is made up of 2 physical computers "Server A" and "Server B" that use the same disk array and have their own individual system hard drive. Each server has the operating system Windows 2000. Both servers are installed identically so that in the case of failure one server can take over from the other.

All redundant system components are managed by the cluster manager.

From an external point of view, the cluster is addressed as a single, virtual server "Server C". The service or user that is accessing the service is automatically connected to the physical server that is currently carrying out the work in the cluster.

If one of the servers fails, then the redundant server in the cluster automatically takes over. The virtual server remains the contact partner, only the physical server that is running, changes.

Registering ActiveEntry Service in a Cluster

When ActiveEntry Service registers in a cluster, it is subject to cluster handling for reliability and load balancing. The service is installed on the virtual server that is simulated by the cluster. All computer related operations and service information go to the virtual server, this is transparent for the service, instead of to the real computer (cluster nodes). This is also valid for the clients that contact the service using the server name, e.g. via RPC (ORPC, DCOM), TCP/IP (Winsock, Named Pipes), HTTP.

Because the service is acting in the context of a virtual server, the following facts need to be noted:

- Service specific settings for nodes where the virtual server is found, are reproduced on all the other nodes! The service, therefore, is always started from the same configuration.
- The service is only ever started on the actual virtual server node (the node that is running the virtual server). The service is halted on all other nodes.
- The service is started up and terminated with the virtual server. If the cluster is inactive, the service is stopped on all the nodes.
- Before registering, the services on the nodes are automatically put into the correct state (Manual and Stopped).

ActiveEntry Service Installation and Cluster Configuration

The installation of ActiveEntry server components from the setup CD needs to be done on all the physical nodes of the cluster. The installation is carried out by ActiveEntry.Net Setup Wizard. Read more in section [ActiveEntry.Net Setup Wizard](#).

Subsequently the ActiveEntry Service configuration is carried out via the JobService Configuration as shown in section [Configuring ActiveEntry Service](#). At the same time, note that the parameter "Queue" contains the name of the virtual server when the JobServiceDestination is configured.

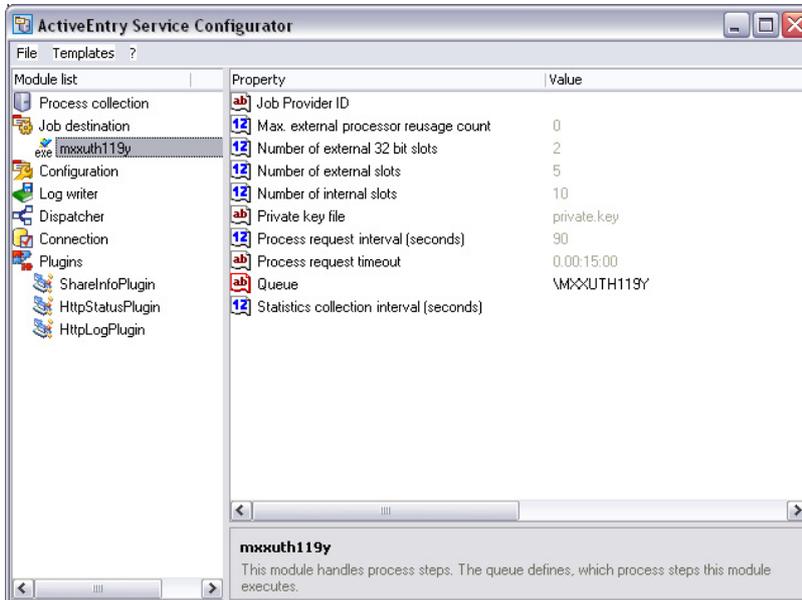


Figure 67: Setting the JobDestination in the ActiveEntry Service Configuration File

After saving the configuration, the configuration file in the ActiveEntry installation directory needs to be copied to all the physical nodes. The name of the configuration file may not be changed.

NOTE!

The ActiveEntry Service Configuration is not part of a cluster resource. Thus, each node keeps its own configuration. For this reason, it is necessary to ensure that the configuration files on the physical nodes are consistent. If this is not the case, correct functionality cannot be guaranteed after changing cluster nodes.

Setting up the ActiveEntry Service Cluster Resource

The Cluster Administrator is responsible for setting up a new cluster resource. It is irrelevant which physical node the cluster is currently running on. The following figure shows a cluster resource setup in ActiveEntry Service.

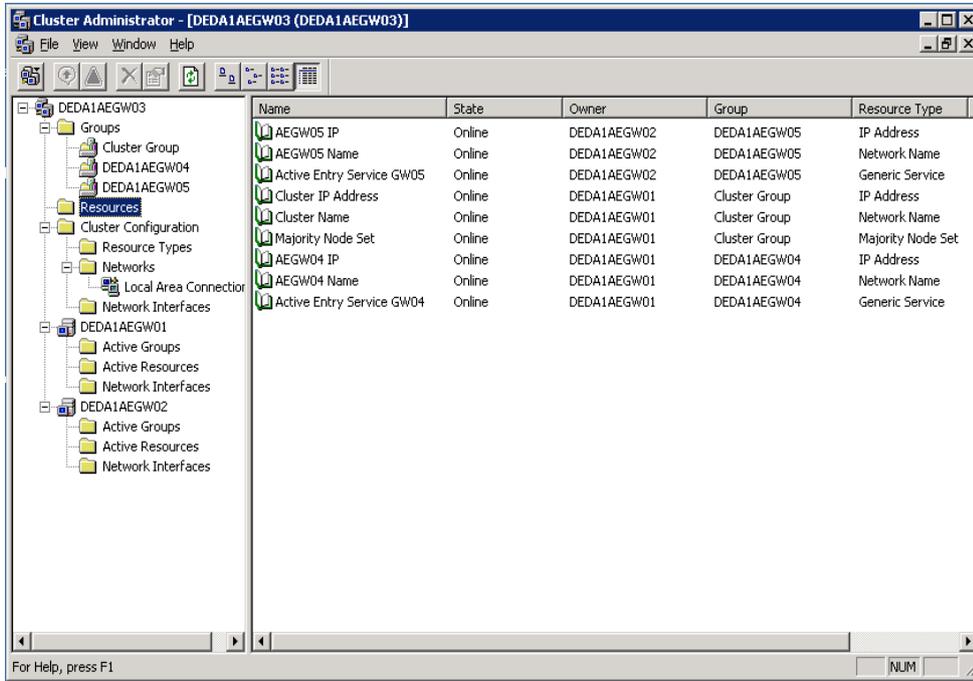


Figure 68: Cluster Administrator

Using the right mouse button you can create a new cluster resource over the menu item "New\Resource". Specify the name of the cluster resource and select "Generic Service" as cluster resource type.

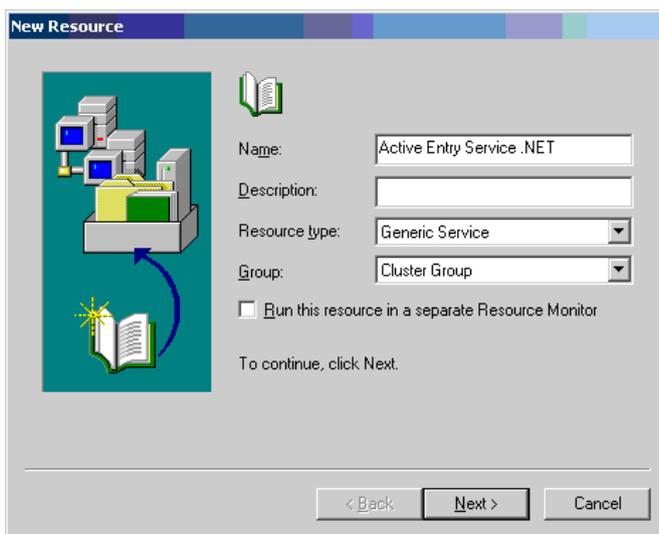


Figure 69: Adding a new Cluster Resource

Add all the physical cluster nodes that ActiveEntry Service can be run on.

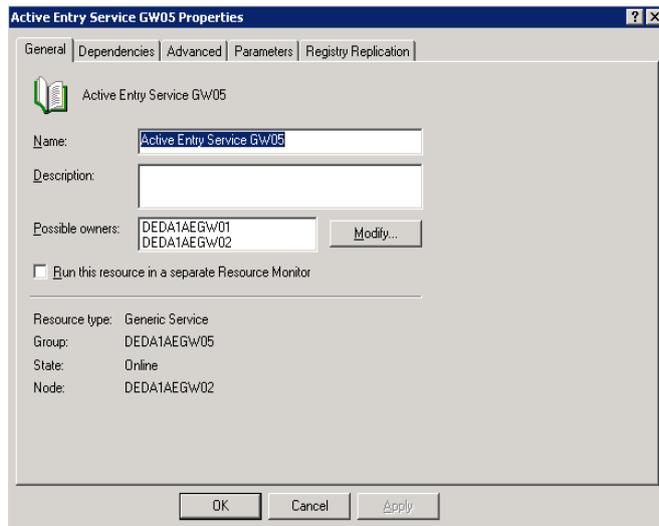


Figure 70: Adding Physical Nodes

The ActiveEntry Service dependencies need to be specified. At least the following dependencies need to be selected:

- Cluster IP address
- Cluster name
- Quorum (e.g. disk: D)

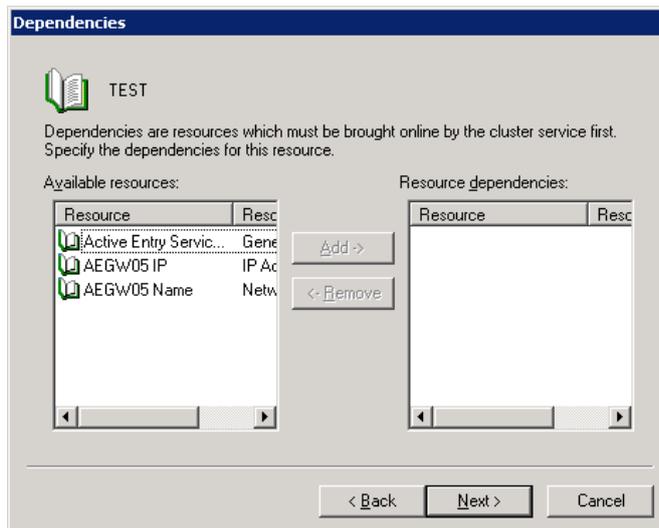


Figure 71: ActiveEntry Services Dependencies

Enter the registry name for ActiveEntry Service.

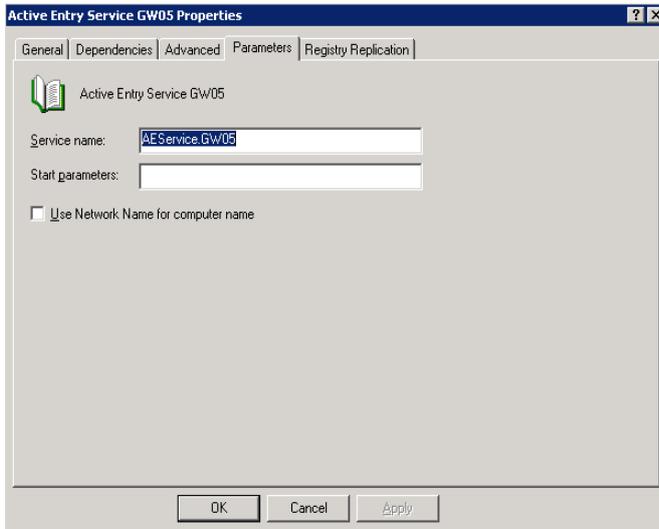


Figure 72: ActiveEntry Service, Service Name

No further registry keys are required.

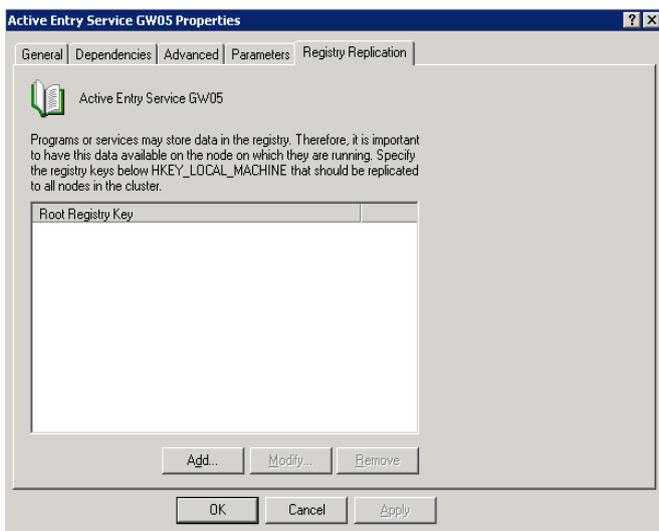


Figure 73: Additional Registry Key

In the next step the cluster resource is created.

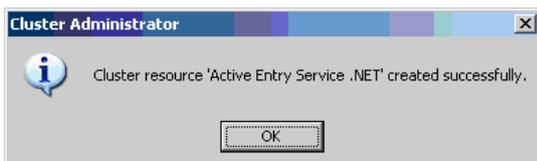


Figure 74: Cluster Resource Success Alert

After the cluster resource has been successfully created it can be put online.

NOTE! After setting up ActiveEntry Service in a cluster network it is a good idea to simulate a fail situation so that preventable problems do not occur when the service is first put in to live operation.

Storing the ActiveEntry Service Log file on a Shared Volume

Create a directory with the name "AELog" on the shared volume of the virtual server. Then add a cluster resource in the cluster manager in the following sequence.

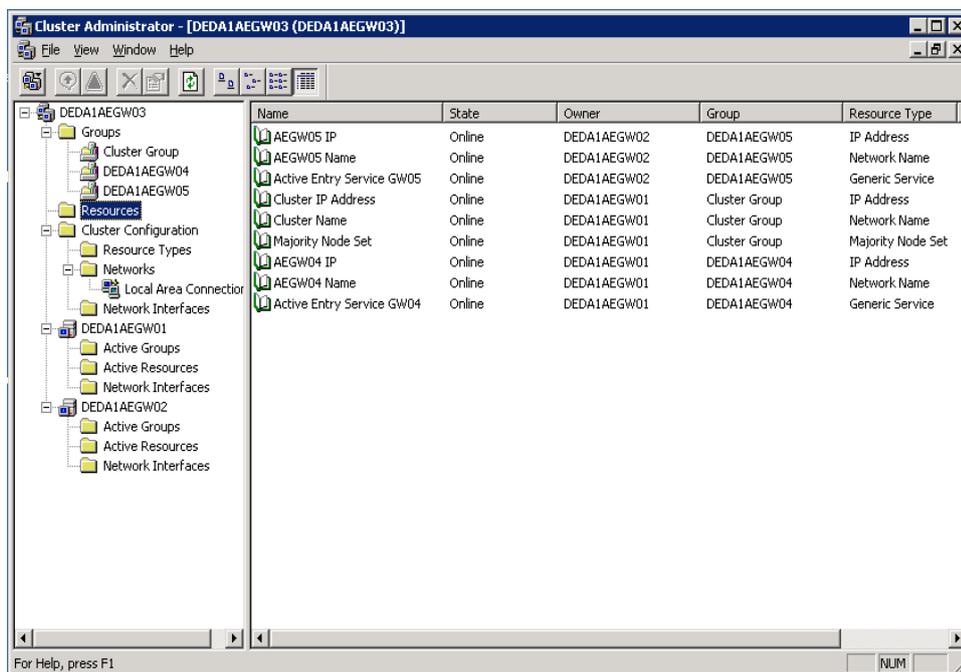


Figure 75: Cluster Administrator

Using the right mouse button, select the menu item "New\Resource" to create a new cluster resource. Specify the name of the cluster resource and select "File Share" as cluster resource type.

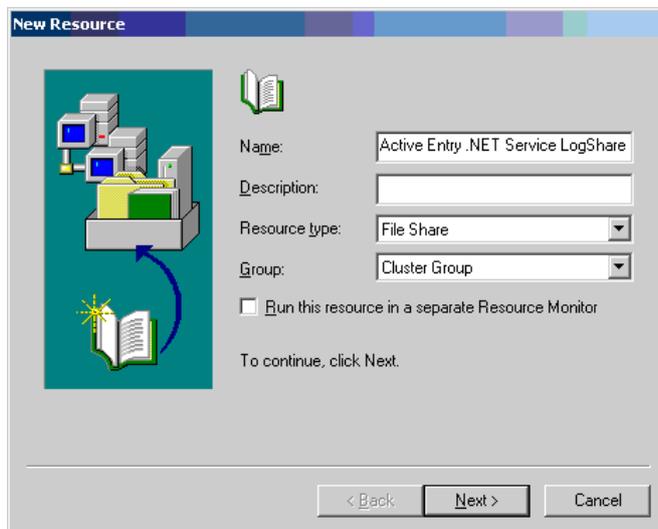


Figure 76: Adding a New Cluster Resource

Add the physical cluster node that the cluster resource is able to run on.

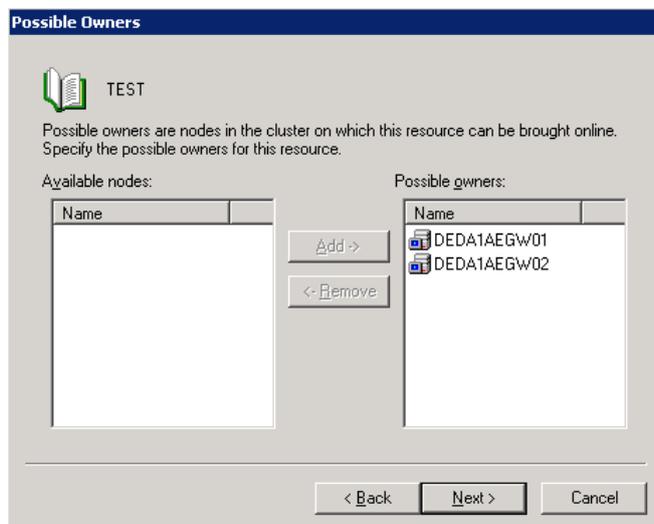


Figure 77: Adding the Physical Nodes

The resource dependencies need to be specified. At least the following are required:

- ActiveEntry Service.NET

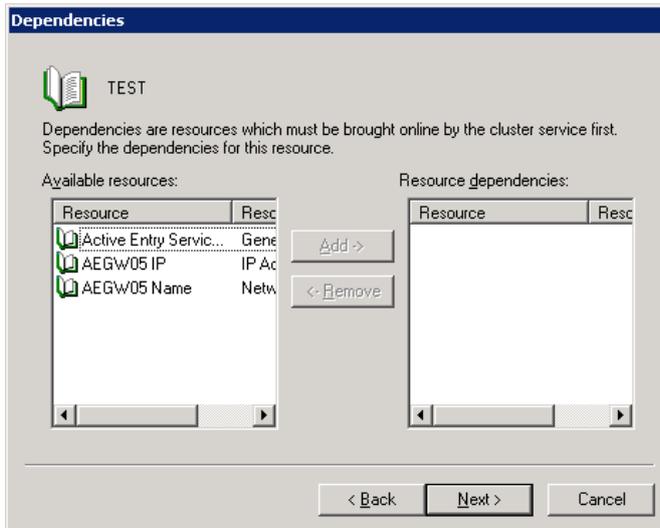


Figure 78: Cluster Resource Dependencies

The share name and the log file path as well as the access restrictions need to be entered.

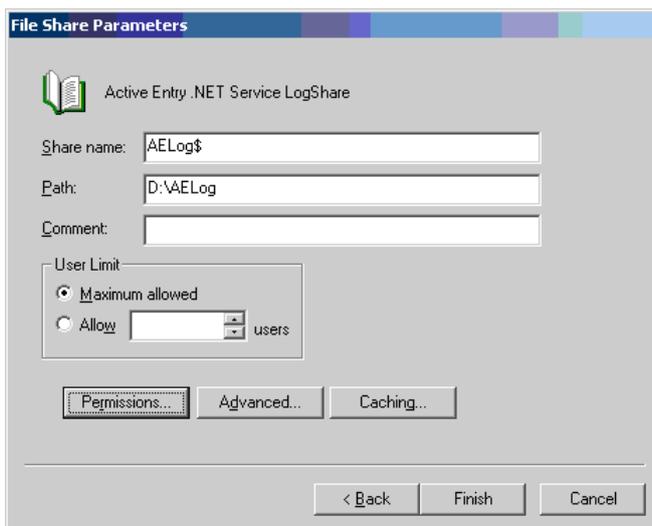


Figure 79: Cluster Resource Parameters

Set security permission for access to the log file.

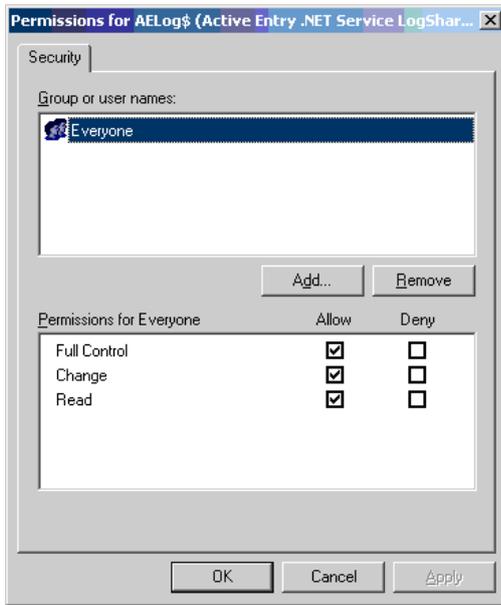


Figure 80: Access Rights to the Share

In the next step, the settings to regulate cache behavior are made.

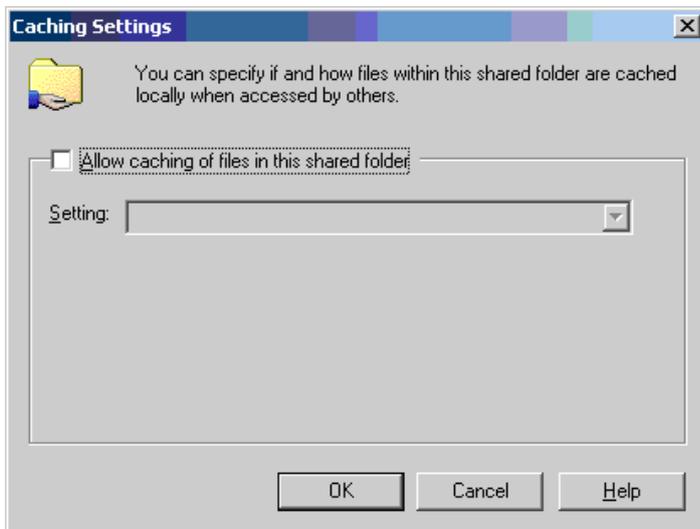


Figure 81: Cache Settings

Then the cluster resource is created.



Figure 82: Cluster Resource Creation Success Alert

After the cluster resource had been successfully created it can be put online. Then the Logwriters parameter "OutPutFile" in the ActiveEntry Service configuration file needs to be entered (see [The Log Writer Module](#)).

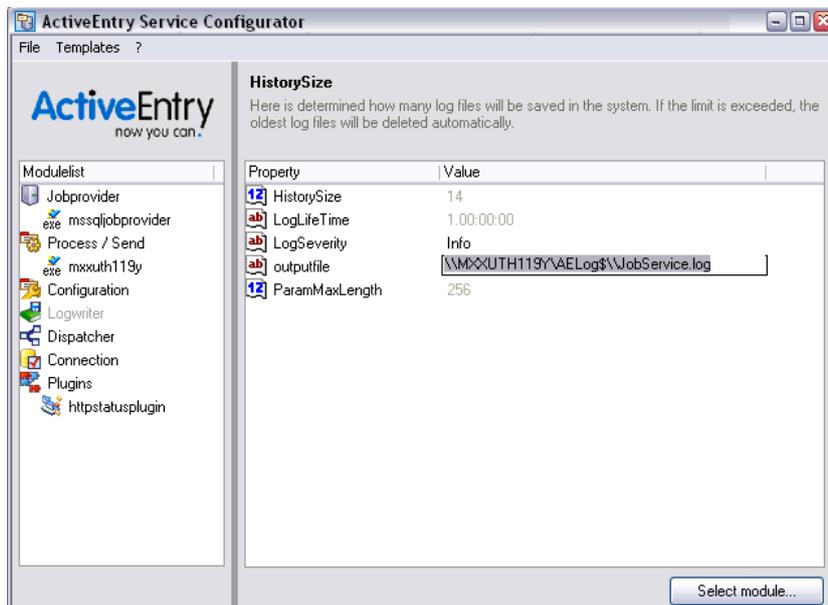


Figure 83: Changing the Logwriter in the ActiveEntry Service Configuration File

The configuration file is then saved and needs to be copied to all the physical nodes of the cluster in the ActiveEntry Service installation directory.

4.7 Updating ActiveEntry Tools

Updating ActiveEntry tools includes updating the ActiveEntry database and the existing installations on ActiveEntry network workstations and servers.

Database updates are necessary when hotfixes and service packs or complete version updates are available for ActiveEntry. Any customer specific changes also need to be transferred from the development database to the productive system's database.

Manually installing and updating software locally poses problems due to the physical distribution of servers and workstations between rooms. In order to ensure an acceptable workload for the network administrators, an automatic method for updating ActiveEntry tools has been developed. In addition to updating established ActiveEntry installation files, new customer specific files can be added with this method by simple means and, therefore, distributed to workstations and servers in an ActiveEntry network using automatic software updating.

Automatic software update is normally the default method for updating ActiveEntry tools on the job servers. However, the update method takes into account that individual servers may need to be excluded from the automatic update in certain circumstances and updated manually.

Hotfixes and service packs for the main version or even a completely new version are issued by Völcker Informatik for updating ActiveEntry tools.

Hotfix

A hotfix contains corrections to the default configuration of the current main version but no extension of functionality.

Service Pack

A service pack contains minimal extensions of functionality and all the modifications since the last main version that were already included in the hotfixes.

Version Change

A version change is connected with significant extensions of functionality and involves a completely new installation.

4.7.1 Updating the Database

The ActiveEntry database is customized by loading so-called 'transport packages'. ActiveEntry recognises the following types of transport packages that can be copied to the database depending on requirements:

- Migration package
Migration packages are provided by Völcker Informatik for the initial database migration, for service pack and complete version updates. A migration package contains all the necessary tables, data types, database procedures and the default ActiveEntry configuration. When a migration package is imported the migration version is changed in the database. Migration package import is done by the "DBMigrator" and is described in detail in section [Migrating an ActiveEntry Database](#).
- Hotfix packages
Hotfix packages are provided by Völcker Informatik to load individual corrections to the default configuration such as templates, scripts, processes or files. When a hotfix is installed, the migration version in the database is tested but not changed. Hotfix package import is done by the "ActiveEntry Transporter" and is described in detail in section [Creating a Customer Configuration Package](#).
- Customer configuration package
A customer configuration package is used to exchange customer specific changes between the development, test and productive system database. This transport package is created by the customer and loaded into the database. When a customer configuration package is imported, the migration version is tested in the database but not changed. Confi-

guring and importing a customer configuration package is done with the program "ActiveEntry Transporter" and is described in section [Creating a Customer Configuration Package](#).

NOTE!

Take the following into account when using the ActiveEntry IT Shop: After you have updated your database by migrating, you need to share the project again that your web application is based on. Refer to the section "Sharing Projects Dialog" in the Web Designer Reference.

When a transport package is imported into an ActiveEntry database the following operations are carried out:

- **Insert**
No data set was found in the destination database using an alternative key, a new data set is created with this key value.
- **Update**
If data set is found in the destination database using an alternative key, this data set is updated.
- **Delete**
Data sets that are no longer needed are deleted. The "Delete" operation is only executed if a migration package or a full customer configuration package is being dealt with.

The import date, the import description, the database version, the transport package name, the export criterion as well as the data source name are recorded in the database history of the destination database.

Importing a Transport Package

The program "ActiveEntry Transporter" transfers the transport packages to the database. The program guides you through each step. The <Next> button takes you to the next step in the program. Use the <Back> button to return to the previous step. The <Cancel> button discards all the changes and exits the program.

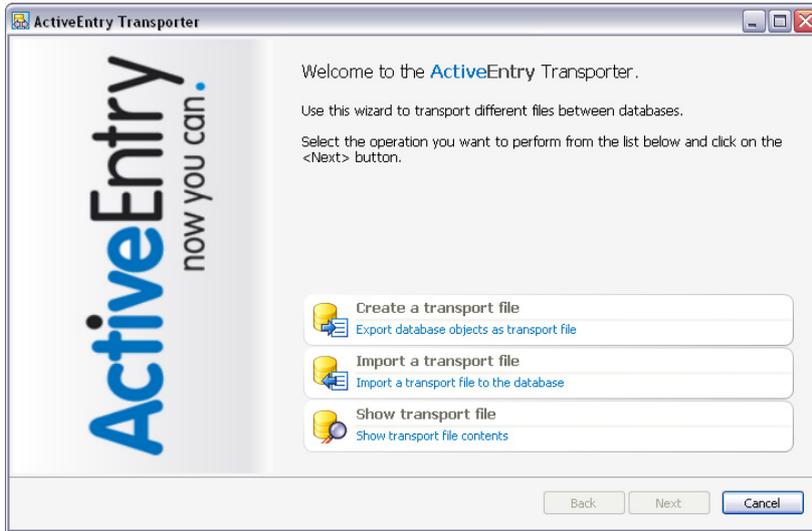


Figure 84: "ActiveEntry Transporter" Startup Screen

After starting the program, select the option <Import transport file> as the next step. In the next dialog window, enter the connection data for the ActiveEntry database you want to import the transport package into. The login is done as described in section [Logging into the Database with a Database User](#). After entering the required connection data continue with <Next>.

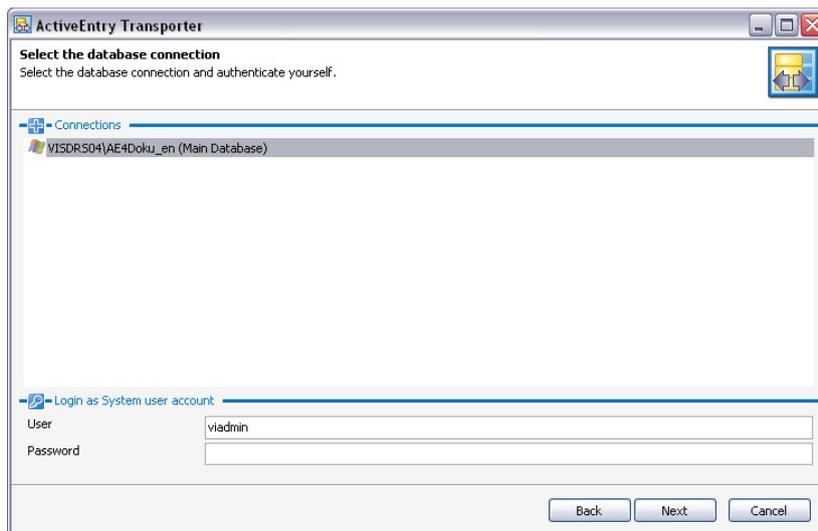


Figure 85: Creating a Database Connection

Select the transport package with the import data. Start the data import with the <Next> button.

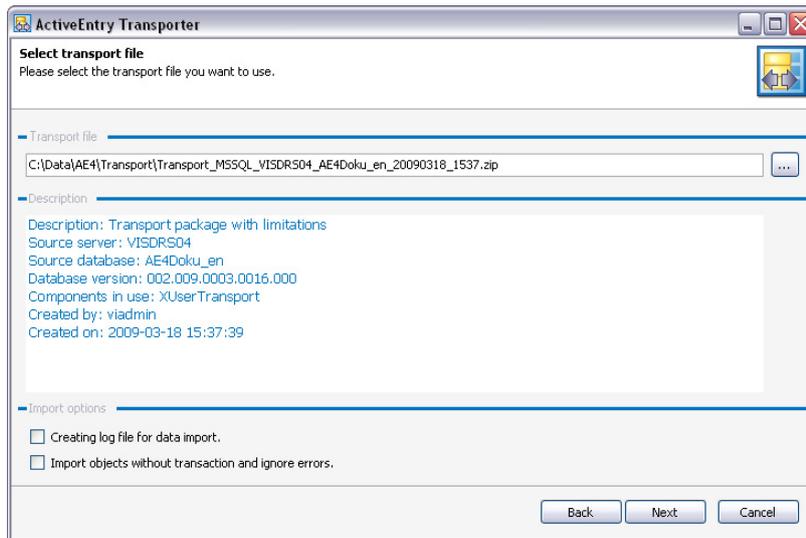


Figure 86: Selecting the Transport Package

The program determines the import steps to be carried out and displays them in the dialog window.

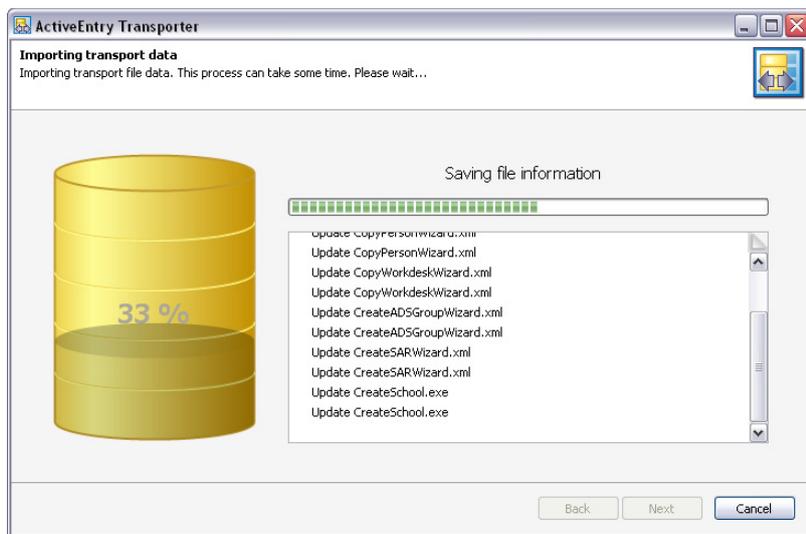


Figure 87: Importing the Data

The final step is to set up computation tasks for the DBScheduler. These are displayed in the next dialog window.

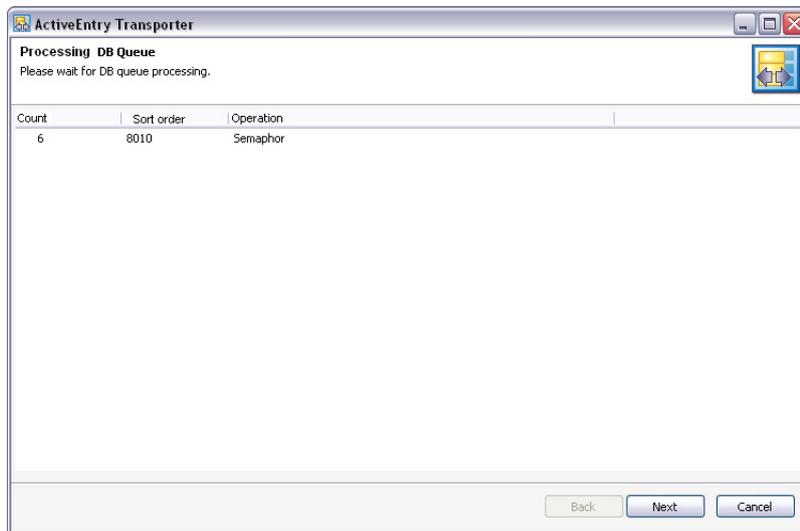


Figure 88: Tasks Set Up for the DBScheduler

If changes have been made to the system configuration, i.e. processes or scripts imported, you have to compile the database after the tasks have been processed. Compilation is started automatically once importing is complete.

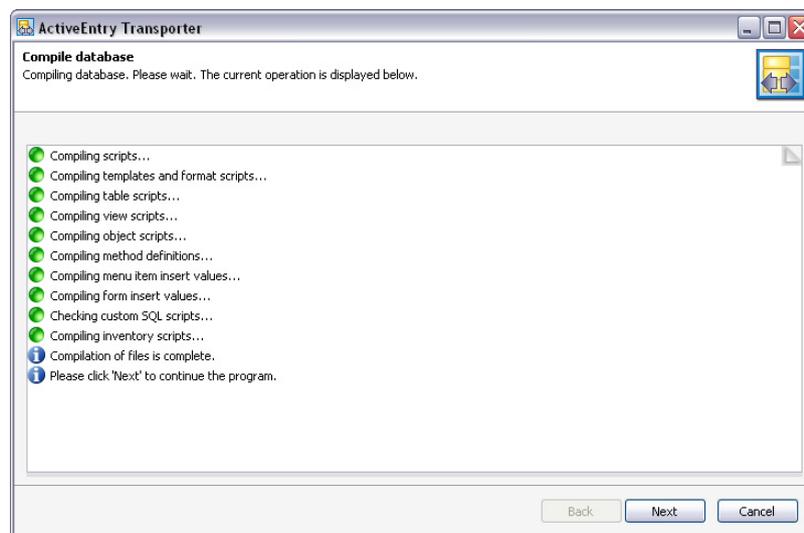


Figure 89: Compiling the Database

Once the import procedure is finished you can close the program with the <Finish> button.

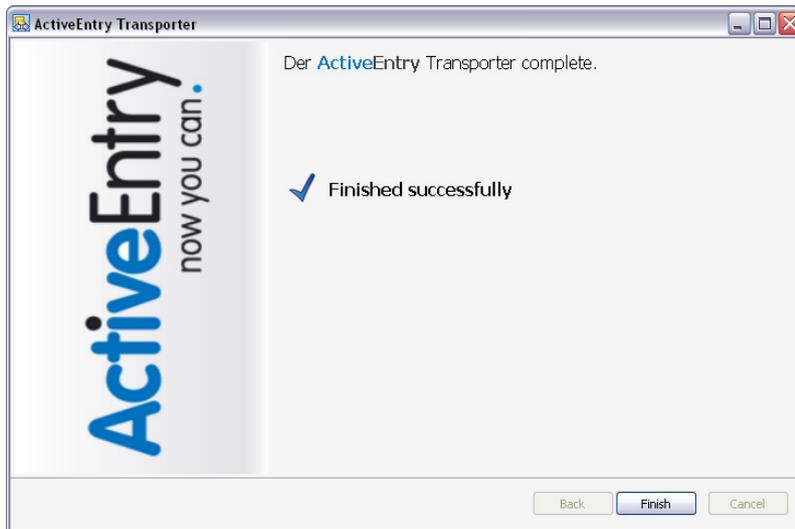


Figure 90: Quitting the Program

4.7.2 Transferring Custom Data

Create a customer configuration package to exchange custom changes between the development database, test database and the live database.

Prerequisites for data transfer are:

- Source and destination database have the same migration version.
- Source and destination database have the same database system base

You can specify restricting export criterion for creating customer configuration packages. System user modifications, modifications as from a defined date, or individual objects can be exported. A limited customer configuration package is recommended for transporting individual changes from a development database to a test database. However, you should create a complete customer configuration package to transfer changes from the test database to the productive database. When a full customer configuration package is imported, new data sets are added to the destination database and existing data sets are updated. Redundant data sets are deleted from the target database.

The export date, the export description, the database version number as well as the export criterion and export file name are recorded in the database history in the source database.

Creating a Customer Configuration Package

Create the customer configuration package with program "ActiveEntry Transporter". This program guides you through step by step. Use the <Next> button to move on to the next step in the program. The <Back> button takes you back to the previous step. The <Cancel> button discards all changes and exits the program.

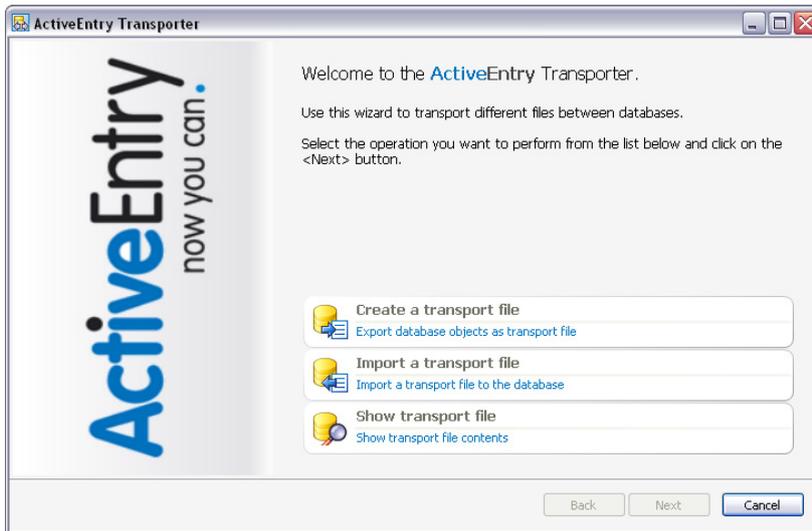


Figure 91: "ActiveEntry Transporter" Startup Screen

After starting the program, select the option <Create transport file> as the next step. Enter the connection data for the ActiveEntry database you want to create the transport package from in the next dialog window. The login is done as described in section [Logging into the Database with a Database User](#). After entering the required connection data continue with <Next>.

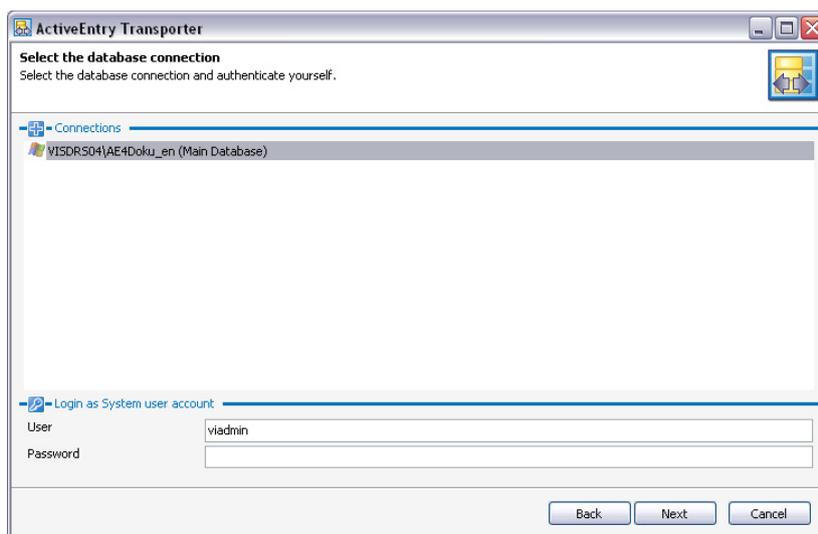


Figure 92: Creating a Database Connection

In the next dialog window, enter the name of the file to export the changes to. Start the export with the <Next> button.

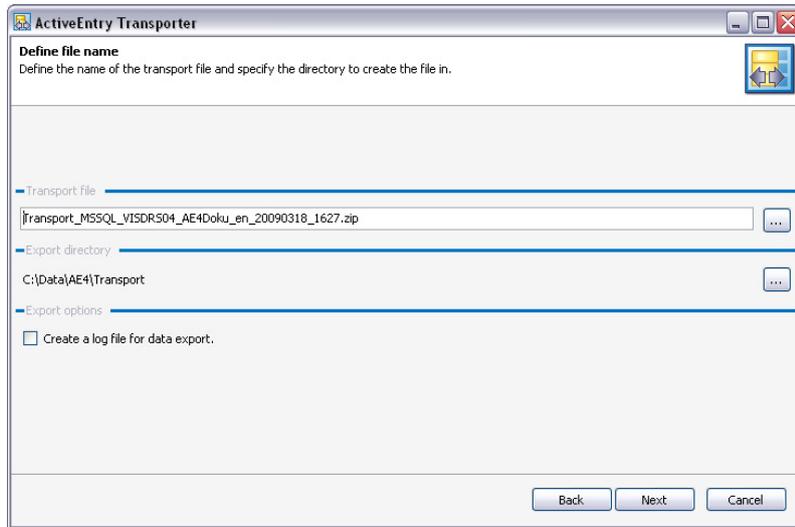


Figure 93: Transport Package Name

The transport parameters are shown in the next dialog window. Enter a description of the transport data here. Then the <Next> button is enabled and you can specify export criteria.

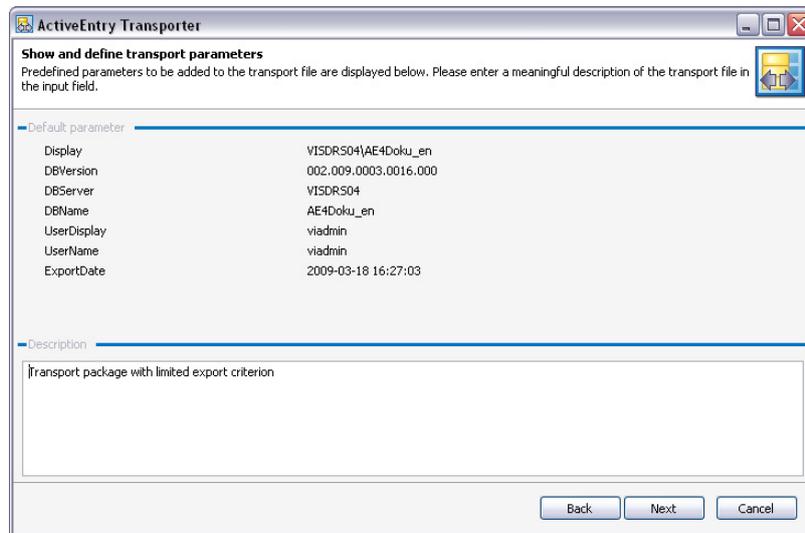


Figure 94: Data Export

Each export criterion is described in more detail in the next section. You may use several export criteria for creating the customer configuration package.

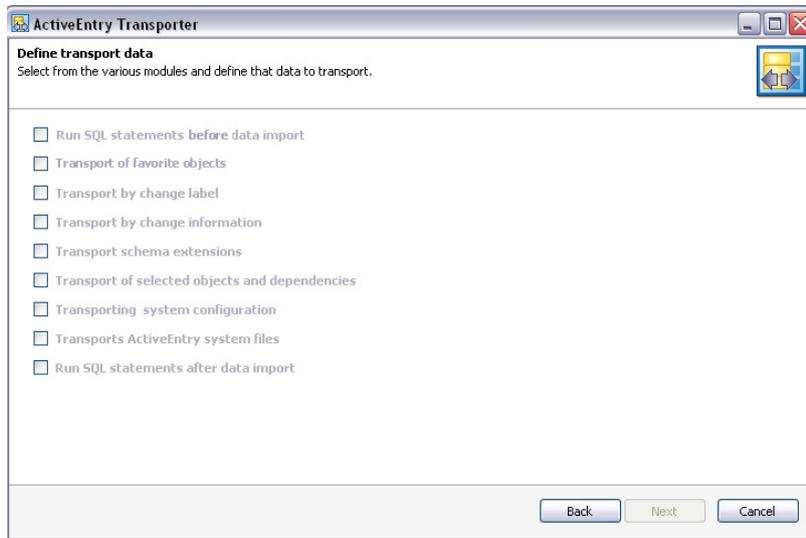


Figure 95: Specifying Export Criteria

Once you have specified the export criteria, the export is started. The program determines the data to export and displays the progress of the export in the dialog window. The export procedure can take some time.

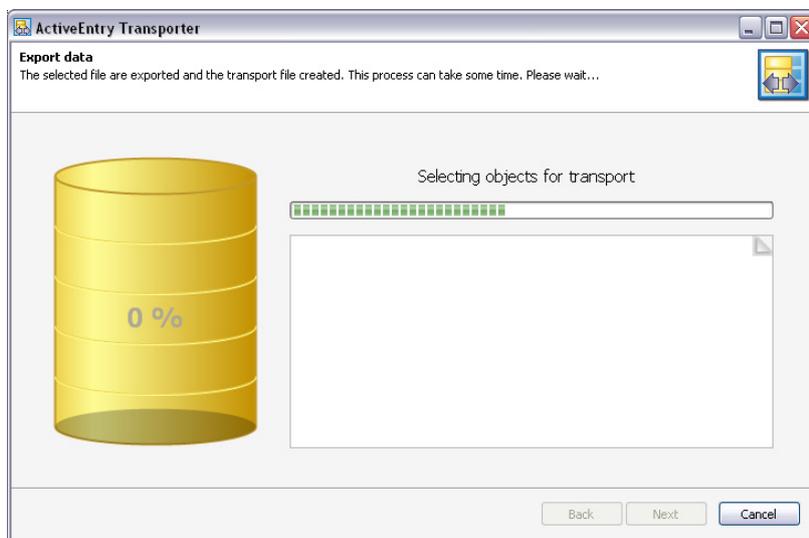


Figure 96: Exporting Transport Data

After the export procedure is finished, you can quit the program with the <Finished> button.

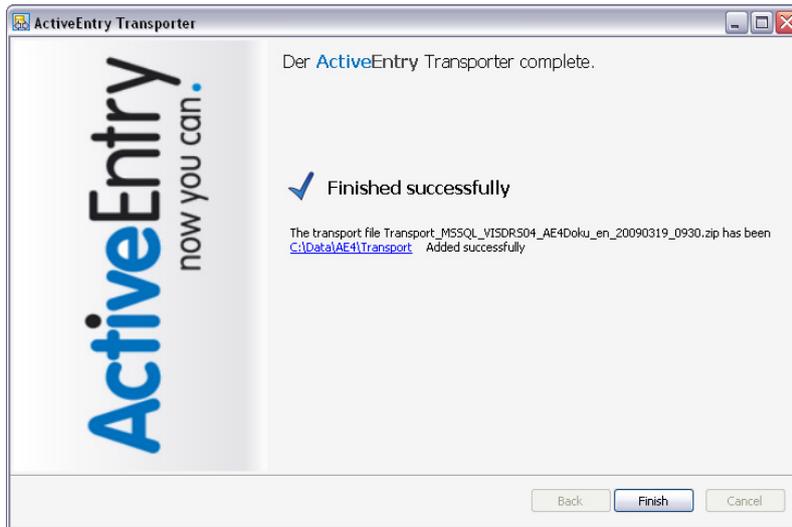


Figure 97: Quitting the Program

Transport by Change Label

Several objects are grouped together under a change label and can be swapped between source database and target database in this way. When a customer configuration package is imported with change labels, new data records are added to the target database and existing data records are updated. In addition, data records that are marked for deletion in the change labels are deleted from the database.

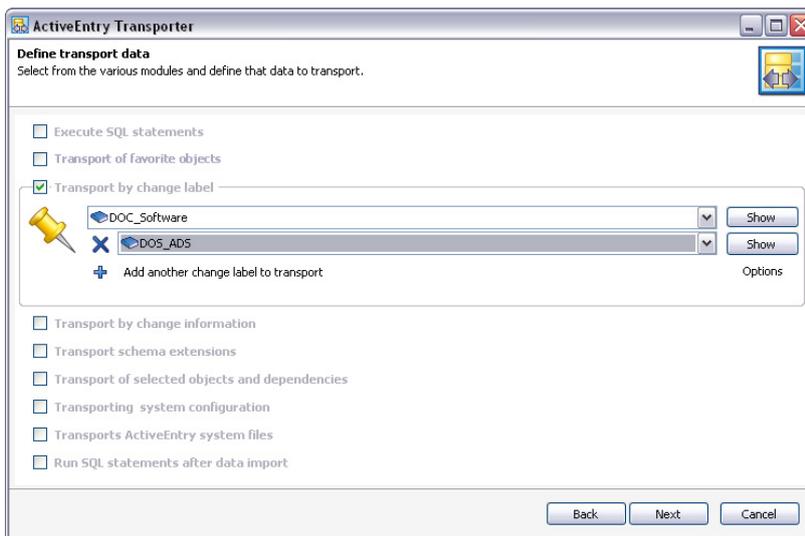


Figure 98: Exporting Change Labels

NOTE!

Refer to section [Working with Change Labels](#) to find out how to create change labels. There are no change labels available after initial migration.

You can group together several change labels in one customer configuration package. Use the <Show> button to show the objects that belong to a change label.

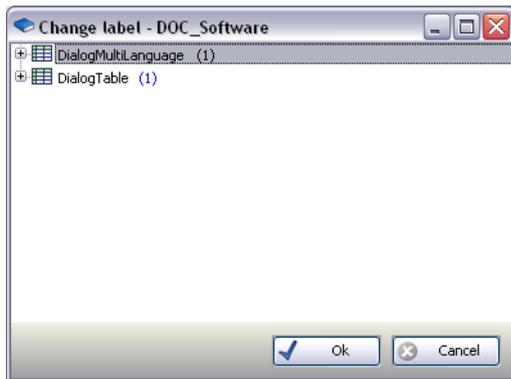


Figure 99: Contents of a Change Label

Use the <Options> button to enter addition change label transport settings. The option <Close change labels after export> ensures that the change label is closed. This means that no more changes can be booked to this change label.

Transport by Change Information

Use transport by change information to limit transportation data by user, time period and database tables.

Use the user selection to specify whether only your changes (entry <me>), changes from all users (entry <all users>) or changes from specified users (entry <selected users>) are added to the customer configuration package. A selection list of system users is displayed from which you

can make a selection in the case of specified users. You can also specify additional users directly in an input field or by using a selection dialog. You can select more than one user with the key combination <shift + select> or <ctl + select>.

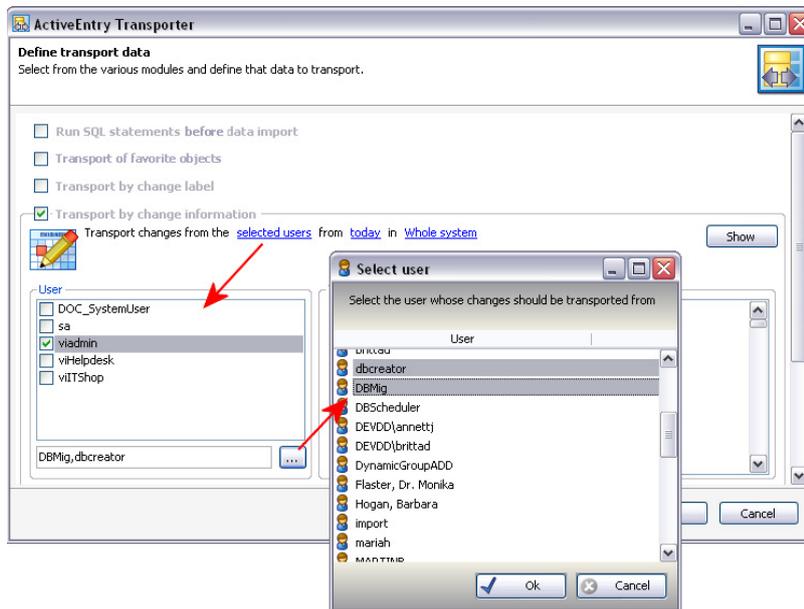


Figure 100: User Selection for Transport by Change Information

Use the date filter to export changes for the selected user(s) from a specified date. There are some predefined date suggestion to speed up date selection. However, you can also choose another time period for the transport.

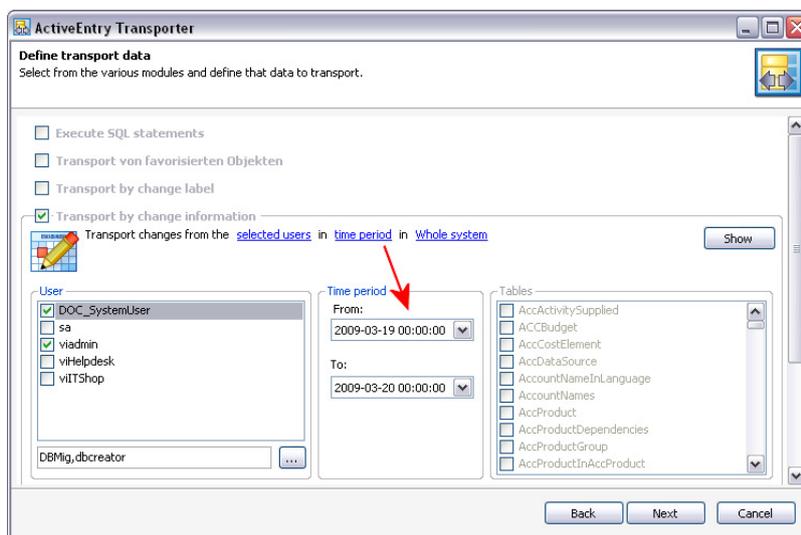


Figure 101: Selecting a Date for Transport by Change Information

You can limit transportation data even further by selecting database tables. Specify whether changes to all tables <entry <whole system>), to system data (entry <system data>), to user data (entry <user data>) or to specified tables (entry <selected tables>) should be added to the configuration package.

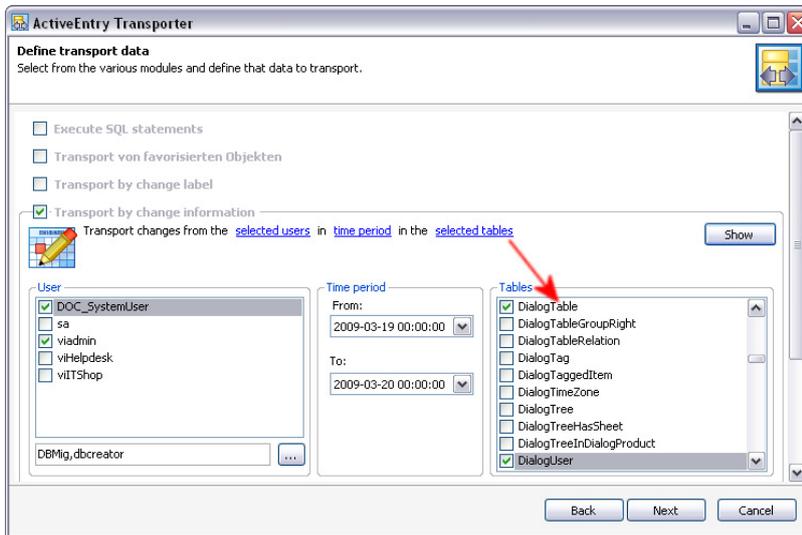


Figure 102: Selecting Tables for Transport by Change Information

Use the <Show> button to show the objects that meet the specified export criteria.



Figure 103: Transport Data Display

Transporting Schema Extensions

Custom database procedures, functions, triggers, views and indexes have to be labeled with a customer specific prefix with a maximum of 5 characters before they can be transferred to the database. This customer prefix must be given for the main database. Furthermore, only custom database procedures, functions, triggers, views and indexes that are not encoded and are smaller than 64 kb are included. Custom database procedures, functions, triggers and views are always exported in their entirety.

Use the <Show> button to show the schema extensions that are effected.

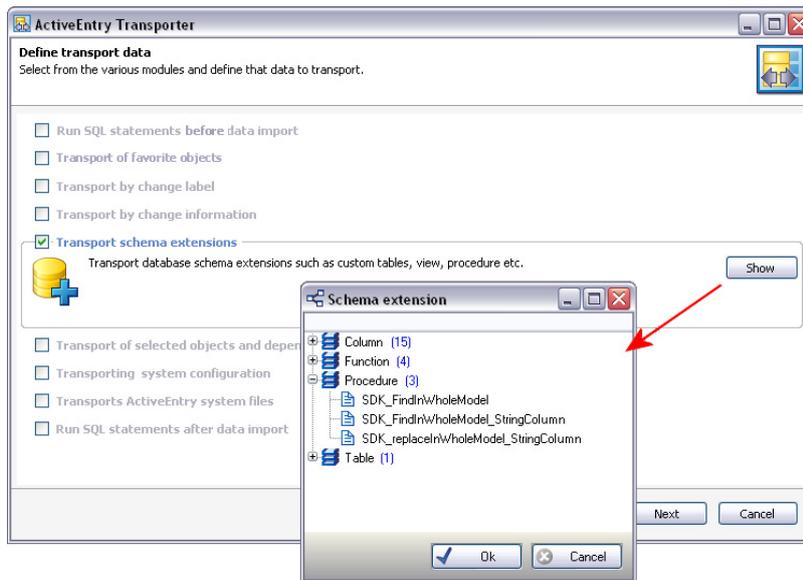


Figure 104: Displaying Schema Extensions

Transporting Selected Objects and their Dependencies

Use these export criteria to add individual objects and their dependencies to the configuration package. Use the <Select> button to open a selection dialog where you can choose each object to be transported.

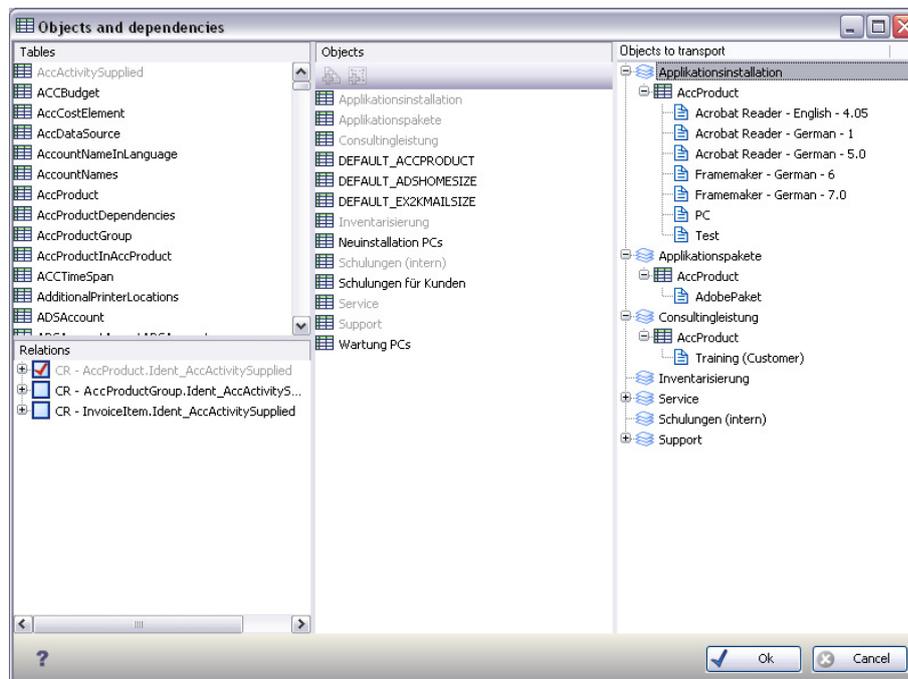


Figure 105: Selecting Single Objects

To choose the objects, select the database table in the <Tables> list that you want to take the objects from that will be added to the customer configuration package. All the objects in the selected table are shown in the <Object> list. Select the object and use the context menu to add it to the transport. You can select more than one object in the list with <shift + select> or <ctl + select>. All selected objects and their dependencies are shown in the <Transport object> selection list.

Table 9: Meaning of Icons for Object Selection

Icon	Meaning
	The selected objects and dependencies are added to the transport package. There is no post-processing of objects after data import.
	The selected objects and dependencies are added to the transport package. Redundant objects are deleted after data import.

Table 10: Entries in the Context Menu for Object Selection

Context Menu Entry	Meaning
Add	The selected objects and dependencies are added to the transport package. There is no post-processing after data import.
Add with post-processing	The selected objects and dependencies are added to the transport package. Redundant objects are deleted after data import.
Remove	Selected objects are deleted from the transport package.

You can add objects that are dependencies of the chosen object directly without having to select each one individually. There is a selection list of <Dependencies> (ChildRelation (CR), Foreign-Key (FK) and M:N relations) displayed for the selected database table. Enable the relations you want. The objects linked via these relations are added to the transport when an object is selected.

System Configuration Transport

Use this export criterion to export the customer specific modifications entirely. You should create a full customer configuration package to transfer the changes from the test database to the productive database in their entirety. When a full customer configuration package is imported, new data sets are added to the destination database and existing data sets are updated. In addition, redundant data sets are deleted from the target database.

System File Transport

Use this export criteria to add new or changed files to the customer configuration package. Use the <Select> button to open a selection dialog that shows all the files. You can select more than one file with <shift + select> or <ctl + select>.

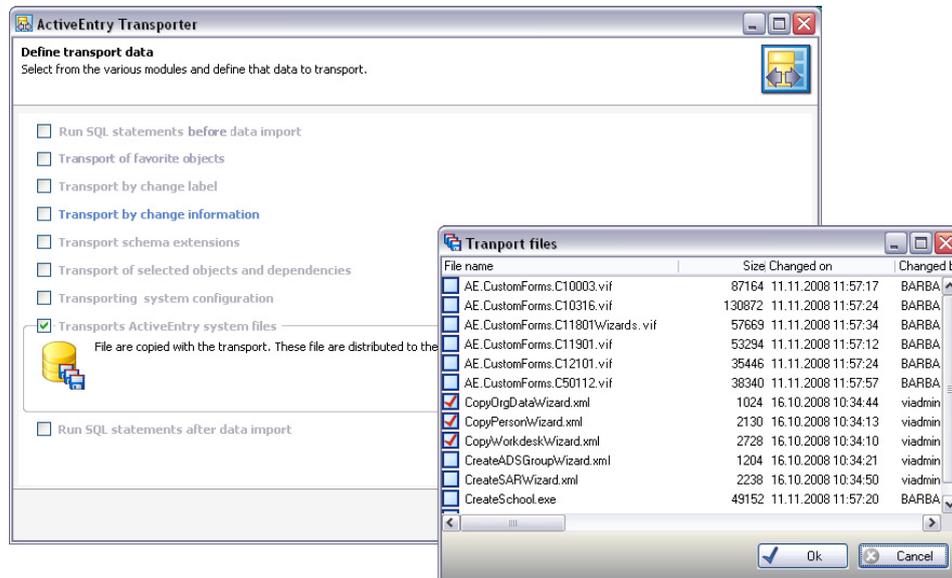


Figure 106: Selecting System Files

4.7.3 Automatic Software Updating

A method has been developed for ActiveEntry to automatically update ActiveEntry tools in order to guarantee an acceptable level of administrative effort.

All files from an ActiveEntry installation are stored in the ActiveEntry database with their names, repository and binary code. Each file is associated to an ActiveEntry tool, such as the ActiveEntry Manager or ActiveEntry Service. In addition, the size and hash values are stored in the database for each file in order to identify them and also the severity level, which specifies how significant the file changes are for the system.

Table 11: Severity Level Meaning

Severity Level	Significance
0	Uncritical modification
1	Non-functional modification
2	Functional modification
3	Critical modification

The necessary files are loaded into the ActiveEntry database and updated when a hotfix, a service pack or a full version update is run.

A semaphore "Softwarerevision" is continually maintained in the database. When a file is added, changed or deleted in the database, the semaphore value is recalculated by the DBScheduler.

In every ActiveEntry installation directory there is a file "Softwarerevision.viv". This file contains the following information:

- The installation revision number
The revision number is determined by the semaphore value 'softwarerevision' in the database.
- The start time of the last modification

Whether a software update is necessary is determined by comparing semaphore values. If the semaphore values are different, each file in the installation directory is checked to see if it is declared in the database.

If the file is in the database, the following is checked:

- Has the file size changed?
If this is the case, the file is added to the list of files to be updated.
- Has the hash value changed?
If this is the case, the file is added to the list of files to be updated.

New files that have been loaded into the ActiveEntry database through a hotfix, a service pack or a full version update with the program "ActiveEntry Transporter" or through a custom file loaded with the Software Loader, are also added to the list. All the files in the list are updated. All actions are logged in the file "update.log". After the update has finished, the current semaphore value is copied from the database to the file "softwarerevision.viv".

Automatic Updating of ActiveEntry Tools

When a program starts up, VI.DB.dll creates a connection to the database and carries out the semaphore test. If the file "softwarerevision.viv" is not found, a new file is added. If the ActiveEntry installation directory does not have write access, an error message is displayed and the software update is continued depending on the configuration parameter "Common\Autoupdate\AllowOutOfTimeApps".

Table 12: Permitted Configuration Parameter Values

Value	Meaning for Software Update
0	Working with VI software that does not correspond to the required revision number is not permitted.
1	Working with VI software that does not correspond to the required revision number is only permitted when a maximum of 'non-functional modifications' are pending.

Table 12: Permitted Configuration Parameter Values

Value	Meaning for Software Update
2	Working with VI software that does not correspond to the required revision number is only permitted when a maximum of 'functional modifications' are pending.
3	Working with VI software that does not correspond to the required revision number is also permitted for 'critical modifications'.

If a file update is necessary, it is initially tested to see if the configuration parameter "Common\Autoupdate" is enabled. If it is disabled, a warning is shown and execution is continued without updating. If the configuration parameter is enabled, the update is done.

To prevent further applications from starting during the update, a file called "Update.lock" is created in the installation directory. The trigger program and the update program (updater.exe) write their process ID's in the file. The lock-file is deleted from the installation directory once updating has been successfully completed. The program is then restarted. To ensure that automatic updating is restarted when an application is restarted after quitting unexpectedly, lock-files older than two hours are ignored. If none of the processes whose ID's are saved in the lock-file exist on the workstation when the application is restarted, the lock-file is also ignored and the update is restarted.

The semaphore test is carried out by VI.DB.dll on a cyclical basis during normal operations. If a file is identified for update, the update process is started automatically. The system user that is logged in, has the possibility to execute the update immediately. For this, a dialog window is displayed in the user interface of the administration tool. If the system user does not choose this option, he can continue working depending on the significance of the update. The update is started the next time the program is started.

Automatic Job Server Updating

ActiveEntry Service returns the actual state of the semaphore "SoftwareRevision" after each request following a process step. If this value differs from the value in the database, the Job server is labeled with "updating" in the database and no more normal process steps are sent to it. A software update process is generated instead.

This process initially determines the start time of the last change from the file "SoftwareRevision.viv". A list is compiled of all files with additional information specifying whether each file is new or not. This list is evaluated on the Job server to be updated and another list is compiled

specifying which files will be updated. A process which updates the files is generated as the last process step. ActiveEntry Service is restarted if any one of the files has changed on the Job server. After the update is completed, the Job server label is reset in the database.

NOTE!

Use the Job Server Editor in ActiveEntry Designer to label a Job server with the option <No automatic update> if it should be excluded from updating. You have to manually update servers that are marked like this.

Automatic software updating was first implemented in the version "AE.NET_2006-05-17" of ActiveEntry. When changing from an older version of ActiveEntry to this or a later version, you must run a one-off update of ActiveEntry Service on the Job server with the program "ActiveEntry Service Updater" or manually (see section [Updating Individual Job Servers](#)). Only then can you use automatic software updating to update the Job server.

4.7.4 Updating Individual Job Servers

Use the program "ActiveEntry Service Updater" to update individual Job servers. The program can be used if you do not want to automatically update Job servers. This program is also used to prepare a Job server for automatic software update when changing from an older version of ActiveEntry to version "AE.NET_2006-05-17" or later. When changing to this version you must run a one-off update of ActiveEntry Service on the Job server with the program "ActiveEntry Service Updater" or manually. Only then can you use automatic software updating to update the Job server.

Login as described in the section [Logging into ActiveEntry Tools](#). After entering the necessary connection data, confirm with <OK>.



Figure 107: Login

The following actions are available for updating:

- **Restart**
An update task is placed in the Job queue that restarts ActiveEntry Service on the selected Job server.
- **Update**
A task is placed in the Job queue that updates ActiveEntry Service on the selected Job server. The necessary files are sent to the Job server. ActiveEntry Service is stopped, updated and restarted. A log file "Autoupdate.log" is created in the service's installation directory.
- **Senden**
A request is placed in the Job queue to send files to the selected Job server. The files are stored in the directory "Update" in the service's installation directory. ActiveEntry Service is not automatically updated.

Enter the directory with the installation sources under <Directory with installation sources> and select the Job server to update in the list <Server to update>.

ActiveEntry Service can also be updated over a network share or an HTTP server. For this, you have to select the option <Copy files from URL> under <URL from update directory> and enter the path as URL for the directories with the installation sources. The files to send to the Job server are not passed as process step parameters in this case, only the URL is sent to ActiveEntry Service which then takes the files from there.

You can specify the time at which to start the generated processes by inputting the time in <Start time> and with it the time for updating.

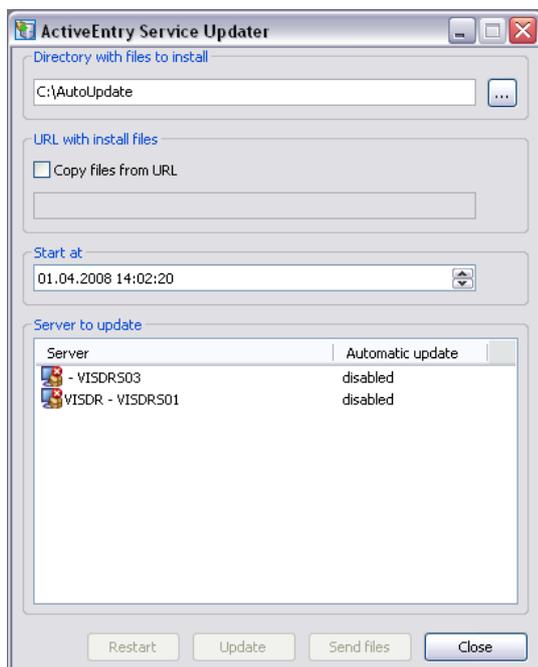


Figure 108: Updating the Job Server with ActiveEntry Service Updater

4.7.5 Importing New Files into the ActiveEntry Database

In order to distribute new or changed custom files, such as custom form archives, through automatic software updating, the files are loaded into the ActiveEntry database with the program Software Loader“.

The program guides you through each individual step. Use <Next > to move on to the next step in the program. The <Back> button takes you back to the previous step. Use <Cancel> to discard any changes and quit the program.



Figure 109: "Software Loader" Startup Screen

After the program has started, enter permitted connection data for the ActiveEntry database. Use the <Select> button to reach the connection dialog window.

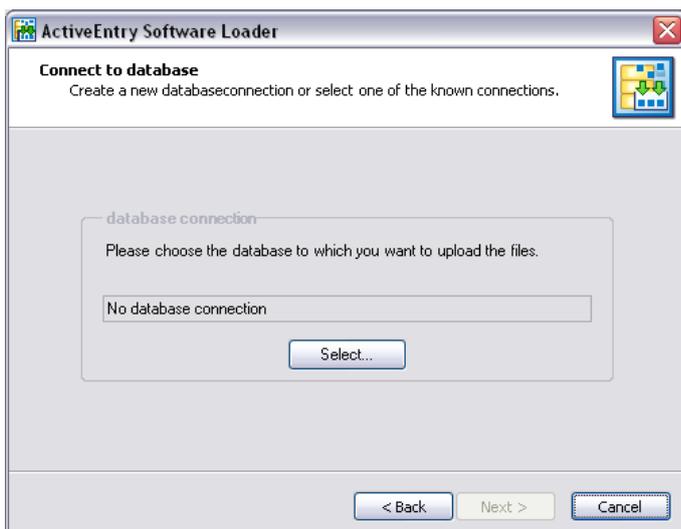


Figure 110: Connecting to the Database

Login as described in the section [Logging into ActiveEntry Tools](#). After entering the necessary connection data, confirm with <Next>.



Figure 111: Login

Select the option to import new files into the ActiveEntry database.

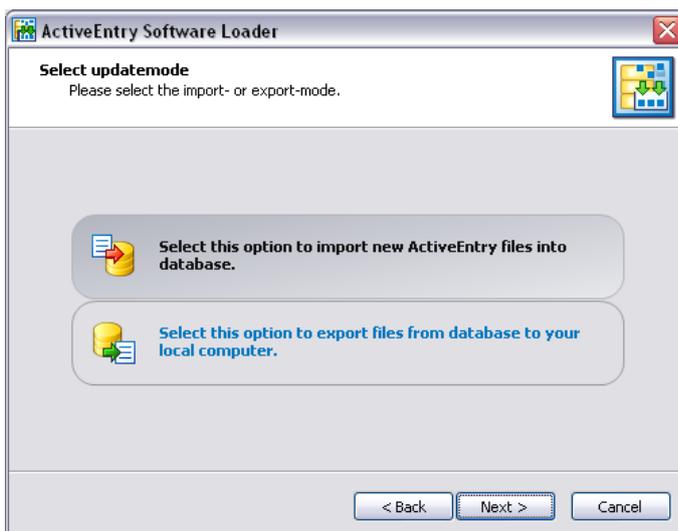


Figure 112: Selecting the Transfer Direction

Use the next step to decide the significance of the file update on system behavior.

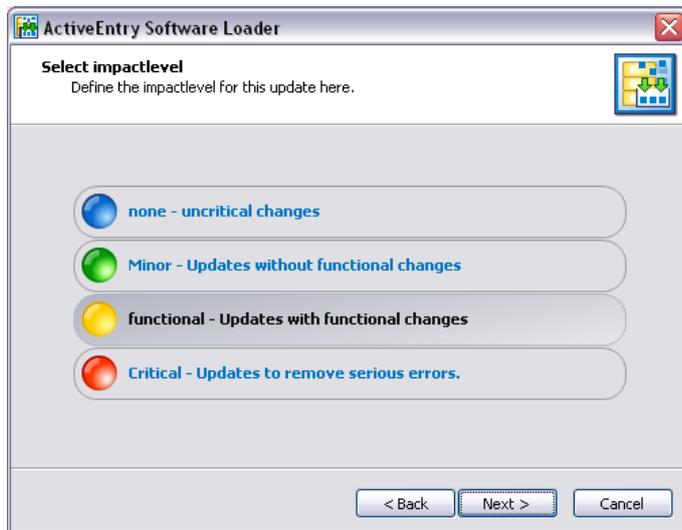


Figure 113: Specifying the Severity Level for New Files

Issue a change label to mark files in order to simplify the transfer of new files between various databases (test database, development database, operational database). Change labels are displayed as a export criterion when a customer transport package is created in "ActiveEntry Transporter"

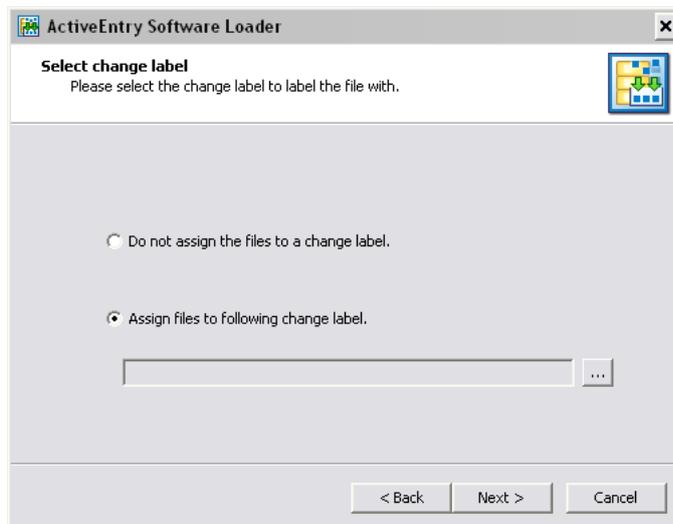


Figure 114: Change Labels for New Files.

If you want to assign a change label use the button next to the appropriate option to open a dialog window which shows all known change labels. Here you can accept an existing change label or create a new one. You can also edit the change labels at this point. Read the section [Working with Change Labels](#) in the Configuration Manual for more information.

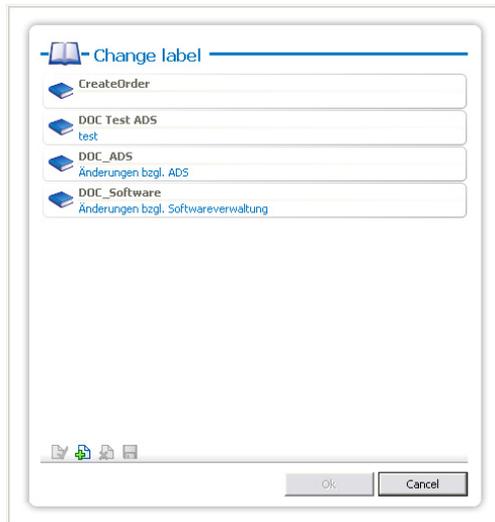


Figure 115: Selecting a Change Label

In the next step, specify the base directory where the new files will be kept. The status and file size of all the files in the selected directory are displayed in the file list. The status is determined from the file information in the database. To test the file version, the file size and the hash value are determined and compared to the entry in the database.

Table 13: Meaning of Status

Status	Meaning
Unknown version	The file is known but is not loaded in the database. There is no version information in the database.
Unknown file	The file is new. The file is in the list of known files but has not been loaded in the database yet. There is not version information in the database.
Version OK	The file version matches the version in the database.
Version changed	The file version has changed with respect to the one in the database.

Select the column you want to sort by, by clicking with the mouse in the column title bar. Mark the files to be loaded into the ActiveEntry database. You can select several files at one time (<shift or ctrl> + select).

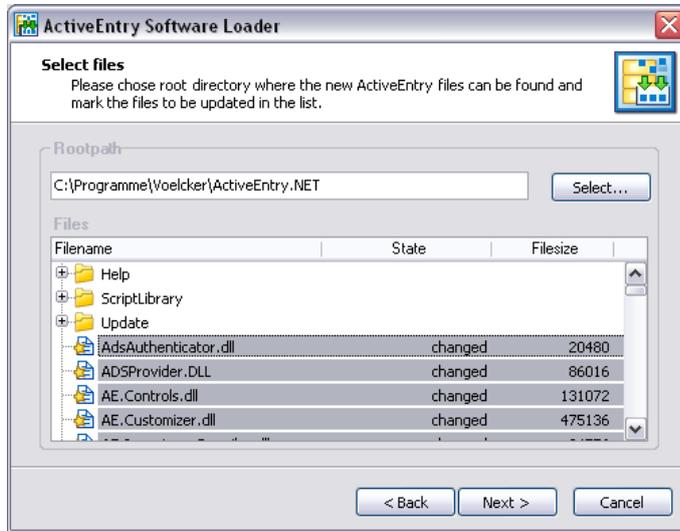


Figure 116: Selecting Which Files to Load

Specify the application group that the file belongs to before the loading so that not every file in every installation folder is copied. In addition, specify whether a backup should be made of the existing files during automatic software update before the files are exchanged.



Figure 117: Question Loading New Files

The following application groups are available for selection:

- Service
The file is required by ActiveEntry Service.
- GUI
The file is required by ActiveEntry administration tools.
- Web
The file is required by ActiveEntry web applications.

Confirm the setting for each file with the <OK> button if you want to handle them separately. If certain files should not be loaded into the database, select the button <Ignore>. Use the <All> button to load all of the files with the same setting into the database.

In the next step, the marked files are loaded into the database. Loading may take some time. The loaded files are displayed in the dialog window.

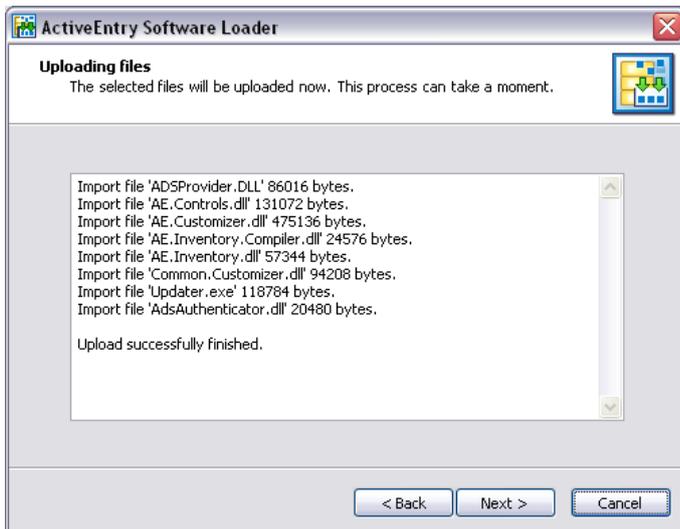


Figure 118: Loading the Files into the Database

After successfully loading the files into the database, the semaphore value "Softwarerevision" is updated in the database by the DBScheduler. In this way, the files to be updated are added to the update file list at the next semaphore test and distributed to the workstations and the Job servers. The computation tasks for the DBScheduler are displayed in the next dialog window.

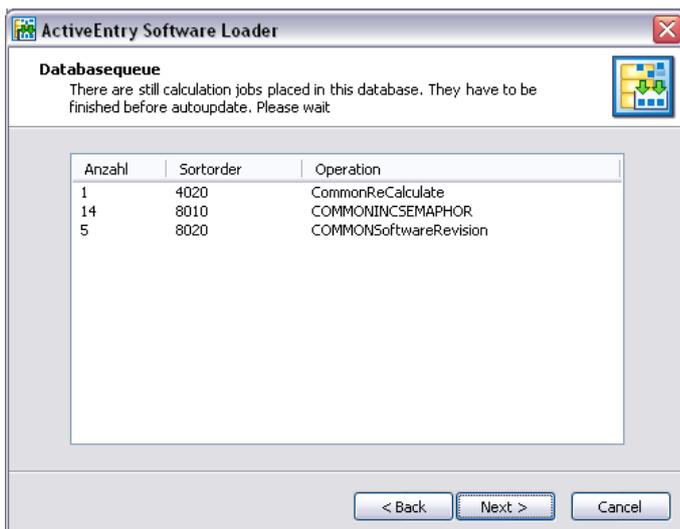


Figure 119: DBScheduler Computational Tasks

Use the <Next> button to move to the next dialog window where you can quit the program with the <Finish> button.

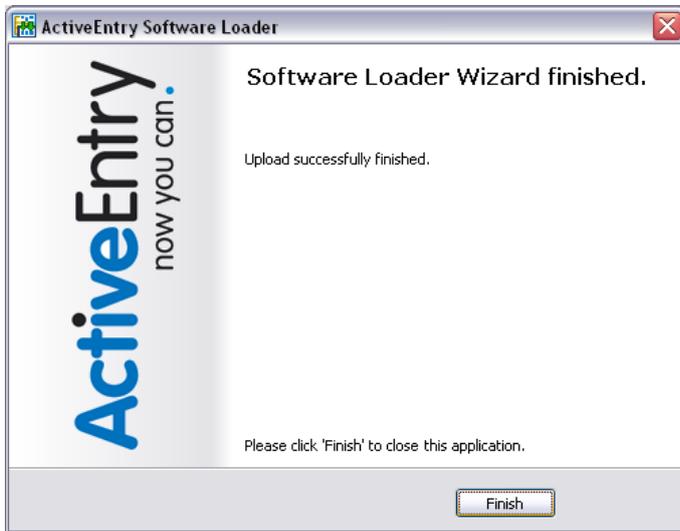


Figure 120: Program Exit

4.7.6 Exporting Files from the ActiveEntry Database

In order to equip individual Job servers with the newest software version, you have to export the files from the ActiveEntry database. The program "Software Loader" exports the files from the database.

The program takes you through each step. Use the <Next> button to move to the next step in the program. The <Back> button takes you back to the previous step. Use the <Cancel> button to discard any changes and exit the program.



Figure 121: "Software Loader" Startup Screen

After the program has started, enter valid connection data for the ActiveEntry database. Use the <Select> button to open the connection dialog window.

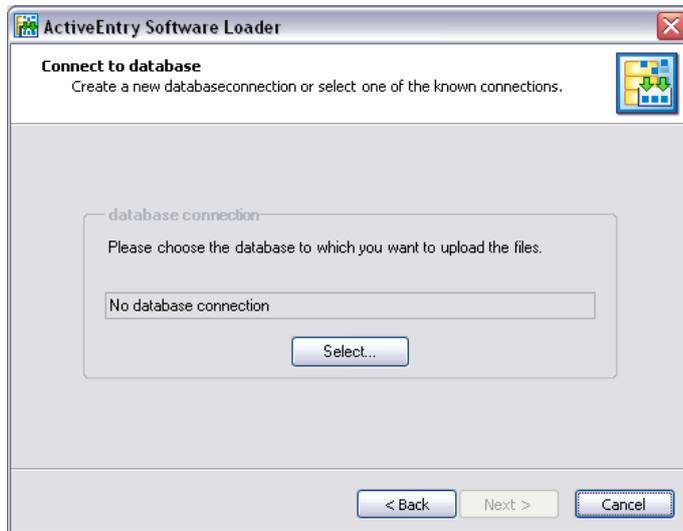


Figure 122: Connecting to the Database

Login as described in the section [Logging into ActiveEntry Tools](#). After entering the necessary connection data, confirm with <Next>.



Figure 123: Login

Select the option to export files from the ActiveEntry database.

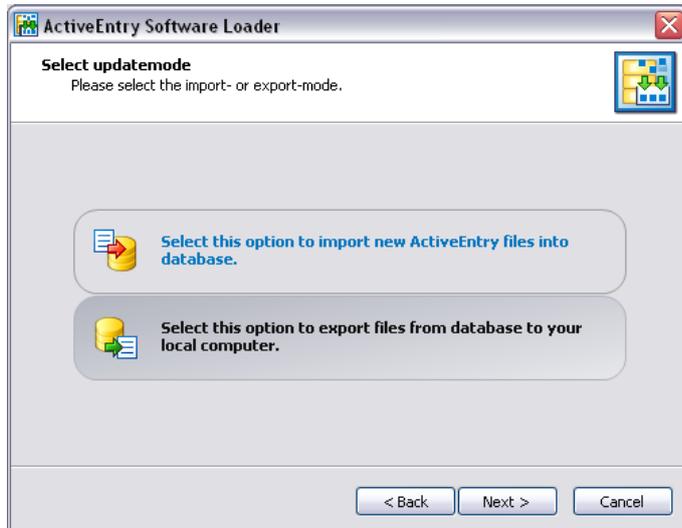


Figure 124: Selecting the Transfer Direction

Select the application group for the file export in the next dialog window. The program loads the file information from the ActiveEntry database depending on which application group is selected.



Figure 125: Selecting the Application Group

The following application groups are available:

- Service
All the files are loaded that ActiveEntry Service needs.
- GUI
All the files are loaded that are needed by ActiveEntry administration tools.
- WEB
All the files are loaded that are required by ActiveEntry web applications.

Specify the directory for storing the files.

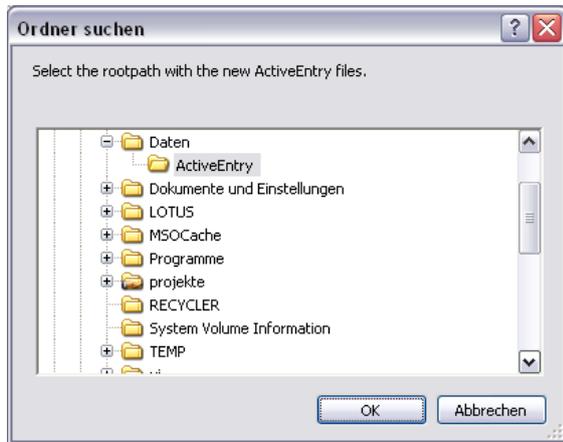


Figure 126: Selecting the Directory

Exportable files are displayed in the next dialog window with their size and status. The program checks whether ActiveEntry files already exist in the given directory in order to determine the status. If this is the case, the files are updated, otherwise the files are recreated.

Table 14: Meaning of Status

Status	Meaning
Unknown file	The file has not been exported from the database into the given directory yet.
Version OK	The file version matches the version in the database.
Version changed	The file version has changed with respect to the one in the database.

Select the column you want to sort by, by clicking with the mouse in the column title bar. Mark the files to be loaded into the ActiveEntry database. You can select several files at one time (<shift or ctrl> + select).

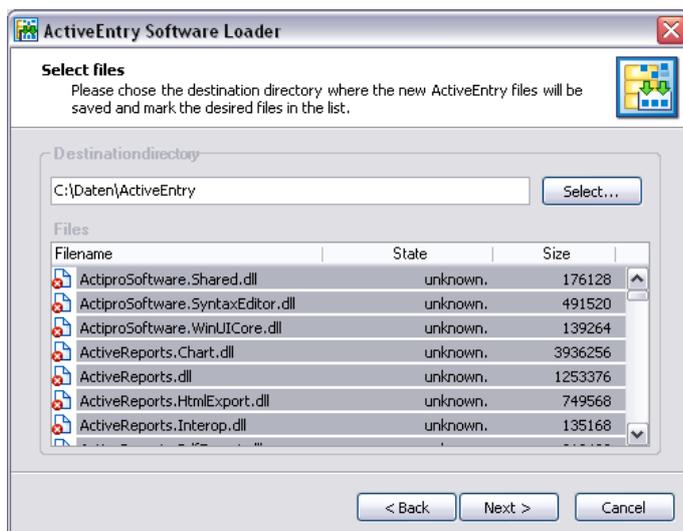


Figure 127: File Selection

The files that are marked in the given directory are exported in the next step. This may take some time depending on the number of files selected. Any export errors are output to the dialog window.

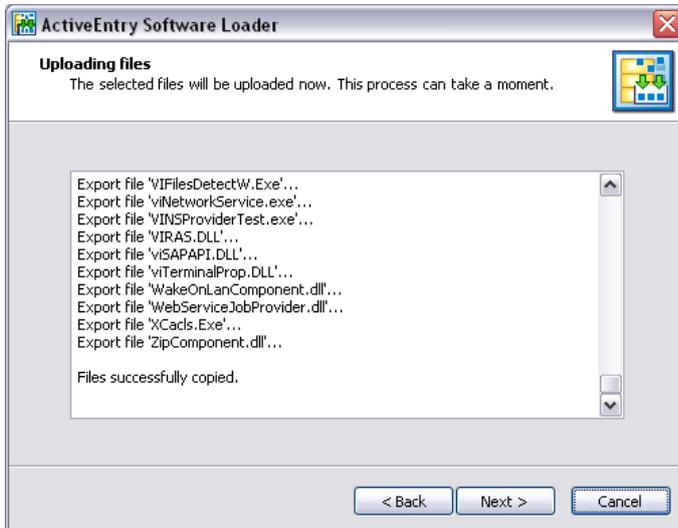


Figure 128: Exporting the Files

Use the <Next> button to move to the next dialog window where you can exit the program with the <Finish> button.

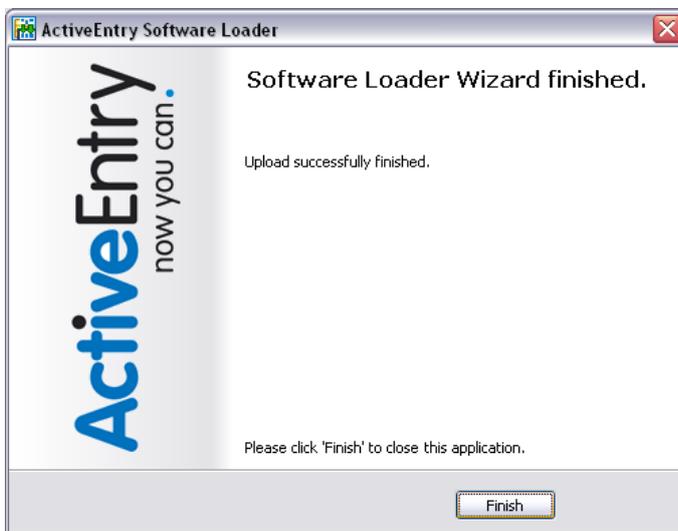


Figure 129: Program Exit

4.8 Compiling the ActiveEntry Database

After importing a migration package or a full customer configuration package, the programs "ActiveEntry Transporter" and "DBMigrator" cause the database to be compiled immediately. The database also has to be compiled after importing hot fixes or limited customer configuration pa-

ackages such as changes to processes, scripts, templates, object definitions, method definitions and preprocessor relevant configuration parameters. For this you need the program DBCompiler“

The DBCompiler guides you through one step at a time. Use <Next> to move to the next step. The button <Back>, returns you to the previous step. With <Cancel> all the changes are ignored and the program ends.



Figure 130: Startup screen for the DBCompilers

Enter the connection parameters for the ActiveEntry database by selecting the connection with the button <Select>.

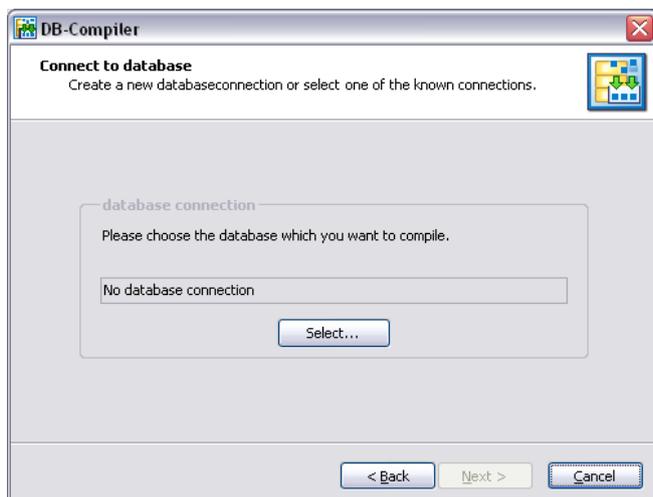


Figure 131: Creating the Database Connection

The login is the same as described in the section [Logging into ActiveEntry Tools](#). You can use any authentication module to log into the DBCompiler. After entering the connection data confirm the step with <Next>.



Figure 132: Login

Connection parameters for the ActiveEntry database must be entered for the initial compilation. The following data requires checking and if necessary should be changed for an ActiveEntry database under Microsoft SQL:

- **ConnectionString**
`User ID=<database user>;initial Catalog=<database>;
 Data Source=<server>;Password=<database user password>;
 pooling='false'`

If a known instance of the database server is used as data source, the notation of the input string is:

```
User ID=<database user>;initial Catalog=<database>;
Data Source=<server>\<instance>;Password=<database user password>;
pooling='false'
```

- **ConnectionProvider**
`VI.DB.ViSqlFactory,VI.DB`

An ActiveEntry schema under Oracle needs to be checked and, if necessary, changed in the same way:

- **ConnectionString**
`Data Source=<data source from TNSNames.ora>;User ID=<database user>;
 Password=<database user password>`
- **ConnectionProvider**
`VI.DB.Oracle.ViOracleFactory, VI.DB.Oracle`

Use the button next to the input field to change the "ConnectionString". Select the connection data for your database. You can find more detailed information about connection data in the section [Logging into the Database with a Database User](#). Furthermore, enter the full customer name, customer prefix and the customer number.

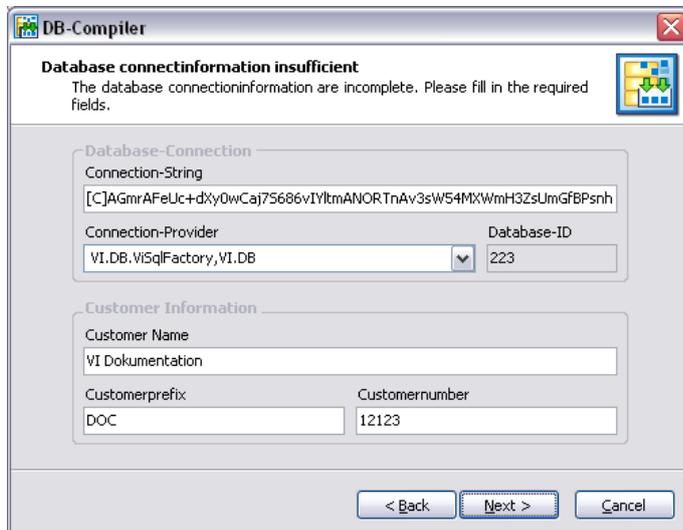


Figure 133: Completing the Database Connection Information

Before you begin the compilation, all the DBScheduler tasks have to be processed. If there are still outstanding tasks on the database, you are notified by the DBCompiler. In this case the compilation can not go ahead.

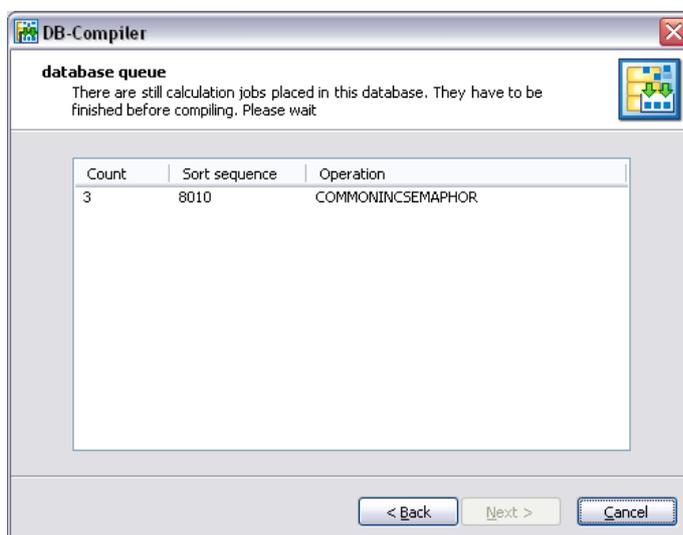


Figure 134: Displaying Outstanding DBScheduler Tasks before Starting the Compilation

Enter which parts of the compiler need to be recompiled. The following components can be compiled:

- Scripts from the script library

- Script expressions such as templates, format scripts and method can be used for example.
- Processes
- Conversion scripts for hardware inventory

Custom SQL procedures can also be tested.

First specify how the library script should be compiled. The selection of other components to be compiled depends on this. Here you can select:

- Do not compile scripts
- Script without dependencies
This method means that changes to scripts do not become active until ActiveEntry is restarted.
- Scripts including all dependences
This recompiles the scripts and all dependencies (templates, methods, processes). This guarantees that the script changes are loaded and become effective immediately. ActiveEntry tools do not need to be restarted.

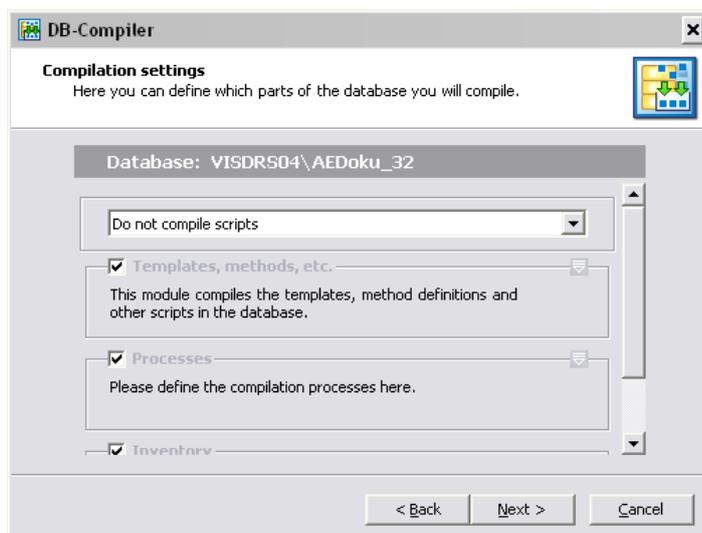


Figure 135: Specifying which Database Parts to Compile

For script compiling, you can further specify which code snippets should be compiled, e.g. templates, format scripts, selection scripts, method definitions or input values.

- Templates and formatting scripts
- Selection scripts for tables, views and objects
- Method definitions

- Insert values

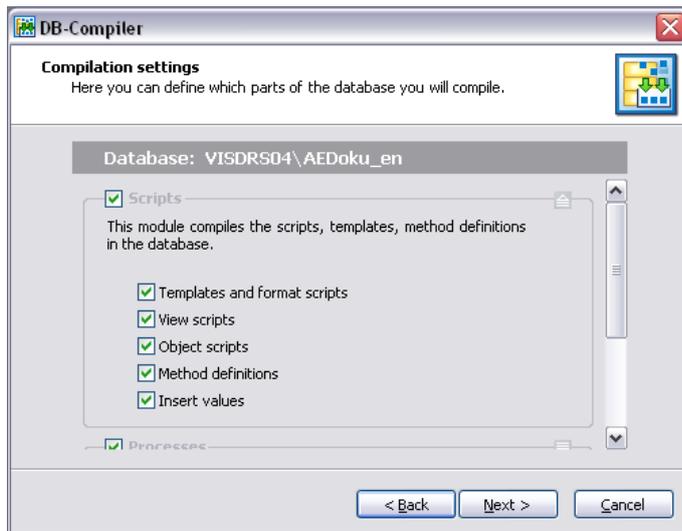


Figure 136: Selecting Which Scripts to Compile

Enable the option <Processes> when you want to compile them. Use the other options to specify which processes should be compiled. These are the available options:

- All processes
- Changed processes
Compiles processes that have been changed since the last compilation.
- Selected processes
Here you can select individual objects whose processes should be compiled.

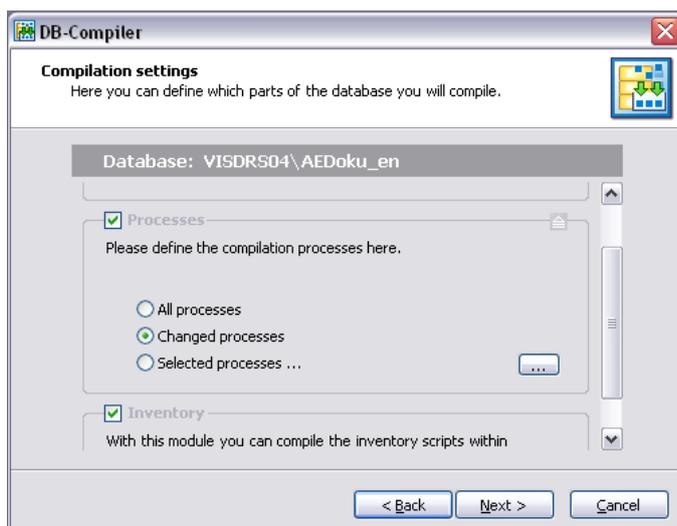


Figure 137: Selection of processes to Compile

Use the button next to the option <Selected processes...> to open the dialog. Here, you can choose between changed processes, all processes or only selected processes (user defined). You can limit the preselection even more. After confirming the process selection, use the <OK> button to return to the previous dialog window.

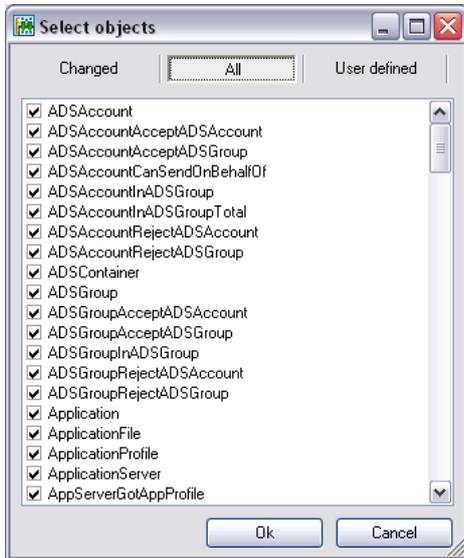


Figure 138: Selecting processes to be Compiled

Enable the option <Verify SQL procedures> to check the syntax of the custom SQL scripts.

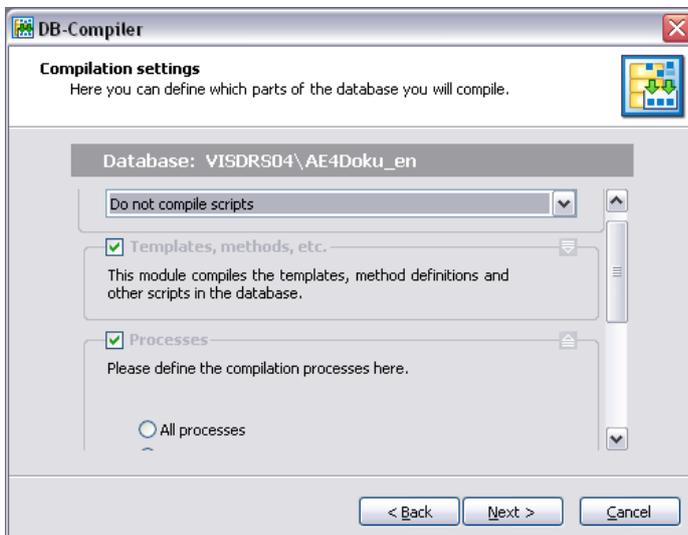


Figure 139: Option for Testing SQL Procedures

Enable the option <Inventory> if you want to compile the hardware inventory conversion scripts. This option is only available when the hardware inventory components are installed and released.

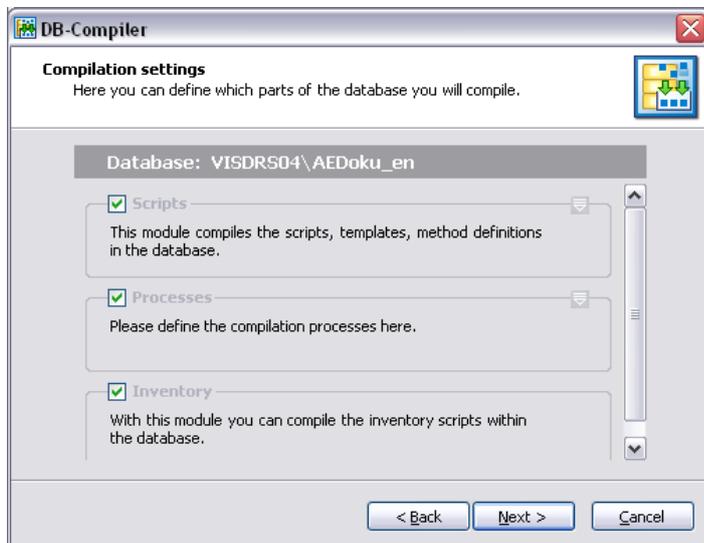


Figure 140: Selecting the Conversion Scripts

The compilation can start after confirming the selection of the database parts to be compiled with <Next>. Compiling the database can take some time. The object that is currently being processed is shown in a dialog window.

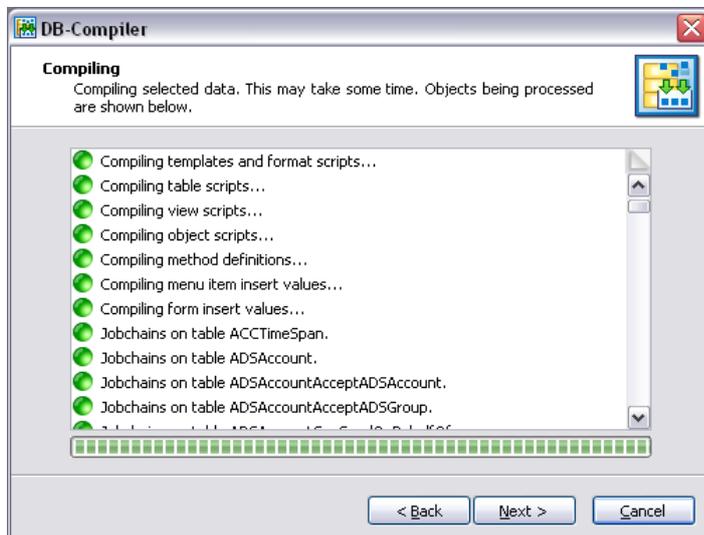


Figure 141: Compiling

Possible compilation errors are outputted to the log window. If you double click on the error message with the mouse you jump to the corresponding line in the source code view (upper part of the window). The code can be edited here.

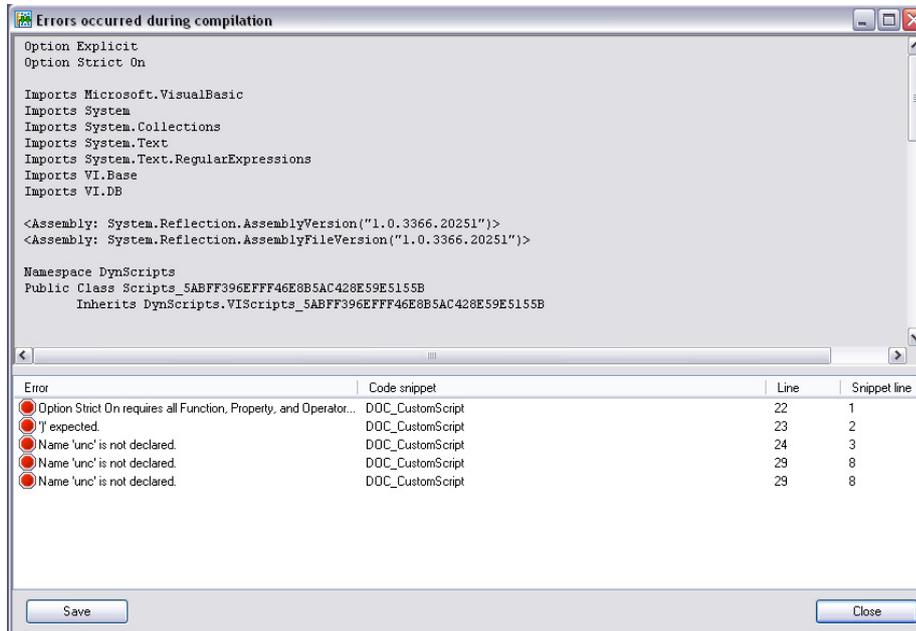


Figure 142: Error Message Output

Use the <Save> button to save error messages in a file. Once the window is closed (button <Close>), the compilation is continued. You can correct any errors after compiling. Note that the changes have to be recompiled.

If warnings occur during compilation they are shown at the end.

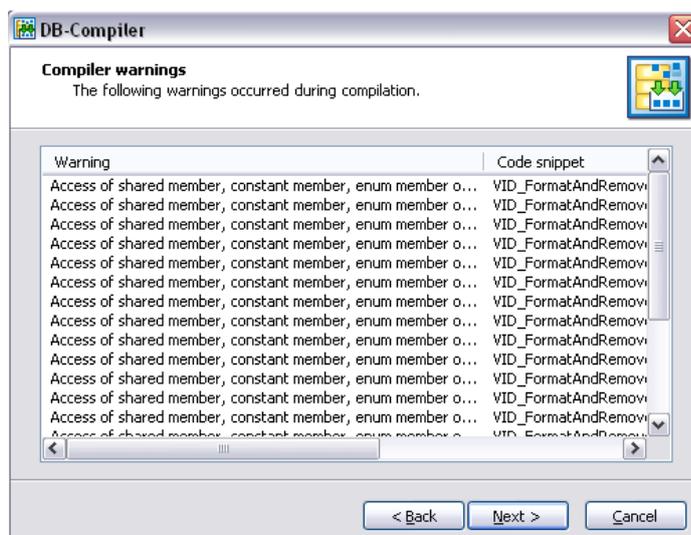


Figure 143: Warning Message Output

After the compilation is successfully completed, you can quit the DBCompiler with the button <Quit>. You can subsequently use ActiveEntry tools to access the ActiveEntry database.

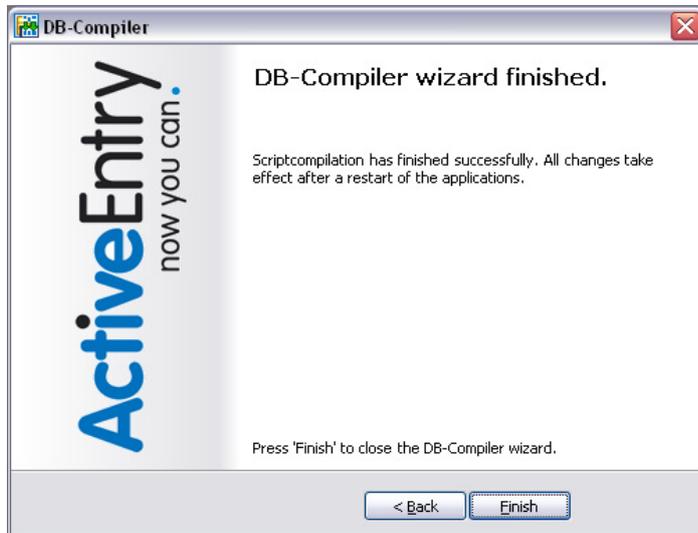


Figure 144: Quitting the DBCompiler

4.9 Database Information Encryption

In certain circumstances it may be necessary to store encrypted information in the database. Encryption is carried out by the program "Crypto Configuration". With this program an encryption file is created and the contents of the database columns that are affected are converted.

The encrypted information is stored in the database table "Dialogdatabase". The program takes you through step-by-step. Using the button <Next> you reach the next step. The button <Back>, returns you to the previous step. With <Cancel> all the changes are ignored and the program ends.



Figure 145: Startup screen for the Program "Crypto Configuration"

Create a connection with the ActiveEntry database by entering the connection data over the button <Connection>.



Figure 146: Creating the Database Connection

The login takes place as described in section [Logging into ActiveEntry Tools](#). After entering the required data, confirm with <Next>.



Figure 147: Login

You can specify, which actions should be carried out. You can chose from:

- Creating or changing a database key.
- Encryption using an existing key.
The encryption information already has to be entered in the database table "DialogData-base".

- Data decryption.
An encryption file has to exist that can be used for decoding.

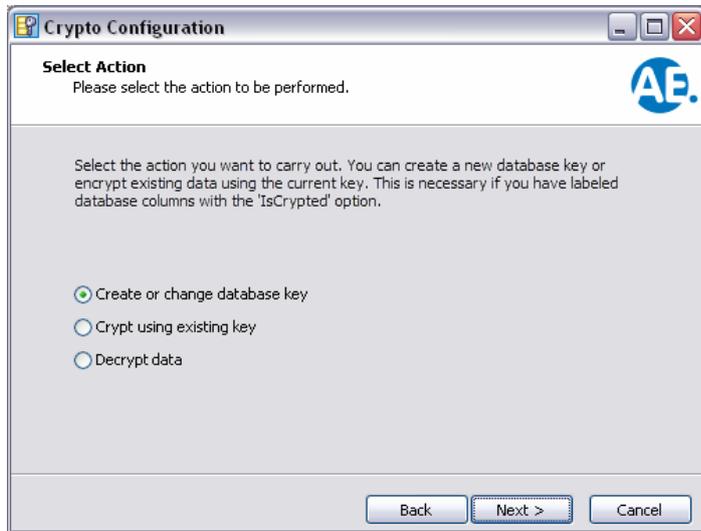


Figure 148: Creating a Database Key

In the next dialog window you confirm if an encryption is activated or not. If no encryption was active until now, then a new key file is created in the following steps.

If an encryption is already active, you need to select the file which contains the key. The key can then be changed and saved in a new file.



Figure 149: Creating a New Private Key (1)

Use the <Create key> button to generate a key.



Figure 150: Generating a New Private Key (2)

After you have entered a storage the path for the key, it appears in a dialog window and the button <Next> is activated.



Figure 151: Creating a New Private Key (3)

In the next dialog window you can see the database tables with columns that are going to be converted. All the columns are shown that have property "encrypted (IsCrypted)" set. Start the conversion of the database entries with the button <Convert>.

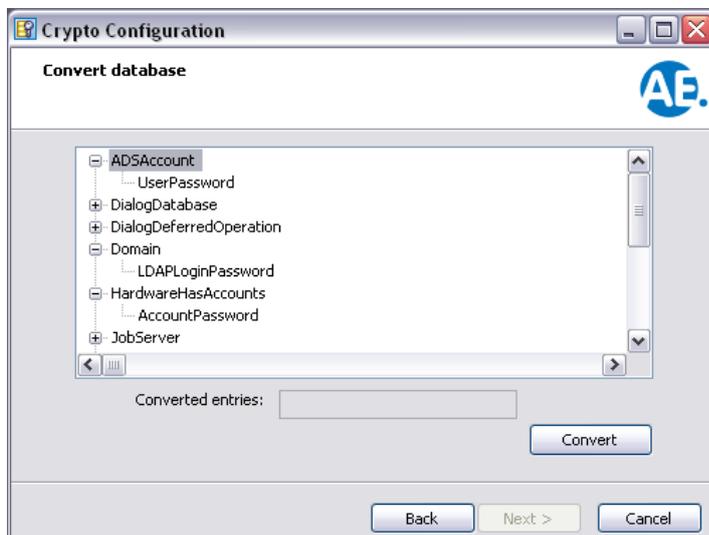


Figure 152: Database Tables and Columns with the property "Encrypted"

After confirming the following two security alerts with <Yes> the conversion begins.

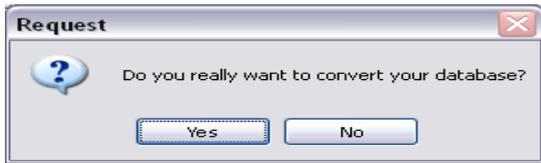


Figure 153: Database Conversion Prompt

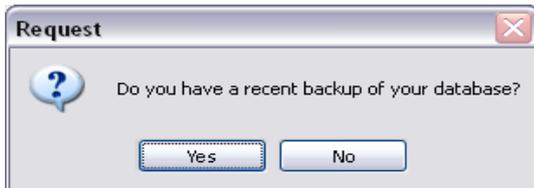


Figure 154: Backup Prompt

The progress of the conversion is displayed.

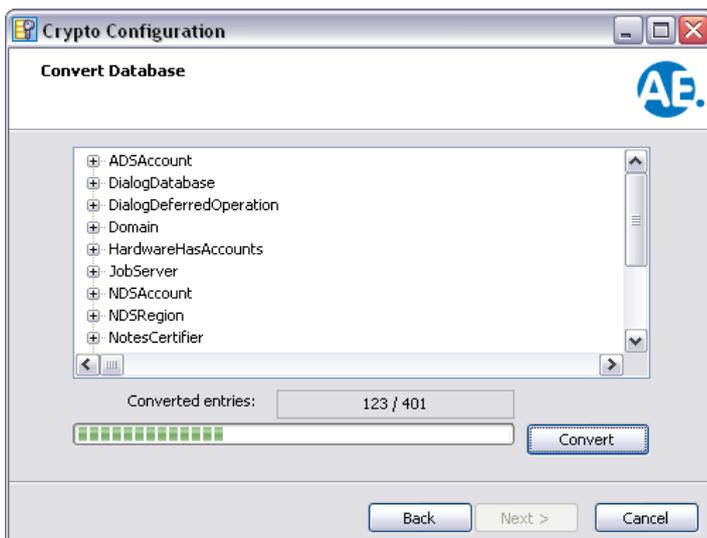


Figure 155: Conversion Progress Bar

After the data conversion is successfully completed, quit the program with the <Quit> button.



Figure 156: Quitting the Program

NOTE!

If the encryption affects data that ActiveEntry Service needs to access, then the key file must be put into the service’s installation directory on all the servers with an active ActiveEntry Service. When ActiveEntry Service finds a private key at the beginning, it stores it in a user related key container and deletes the file from the hard disk. If the ActiveEntry Service user account has changed, you need to copy the key file to the service’s install directory again. The ActiveEntry Service configuration file needs to be have the corresponding Job destination entry added to it. Read more in the section [Configuring ActiveEntry Service](#).

4.10 Setting Up an ActiveEntry Database for Full-Text Search

Table 15: Configuration Parameters for Full-text Search

Configuration Parameter	Effect
Common\Fulltext	This configuration parameter specifies whether the database should be indexed for full-text search. A full-text catalog is created if the parameter is enabled and at least on column is marked for full-text search.

The following prerequisites are required in order to use a full text search in the ActiveEntry administration and configuration tools:

- Installation and start up of Microsoft Full-Text Engine for SQL Server on the database server
- At least one database column is marked for full-text search
- Configuration parameter Common\Fulltext must be enabled

Refer to the Microsoft SQL Server documentation for information for installing and starting the service, Microsoft Full-Text Engine for SQL Server.

4.10.1 Labeling Database Columns for Full-Text Search

WARNING!

- Building or updating the full-text catalog may require a lot of time depending on the data volume in the columns marked for full-text search!
- Processes that are running are stopped and may produce errors.
- Only mark those columns for full-text search that really come into question based on your requirements.

To create a full-text catalog, you need to label the columns in the ActiveEntry database with full-text search. You can use the Schema Editor in ActiveEntry Designer to apply the full-text search label. Read section [Mapping Column Definitions](#) in the Configuration Manual for more information.

Alternatively, ActiveEntry provides you with a procedure that marks all columns for full-text search that have the data type "char", "nchar", "nvarchar", "text" or "ntext", that have more than 15 characters and do not contain UIDs or object keys. If you use this procedure, please note the warning above!

You will find the procedure "SDK_SetAllFulltextColumns" on the ActiveEntry Setup CD in the directory "SDK\SQLSamples\MSSQL2K". You can add the procedure with an appropriate query tool to the database and then run it from there.

Call:

```
exec SDK_SetAllFulltextColumns
```

If the full-text search service has not been started, the procedure has no effect.

NOTE!

Columns that are disabled by preprocessor conditions, are not taken into account when the full-text catalog is created. If columns are enabled at a later point in time due to preprocessor conditions, you have to label them manually for full-text search in the Schema Editor or run the procedure "SDK_SetAllFulltextColumns" again.

4.10.2 Setting Up a Full-Text Catalog

As soon as the configuration parameter "Common\Fulltext" is enabled, a task is generated for the DBScheduler. A full-text catalog is created as a result of processing the task. If the prerequisites are met, the full-text catalog is created.

The full-text catalog's name comprises of:

```
Full-text + <database name>
```

ActiveEntry creates a full-text index for all database columns that are labeled for full-text search. Now you can use full-text search in all the ActiveEntry tools. If you change the full-text search label on a database column, the full-text index is update.

If you disable the configuration parameter "Common\Fulltext", the full-text index is deleted. This means that the full-text search is no longer available. Enable the configuration parameter again to recreate the full-text index.

The DBScheduler checks the prerequisites for building a full-text catalog during an ActiveEntry database migration. If the prerequisites are met, the full-text index is recreated.

NOTE!

If the error "[810143] database error: Cannot alter or drop column <...> because it is enabled for Full-Text Search" occurs during an ActiveEntry database migration, disable the configuration parameter and restart the migration. Once the migration has completed successfully, re-enable the configuration parameter. This recreates full-text index again.

4.11 Setting up a Reference Database

In order to create a reference database, for example, a development or test database, from a backup on another system, the following steps are necessary:

- Create a new database on the database server in the reference environment
- Create a database backup of the original database and copy the backup into the reference database.
- Restore permissions to the database user

After copying the database backup, the database schedules have to be set up. The necessary database schedules are explained in the section [Setting Up Database Schedules](#). You can copy the database schedules into the reference database using a suitable query tool.

Creating the database schedules under Microsoft SQL:

```
exec vid_CheckDefaultSchedules
exec vi_CheckDefaultSchedules
```

Creating the database schedules under Oracle:

```
call vid_CheckDefaultSchedules ()
call vi_CheckDefaultSchedules ()
```

Furthermore, the following changes need to be made to the reference database:

- Licensing the reference database
See section [ActiveEntry Licensing](#) for a description of creating a license request and installing the license.
- Create a new database ID in the reference database
- Change the connection data for the database
- Compile the reference database

There is a mechanism implemented that verifies the database ID during the database compilation and changes it if necessary. This database ID verification takes place during database migration with "ActiveEntry Transporter" as well as during a database compilation with the program "DBCompiler". You have the possibility to check and edit the connection data to the database in both programs.

In the next section the process is explained in an example using the program "DBCompiler".

The database ID is verified after starting the program and logging on to the reference database. If, during the verification, it transpires that the database ID is incorrect, you will be requested to create a new ID. Confirm this request with <Yes>.

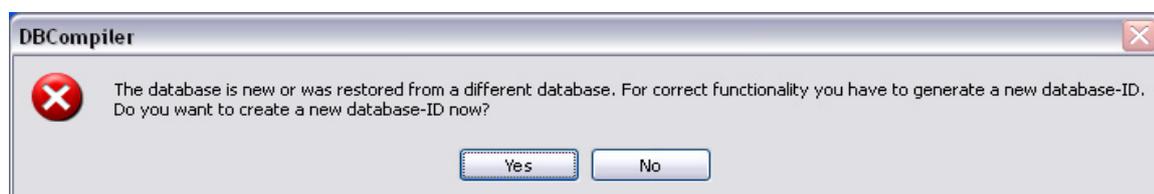


Figure 157: Database ID Verification

As a result, a dialog appears to complete the connection data. Check the data and change it if necessary.

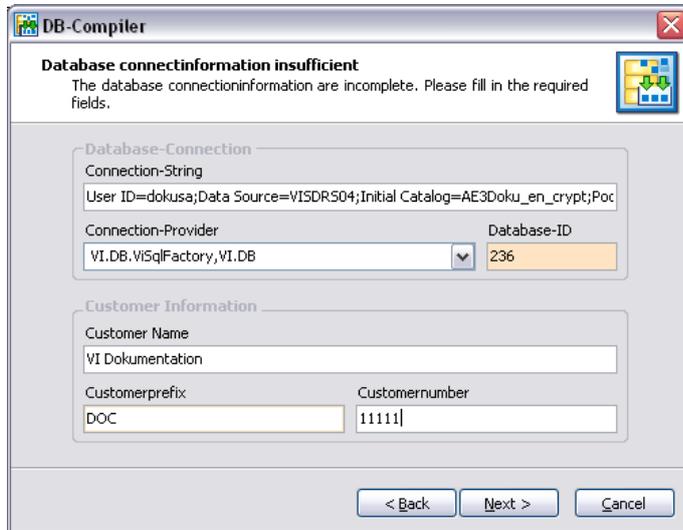


Figure 158: Complete Database Connection Informationen

For an ActiveEntry database under Microsoft SQL, the following data need to be verified and changed if necessary:

- **ConnectionString**
 User ID=<database user>;initial Catalog=<database>;
 Data Source=<server>;Password=<database user password>;
 pooling='false'

If a known instance of the database server is used as data source, the notation of the input string is:

```
User ID=<database user>;initial Catalog=<database>;
Data Source=<server>\<instance>;Password=<database user password>;
pooling='false'
```

- **ConnectionProvider**
 VI.DB.ViSqlFactory,VI.DB

In an ActiveEntry schema under Oracle, the following data needs to be verified and changed is necessary:

- **ConnectionString**
 Data Source=<data source from TNSNames.ora>;User ID=<database user>;
 Password=<database user password>
- **ConnectionProvider**
 VI.DB.Oracle.ViOracleFactory, VI.DB.Oracle

In addition, verify the customer name, customer prefiy and customer number.

If the database is encrypted, you will be explicitly requested to reenter the connection parameter "ConnectionString".

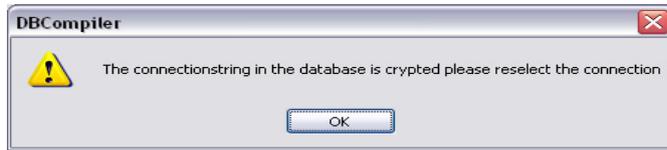


Figure 159: Encrypted Database Alert

In this case, an input mask appears for a database login. Enter your connection data and confirm with <OK>.

For a database connection on a Microsoft SQL Server, enter the server and database name as well as the database user's name and password.



Figure 160: Input Mask for Connection Data under Microsoft SQL Server

If a connection to an Oracle Schema is in use, enter the TNS alias name from the TNSNames.ora into the input mask entry <Data source>. In addition, enter the database user's name and password.

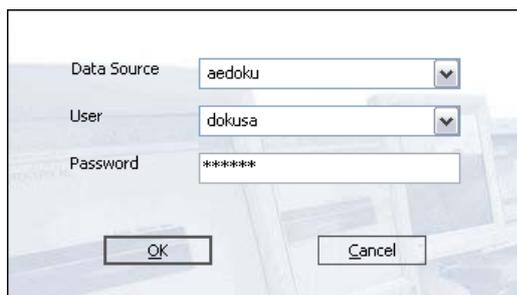


Figure 161: Input Mask with Oracle Connection Data

The connection parameter "ConnectionString" is put together from this data and transferred into the dialog in order to complete the connection data.

After confirming the necessary connection data with <Next>, the database is tested. If the test is successful a new login needs to be done.



Figure 162: Request to Login Again to the Database

After this, the database is compiled, as described in section [Compiling the ActiveEntry Database](#). All parts of the database need to be recompiled. Make sure that all the code snippets, all processes as well as the hardware inventory conversion scripts are marked.

Part II Basics

This section of Getting Started describes the ActiveEntry installation.

The chapters provide an overview of ActiveEntry basic administration and configuration tools functionality. Furthermore, the structure of the user interface is explained and also how to deal with the main ActiveEntry administration tool, the ActiveEntry Manager

Chapter 5 ActiveEntry Tools

ActiveEntry provides you with several configuration tools for managing your networks, controlling on going processes and for configuring ActiveEntry. Installation of the configuration tools is carried out on an administration workstation.

The following sections provide you with an overview of each of the tools. You can find more detailed information in the manuals found using the links next to each tool.

5.1 ActiveEntry Identity Manager

The ActiveEntry Identity Manager is the main administration tool for setting up information about employees and their identities. It displays and maintains all the data required for the administration of employees with their user accounts, permissions and company specific roles in an ActiveEntry network. Company resources that employees need for their work can be entered and assigned to them. Furthermore, the base data for data synchronization between the ActiveEntry database and connected target systems are managed with the ActiveEntry Identity Manager. Different synchronization profiles can be set up for this.

You can also use the ActiveEntry Identity Manager to:

- Define company specific IT policies
- Set up an IT Shop for request company resources and assignments
- Set up special approval processes for authorizing requests and checking compliance to IT policy
- Set up attestation procedures for regularly testing the correctness of data about employees or roles and their assignments

By implementing ActiveEntry application roles, every ActiveEntry user obtains only those access permissions he or she requires to fulfill necessary administrative duties. ActiveEntry Identity Manager is the main administration tool for all ActiveEntry users that belong to an <Administrator> application role. Use the ActiveEntry IT Shop in addition, to maintain personal data, request company resources, authorize and attest assignments or to test rule violations.

Start the program from the start menu "ActiveEntry\Identity Manager" or via "IdentityManager.exe" in the ActiveEntry installation directory.

➤ Manuals: Identity Management, IT Shop

5.2 ActiveEntry IT Shop

The ActiveEntry IT Shop is a web based application that provides stringent workflows for ActiveEntry users in the following areas:

- Change employee master data and own password
- Edit or enter employee master data for staff
- Search, request, cancel or renew products in IT Shop
- Delegate own roles
- Edit assigned approvals, attestation instances and rule violations

In the info system you see several evaluations e.g. about your own request and attestation instances, employee numbers, approvals, rule violation or the ActiveEntry Unified Namespace.

The ActiveEntry IT Shop requires a web server. Once the web server has been configured and a web project in Web Designer has been shared you can start the ActiveEntry IT Shop in your own web browser.

➤ Manual: Web Designer Manual (not yet available 5/11/2010)

5.3 ActiveEntry Manager

The ActiveEntry Manager is the administration tool for displaying and maintaining all ActiveEntry network information. This includes employee details, information over user accounts and user groups, hardware configuration data and organizational data. Applications are assigned to users and workdesks; drivers are assigned to workstations, machine types and workdesks. Objects with the same properties can be grouped together in dynamic roles. Every resource can be assigned to ActiveEntry users with a special request and approval procedure. In addition, the ActiveEntry Manager administrates the necessary basic information for synchronizing data between ActiveEntry databases and each target system, for example, domain properties or servers and their functions.

Start the program from the start menu "ActiveEntry\Manager" or via "Manager.exe" in the ActiveEntry installation directory.

➤ Manuals: Identity Management, IT Shop, Configuration, Service Management

5.4 DBMigrator

The DBMigrator tool is used to install a database on a Microsoft SQL server or an Oracle server for use in an ActiveEntry environment. This tool copies all the necessary tables, data types, database procedures. The database role "base group" is set up and given full rights for the objects in the database. There is an automatic version control integrated into ActiveEntry, which ensures consistency between the ActiveEntry elements and the database. If program updates are implemented that change the structure (e.g. table extensions) then the database has to perform a migration. The program "DBMigrator" carries this out depending on the current status of the database and enters the current status in the version control.

Start the program from the start menu "ActiveEntry\Datenbank\Database Migrator" or via "DB-Migrator.exe" in the ActiveEntry installation directory.

➤ Manuals: Getting Started

5.5 ActiveEntry Transporter

The ActiveEntry Transporter is used to transfer objects and custom changes as well as custom database procedures, triggers, functions and sets from the ActiveEntry database (source) to another ActiveEntry database (target).

Start the program from the start menu "ActiveEntry\Datenbank\Database Transport" or via "Transporter.exe" in the ActiveEntry installation directory.

➤ Manuals: Getting Started

5.6 DBCompiler

The ActiveEntry database has to be compiled after a successful migration and after changes to configuration data. The database compilation is started immediately from the DBMigrator or the ActiveEntry Transporter after a migration package or a customer's complete configuration package has been imported. The DBCompiler tool is used to compile the ActiveEntry database after importing hotfixes or when changes have been made to processes, scripts, formatting rules, object definitions, task definitions, and preprocessor relevant configuration parameters.

Start the program from the start menu "ActiveEntry\Datenbank\Database Compiler" or via "DB-Compiler.exe" in the ActiveEntry installation directory.

➤ Manuals: Getting Started

5.7 Crypto Configuration

In certain circumstances, it is necessary to archive encoded information in the database. The encoding is accomplished with the "Crypto Configuration" tool. This tool creates a code file and converts the contents of the affected database column. The coded information is stored in the database.

Start the program from the start menu "ActiveEntry\Datenbank\Crypto Configuration" or via "CryptoConfig.exe" in the ActiveEntry installation directory.

➤ Manuals: Getting Started

5.8 ActiveEntry Service Configurator

"ActiveEntry Service Configurator" is the tool used to create and customize the configuration file for ActiveEntry Service. ActiveEntry Service and its plug-ins are configured with this file. The configuration file is not only necessary for a Windows based operating system but for the Linux daemon as well.

Start the program from the start menu "ActiveEntry\Zielsystem\Job Service Configuration" or via "JobServiceConfigurator.exe" in the ActiveEntry installation directory.

➤ Manuals: Getting Started

5.9 ActiveEntry Service Updater

The "ActiveEntry Service Updater" tool can be used to update ActiveEntry Service on the Job server. This tool is used to set up requests to send the necessary files as well as to update the Job servers in the job queue.

Start the program from the start menu "ActiveEntry\Zielsystem\Update Job Services" or via "JobServiceUpdater.exe" in the ActiveEntry installation directory.

➤ Manuals: Getting Started

5.10 Job Queue Info

Job Queue Info supports the control of the current state of a service running on the ActiveEntry network. It displays requests in the job queue and the different requests ActiveEntry Service has on the servers in a detailed and comprehensive manner. The tool provides on-the-fly status information and makes fast error detection possible.

Start the program from the start menu "ActiveEntry\Monitoring\Job Queue Info" or via "Job-QueueInfo.exe" in the ActiveEntry installation directory.

➤ Manuals: Process Orchestration

5.11 Schema Extension

Schema Extension extends the existing application data schema of the ActiveEntry database with customer specific tables and columns. Using the object technology in ActiveEntry it is possible to do this on a database level such that these additions are available with full functionality at the object level.

Start the program from the start menu "ActiveEntry\Datenbank\Schema Extension" or via "SchemaExtension.exe" in the ActiveEntry installation directory.

➤ Manuals: Configuration

5.12 ActiveEntry Designer

ActiveEntry Designer is the ActiveEntry main configuration component. The program offers an overview from the entire ActiveEntry data model. It enables the configuration of global system settings, for example, language or configuration parameters as well as customization of the user interface for the different administration tools. The rights structure for different administrative tasks of individual users and user groups is also set up here. Another important task is the definition of workflows for technically illustrating the administration procedures in the company. ActiveEntry Designer provides several editors for configuring the ActiveEntry system. The range of functions and the operating methods of the editors matches the demands of differing configurations.

Start the program from the start menu "ActiveEntry\System Editoren\Designer" or via "Designer.exe" in the ActiveEntry installation directory.

➤ Manuals: Configuration

5.13 License Wizard

Use the License Wizard to set up requests for ActiveEntry licenses. Licensing is necessary after initial migration, when a license expires and when a new ActiveEntry version is installed.

Start the program from the start menu "ActiveEntry\Datenbank\Licence Management" or via "LicenceManager.exe" in the ActiveEntry installation directory.

➤ Manuals: Service Management

5.14 Software Loader

Using the Software Loader program, new or changed files, for example, customer specific form archives, are loaded into the ActiveEntry database. These files can then be distributed to the workstations and Job servers by the automatic software updating mechanism.

Start the program from the start menu "ActiveEntry\Datenbank\Software Loader" or via "SoftwareLoader.exe" in the ActiveEntry installation directory.

➤ Manuals: Getting Started

5.15 ActiveEntry Data Import

The ActiveEntry Data Import wizard allows you to import data from text files into the ActiveEntry database. Use the wizard if you want to import internal resources data from external source into your database.

Start the program from the start menu „ActiveEntry\Datenbank\Data Import“ or via „DataImporter.exe“ in the ActiveEntry installation directory.

➤ Manuals: Getting Started

5.16 Report Editor

The Report Editor is used by ActiveEntry to offer you the possibility of grouping object data together into reports. You can group, accumulate and graphically represent this data. Some of our own reports are supplied with the initial migration. However, you can also create your own reports with the report editor.

Start the program from the start menu „ActiveEntry\System Editoren\Report Editor“ or via „ReportEditor2.exe“ in the ActiveEntry installation directory.

➤ Manuals: Configuration

5.17 Script Debugger

You can use the Script Debugger to create, start and debug scripts. The scripts that already exist in the ActiveEntry database are imported into a VisualStudio ScriptLibrary. Here you can edit the scripts locally and test them. Then you need to decide whether the changes should be transferred to the ActiveEntry database or not.

Start the program from the start menu „ActiveEntry\System Editoren\Script Debugging“ or via „ScriptLibrary.sln“ in the ActiveEntry installation directory.

↗ Manuals: Configuration

5.18 ActiveEntry Analyzer

Use the ActiveEntry Analyzer to automatically detect and analyze data correlations in the database. This information can be used to replace direct permissions assignments with indirect assignments therefore reducing the administration effort.

Start the program from the start menu „ActiveEntry\Target System\Analyzer“ or via „Analyzer.exe“ in the ActiveEntry installation directory.

↗ Manuals: Identity Management

5.19 Web Designer

The Web Designer is ActiveEntry's tool for configuring and expanding an ActiveEntry IT Shop. It makes functions available for customizing ActiveEntry IT Shops and for designing new workflows.

↗ Manual: Web Designer Manual (not yet available 5/11/2010)

5.20 HistoryDB Manager

ActiveEntry historical data is transferred at regular intervals to a history database. Therefore, the history database provides an archive of change information. The tool that displays the data is called the "HistoryDB Manager". Use the HistoryDB Manager to setup access to the source databases.

Start the program from the start menu "HistoryDB\HistoryDB Manager" or via "HistoryDBManager.exe" in HistoryDB installation directory.

➤ Manuals: Configuration

Chapter 6 Logging into ActiveEntry Tools

When a login takes place, we differentiate between a database user and a user of individual ActiveEntry tools (system user ID). It is possible that there is more than one system user working with the same database account. There are two steps needed to login:

- Log into the database with a database user
- Log into a dialog with the ActiveEntry tools with a system user ID
System user login is not necessary for all ActiveEntry tools.

ActiveEntry checks whether you have a valid license during an ActiveEntry tool's login with a system user ID. If a valid license cannot be found, an appropriate message is shown. You can use the License Wizard to create a license request. See section [ActiveEntry Licensing](#).

6.1 Logging into the Database with a Database User

When an ActiveEntry tool starts up, the standard connection window is opened. Create a connection to the database by selecting the button "+". You need choose between connecting to a Microsoft SQL Server or an Oracle Server.



Figure 163: Connection Dialog at Program Startup

A dialog window is opened to enter the required connection data. To create a connection to the Microsoft SQL Server enter the server name and the database as well as the name and password of the database user. You can also use the <Options> menu to test the connection or change details of the database connection. Confirm the input with <OK>.



Figure 164: Input Mask for Connection data under Microsoft SQL Server

Use <Options> to open a pop-up menu with the entries <Test connections> and <Advanced options>. Use <Test connections> to test whether all the connection data is entered correctly. Use the <Advanced options> to alter database connection details.

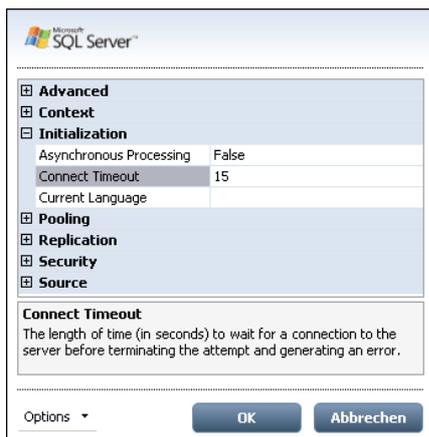


Figure 165: Database Connection Advanced Options

If we are dealing with a connection to an Oracle schema, enter the TNS alias name from the TNSNames.ora file into the input field <Data source>. This can be found in the local Oracle home directory under \Network\Admin. Continue by entering the name and password of the database user and confirm the input with the <OK> button.



Figure 166: Input Mask for Connection Data under Oracle

The database connection data is displayed in the <Connections> field when you log in the next time and you can select your database from there.

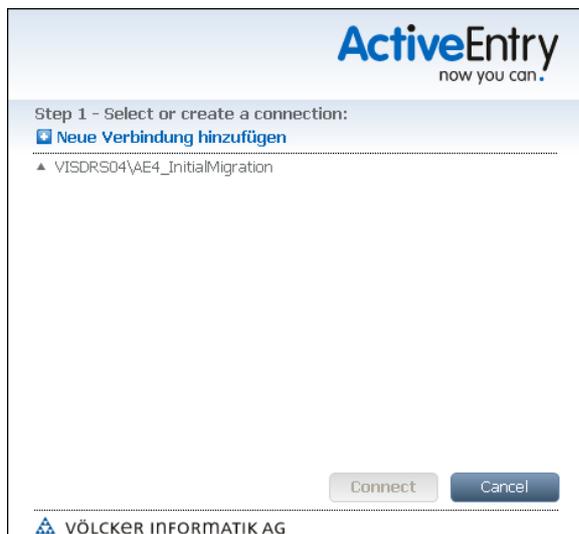


Figure 167: Selecting the Connection

6.2 Logging into ActiveEntry Administration Tool as System User

Following the database login, the user has to log into the administration tool as a system user. Permitted system user IDs are determined by the selected authentication module. ActiveEntry makes several authentication modules available. The following table show with which authentication modules you can login to the various administration tools.

Table 16: Authentication Module and ActiveEntry Tools

Authentication Module	ActiveEntry Tool
System user	ActiveEntry Designer ActiveEntry IT Shop ActiveEntry Manager
ComponentAuthenticator	
Employee	ActiveEntry Designer ActiveEntry IT Shop ActiveEntry Manager
Employee (dynamic)	ActiveEntry IT Shop ActiveEntry Manager
Employee (role based)	ActiveEntry IT Shop ActiveEntry Identity Manager
ADS user	ActiveEntry Designer ActiveEntry IT Shop ActiveEntry Manager
Web ADS user	ActiveEntry IT Shop ActiveEntry Manager

Table 16: Authentication Module and ActiveEntry Tools

Authentication Module	ActiveEntry Tool
ADS user (dynamic)	ActiveEntry IT Shop ActiveEntry Manager
ADS user (manual entry)	ActiveEntry IT Shop ActiveEntry Manager
ADS user (role based)	ActiveEntry IT Shop ActiveEntry Identity Manager
ADS user (manual entry/role based)	ActiveEntry IT Shop ActiveEntry Identity Manager
LDAP user (dynamic)	ActiveEntry IT Shop ActiveEntry Manager

6.2.1 Authentication Module

System User

This module is the basis of the standard login method for an ActiveEntry administration tool user. The login takes place using an existing ActiveEntry database system user and its password as system user ID.

By default, the system user "viadmin" (without password) is available after the initial migration. This system user can compile an initial ActiveEntry database and can be used to login to the administration tools for the first time. The system user "viadmin" has a user interface which is preset by Völcker Informatik and has the access rights to database resources. The interface and access rights for "viadmin" should not be used live or changed, as it is a template system user from Völcker Informatik and is overwritten by each migration. For further productive use of ActiveEntry, a password should be assigned. To do this, log into the ActiveEntry Manager as system user. Use the entry <Change password> in the menu <Database> to make changes to the password.

Component Authenticator

This module integrates the default method for registering process components. This module cannot be used to log into any ActiveEntry administration tools. Use the system user "sa" to register the process components. This system user has the necessary access rights for ActiveEntry process components. The system user "sa" should not be changed, as it is overwritten by Völcker Informatik by each migration.

Employee

Use an employee that already exists in the ActiveEntry database and log in over their system user ID. Only employees that are directly assigned a system user are allowed to login. The login password is the system user's password.

The user interface and the access permissions are loaded through the system user that is directly assigned to the logged in employee. Changes to the data can be assigned to the employee that is logged in. Use this authentication module when you want to have administrative control over ActiveEntry user's access permissions and to make their identity transparent.

Employee (dynamic)

This module uses the central user account of the current employee in the ActiveEntry database as system user ID. Enter the employee's system user password to login. As opposed to the authentication module "Employee", dynamic authentication does not use the system user that is directly entered in the employee data for the login, but a system user that is determined by the application configuration data. Thus, an employee can, for example, be assigned a system user dynamically depending on their department membership. You can find further information in section [Configuration Data for System User Account Dynamic Authentication](#) in the Configuration Manual.

The user interface and access permissions are loaded through the system user that is directly assigned to the logged in employee. Changes to the data can be assigned to the employee that is logged in.

Employee (Role Based)

This module uses the central user account of the current employee in the ActiveEntry database. Enter the employee's system user password to login. As opposed to the authentication module "Employee", role based authentication does not use the system user that is directly entered in the employee data for the login, but a dynamic system user from the employee's membership in ActiveEntry application roles.

The user interface and the access permissions are loaded through the system user that is directly assigned to the logged in employee. Changes to the data can be assigned to the employee that is logged in. The concept of application roles is described in more detail in section [ActiveEntry Application Roles](#).

ADS User

This module uses the employee currently logged into the workstation to log into the the ActiveEntry tools. A system user ID and password are not required for the login. The corresponding ADS user is found in the in the ActiveEntry database using the login name and the domain of the ADS container. ActiveEntry determines which employee is assigned to the ADS user.

The user interface and access permissions are loaded through the system user that is directly assigned to the logged in employee. Changes to the data can be assigned to the ADS user that is logged on.

The advantage of this module is that only one login is necessary in order to log into the workstation and the ActiveEntry administration tools. The password does not have to entered again. If the option <Connect automatically> is set, authentication is no longer necessary in subsequent logins.

Web ADS User

This module is carried out through the web front user that is currently logged on. From the context of the login at the web front, the corresponding ADS user is found in the ActiveEntry database. The employee is then determined that is assigned to the ADS user.

The user interface and access permissions are loaded through the system user that is directly assigned to the logged in employee. Changes to the data can be assigned to the ADS user that is logged on.

ADS User (dynamic)

This authentication module recognises the type of login that is being dealt with i.e. if it is a workstation login or a login over the web front. The login information for the ADS user is determined in the same way as the authentication modules "ADS user" and "web ADS user". The employee is determined that is assigned to the ADS user.

However for the dynamic authentication module, as opposed to the modules "ADS User" und "Web ADS User", the login is not via the employee that is entered as system user but a system user that is specified in the configuration data for the user interface. In this way an employee, depending on their department membership, can be dynamically assigned to a system user. There is more information in section [Configuration Data for System User Account Dynamic Authentication](#).

The user interface and access permissions are loaded through the system user that is directly assigned to the logged in employee. Changes to the data can be assigned to the ADS user that is logged in. If the option <Connect automatically> is set, authentication is no longer necessary for subsequent logins.

ADS User (manual input)

This authentication module does not use the user logged into the current workstation but prompts for manual entry of a system user ID. Use the login name that users log into the ADS domain with, for the system user ID. The user's identity is determined from a predefined list of permitted Active Directory domains. If successful, other login data are found as in the authentication module "ADS user (dynamic)". You specify the permitted Active Directory domains in the configuration parameter `NameSpace\ADS\AuthenticationDomains`.

ADS User (role based)

This authentication model allows you to use the user currently logged into the workstation to log into the ActiveEntry tools. The appropriate ADS user is found in the ActiveEntry database by the login name and the ADS container domain given at login. As opposed to the authentication module "ADS User", role based authentication does not use the system user that is directly entered in the employee data for the login, but a dynamic system user from the employee's membership in ActiveEntry application roles.

The user interface and the access permissions are loaded via this system user. Changes to the data can be assigned to the ADS user that is logged on. If the option <Connect automatically> is set, authentication is no longer necessary in subsequent logins. The concept of application roles is described in more detail in the section [ActiveEntry Application Roles](#).

ADS User (manual input/role based)

This authentication module requires manual entry of the system user ID at login. Use the login name that users log into the ADS domain with, for the system user ID. The user's identity is determined from a predefined list of permitted Active Directory domains. If successful, other login data are found as in the authentication module "ADS user (role based)". You specify the permitted Active Directory domains in the configuration parameter `NameSpace\ADS\AuthenticationDomains`.

LDAP User (dynamic)

This module uses the LDAP user’s login name as system user ID. Enter the password that is saved in the LDAP user’s master data as password. ActiveEntry finds the employee that is assigned to the LDAP user. ActiveEntry determines the system user from the application configuration data. You can find further information in the section [Configuration Data for System User Account Dynamic Authentication](#) in the Configuration Manual.

The user interface and the access permissions are loaded via the system user that the logged in employee is dynamically assigned to. Changes to the data can be assigned to the ADS user that is logged in.

6.2.2 Login

Follow the procedure described here to perform an ActiveEntry tools login. Please take into account: only those authentication modules are shown in the login window that you have selected for the login. Read the section [Authentication Module](#) in the Configuration Manual to find out how to setup more authentication modules for the login.

NOTE! After initial migration, only system user (DialogUser) authentication modules, component authenticators and role based authentication modules are enabled.

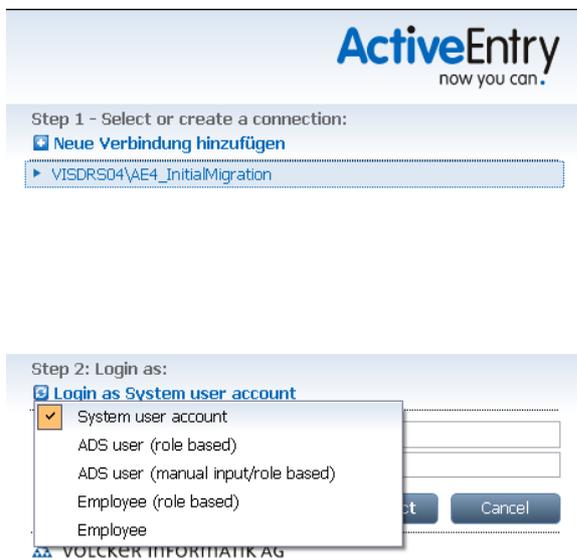


Figure 168: Connection Dialog with Login to Administration Tools

1. Click with the mouse on the icon under "Step 2 - login as:"
A selection list is displayed that shows all available authentication modules.
2. Select an authentication module
The permitted system user IDs are determined via the authentication module.

3. Enter system user ID and password in the <User> and <Password> fields.

If you login successfully, the connection data is saved and is available the next time you login. If you have entered a system user ID that is not supported by the selected authentication module, the follow error message appears.

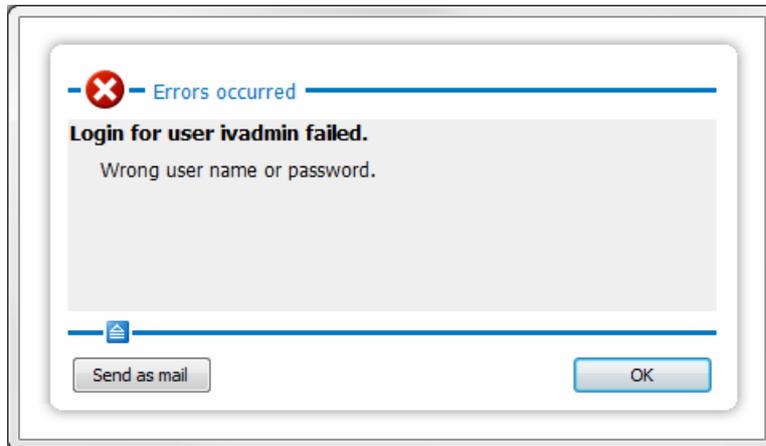


Figure 169: Error Message Logging in with a System User ID

Repeat the Login by selecting another authentication module or system user ID.

Chapter 7 ActiveEntry Tools User Interface

The user interface of ActiveEntry's main tools - ActiveEntry Manager and ActiveEntry Identity Manager - is identical. The components that make up the user interface for both of these tools are described in this chapter. The user interface of the ActiveEntry Designer is also basically the same. The elements of the user interface that are different are described in the Configuration manual in section [ActiveEntry Designer User Interface Layout](#).

Certain components of the ActiveEntry Manager's graphical user interface are stored in the database and can be tailored to suit customer requirements. Menu items in the navigation structure, interface forms and task definitions can be configured in this way. Menu items, interface forms and task definitions are assigned to permissions groups. If a user logs into the ActiveEntry Manager with a system ID (system user), the menu items, interface forms, task definitions and separate program functions that are displayed depend on the role memberships the system user belongs to. The user interface is customized for the user when it is loaded.

The various possibilities to customize the user interface to suit your requirements are described in detail in the Configuration manual in chapter [Editing the Administration Tools Graphical User Interface](#).

Data stored in the database is displayed in the form of objects. User interface objects are meta-objects. They provide a selection of configurable elements that mirror the data stored in the database. These objects allow data to be differentiated depending on particular properties. Therefore, they offer an additional functionality for controlling the appearance of the user interface. This means that interface forms and tasks are linked to an object and are only available when the object is selected.

7.1 The User Interface Layout

You can control the ActiveEntry tools' graphical user interface with the mouse and with keyboard combinations. For an optimal graphics display, we recommend a minimum screen resolution of 1280 x 1024 pixels and at least 16-bit color.

The user interface contains a title bar, a status bar, an edit field, a menu bar and different toolbars. There are different views for displaying and processing data defined within the edit interface.

7.1.1 Title Bar

The title bar shows the program icon, the program name and the database connected in the notation <user>@<database server>\<database (description)>.



Figure 170: Program's Title Bar

7.1.2 Status Bar

The status bar is used to display the names of connected databases in the notation <server>\<database (description)> and the currently connected system user. The system status is also shown. A status icon indicates database activities such as loading or saving objects. The name of the current form can also be displayed with the notation <user interface form>[(<form template>)] by enabling the <Show additional navigation information> option in the program settings. Additional icons are used to show quick edit mode, field definitions and simulation mode. Likewise, the user is alerted to new entries in the error log by the appearance of an icon in the status bar.

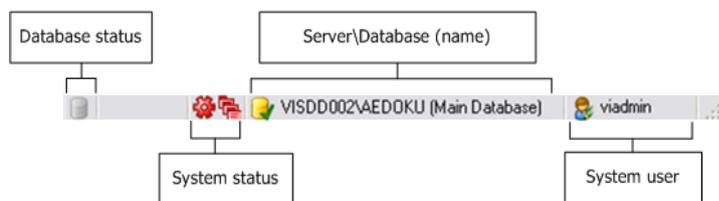


Figure 171: The Program's Status Bar - Simple

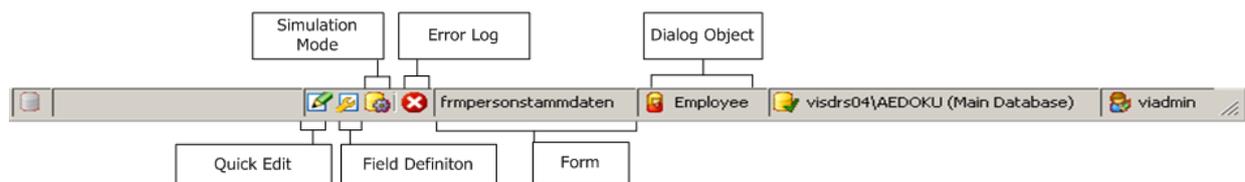


Figure 172: The Program's Status Bar - Extended

Table 17: Icons in the Status Bar

Icon	Meaning
	System user without VI configuration entitlements.
	System user with temporary VI configuration entitlements.
	The DBScheduler was stopped.
	The services were stopped.

Table 17: Icons in the Status Bar

Icon	Meaning
	The database is connected.
	Database status and indicator for the DBSchedulers calculation tasks.
	Quick edit is activated.
	Field definitions (table and column descriptions) are displayed.
	Program is in simulation mode.
	A warning has been entered in the error log.
	A error has been entered in the error log.

7.1.3 Menu Bar

The menu bar contains different menus, The <database> and <help> menus are always displayed. The <object> and <view> menus are only enabled when a database is connected.

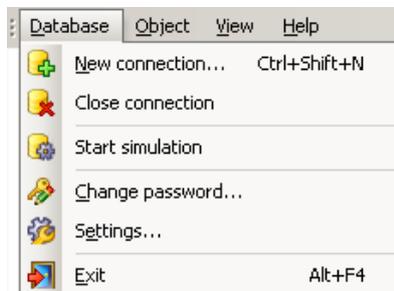


Figure 173: The Program Menu Bar

Table 18: General Key Combinations in the Menu Bar

Key Combination	Action
Alt + underlined letter	Select menu
Up arrow, down arrow	Move between items in a menu
Enter	Select a menu item
Esc	Cancel the menu

Table 19: The Meaning of Menu Bar Entries

Menu	Menu Item	Meaning	Key Combination
Database	New Connection	Establish a new database connection.	Ctrl + Shift + N
	Close Connection	Close the current database connect.	
	Start/Stop Simulation	Switches program in and out of the simulation or work mode. This option is available when the logged in user is authorized to user this program function.	
	Export data	Opens a form for specifying the export data. This option is available when the logged in user is authorized to user this program function.	
	Show deferred operations	Shows deferred operations. This item is available if the configuration parameter "Common\DeferredOperation" is enabled.	
	Check data inconsistencies	Opens a form for checking data consistency. This option is available when the logged in user is authorized to user this program function.	
	Change password...	The current user's password can be changed.	
	Settings...	Program settings can be configured.	
	Exit	Exit out of the program.	Alt + F4

Table 19: The Meaning of Menu Bar Entries

Menu	Menu Item	Meaning	Key Combination
Object	New	Inserts a new object of the type that is currently displayed	Ctrl + N
	Save	Saves changes to an object	Ctrl + S
	Schedule changes...	Allows changes to be scheduled. This item is available when the configuration parameter "Common\DeferredOperation" is set.	
	Delete	Deletes the selected object	Ctrl + D
	Schedule deletion...	Allows deletion to be scheduled. This item is available when the configuration parameter "Common\DeferredOperation" is set.	
	Undo delete	Prompts whether the delete should be reversed. This option is only available if the object is labeled for deletion.	
	Discard changes	Discard changes in the form which are not yet saved	Ctrl + Shift + D
	Reload	Loads the object again	Ctrl + F5
	Properties	Displays more properties for the current object. This entry is available in advanced mode.	
	Reapply templates	All object templates are reapplied. This menu item is only available on the mater data form for the object.	
	Add to/Remove from to favorites	Insert/Delete the current form into/from the favorites list.	Ctrl + B
	Backward	Shows the previous form the order that the forms were viewed	Alt + left arrow
	Forwards	Shows next form in the order that they have already been viewed.	Alt + right arrow

Table 19: The Meaning of Menu Bar Entries

Menu	Menu Item	Meaning	Key Combination
View	Navigation	The navigation view.	Ctrl + Q
	Result list	Activates the result list.	Ctrl + R
	Documents	Activates the current document.	Ctrl + O
	Favorites	Show/hide favorites list.	
	Tasks	Show/hide tasks field.	
	Error log	Show/hide the error log.	
	Process information	Show/hide the process view. This entry is available when the configuration parameter "Common\ProcessState" is set and the user has permissions to use this program function.	
	TimeTrace	The TimeTrace view is either shown or hidden. This entry is available when the configuration parameter "Common\ProcessState" is set and the user has permissions to use this functionality.	
	Database search	Show/hide the database search function.	
	Home	Activate the home page.	
	Layout	Changes the layout of the program interface. There is a choice between the default layout and two predefined views. It is only possible to restore the default layout in 'advanced mode'.	
	Enable quick edit	Enables/disables quick editing of objects. This entry is only takes effect temporarily. If quick edit mode should be enabled/disabled as default the program settings have to altered (see section User Settings)	
	Show field definitons	Show/hide field definitons. This menu item is only available in advanced mode.	
Show navigation data	Shows the loaded navigation data. This menu item is only available when the program setting "additional navigation information" is selected.		
Plugins	Plugin name	Displays plugins available for use. Plugins are enabled or disabled in the program settings (see Available Plugins).	
Help	Form help	Opens help window for the currently displayed form.	F1
	Info	Shows the version information for the program.	
	License information...	Show the current license information.	

7.1.4 Toolbars

The ActiveEntry tools feature a standard toolbar and a toolbar relating to forms. The icons are enabled or disabled depending on the current view. Some icons have a menu selection. This can be opened using the arrow next to the respective icon. The standard toolbar comprises of functions, which are effective independent of the form that is currently on view. All the functions on the forms related toolbar refer to the current form.



Figure 174: Toolbar

Table 20: Functions in the Standard Toolbar

Icon	Meaning
	Show home page.
	Shows the previous form in the order that the forms were viewed. You will find the forms history in the drop-down menu. Use it to chose any one of the forms.
	Shows next form in the order that they have already been viewed. You will find the forms history in the dop-down menu. Use it to chose any one of the forms.
	Show/hide database search.

Table 21: Functions in the Forms Related Toolbar

Icon	Function
	Insert a new object of object type currently displayed
	Save changes to an object
	Schedule the time for saving object changes. This option is available when the configuration parameter "Common\DeferredOperation" is set.
	Delete selected object
	Schedule deletion of selected object. This item is available if the configuration parameter "Common\DeferredOperation" is enabled.
	Reverses the deletion procedure. This option is only available when the object is marked for deletion.
	Discard unsaved input in a form
	Reload object
	Insert the current object in the favorites list

Table 21: Functions in the Forms Related Toolbar

Icon	Function
	Remove the current object from the favorites list
	Print current form

7.1.5 Context Menus

Some elements have a separate context menu. Context menus are opened with the hot key <shift+F10>, the context menu key or the right mouse button. The contents of the menu depends on the view that is being displayed at the time.

Table 22: Standard Key Combinations for the Context Menu

Key Combination	Action
Shift + F10 or context key	Call context menu
Up arrow, down arrow	Move between items in a context menu
Enter	Select a context menu item
Esc	Cancel the context menu

7.1.6 Structure of the Edit Interface

There are different views defined in the edit interface of the program for displaying and editing data.

- **Navigation view**
In the navigation view, entry points for the interface navigation are predefined for the user that is logged in. The functions in this view are explained in the section [Functions in the Navigation View](#).
- **Result list**
When a menu item is selected in the navigation view, all objects that correspond to the object definition and conditions of the menu item are displayed in the result list. The functions in this view are explained in the section [Functions in the Result List](#) erläutert.
- **Document view**
Forms for editing a selected object are displayed in the document view. The functions in this view are explained in the section [Functions in the Document View](#).
- **Task view**
The tasks view displays the tasks and reports that are available for the selected object. The functions in this view are explained in the section [Functions in the Tasks View](#).

- Favorites list
This view is used to group individual menu items into a favorites list for the user. The functions in this view are explained in section [Functions in the Favorites List](#).
- Error log
Errors and warnings that occur while working with the ActiveEntry Manager or the ActiveEntry Identity Manager are recorded in an error log. The functions in this view are explained in section [Logging Error Messages](#).
- Process view
The process view is used to evaluate process information that results from changes to data. The functions in the view are explained in section [Evaluating the Process Monitoring Information](#).

7.1.7 Selecting the Display Mode

You can display the edit interface in the ActiveEntry Manager and the ActiveEntry Identity Manager in simple or advanced mode. Modes differ in the view layout. Specify which layout you want with the program setting <Windows dockable>. (see section [User Settings](#))

In simple display mode there are two other layouts possible apart from the standard one. You can select these using the menu item <View>\<Layout>.

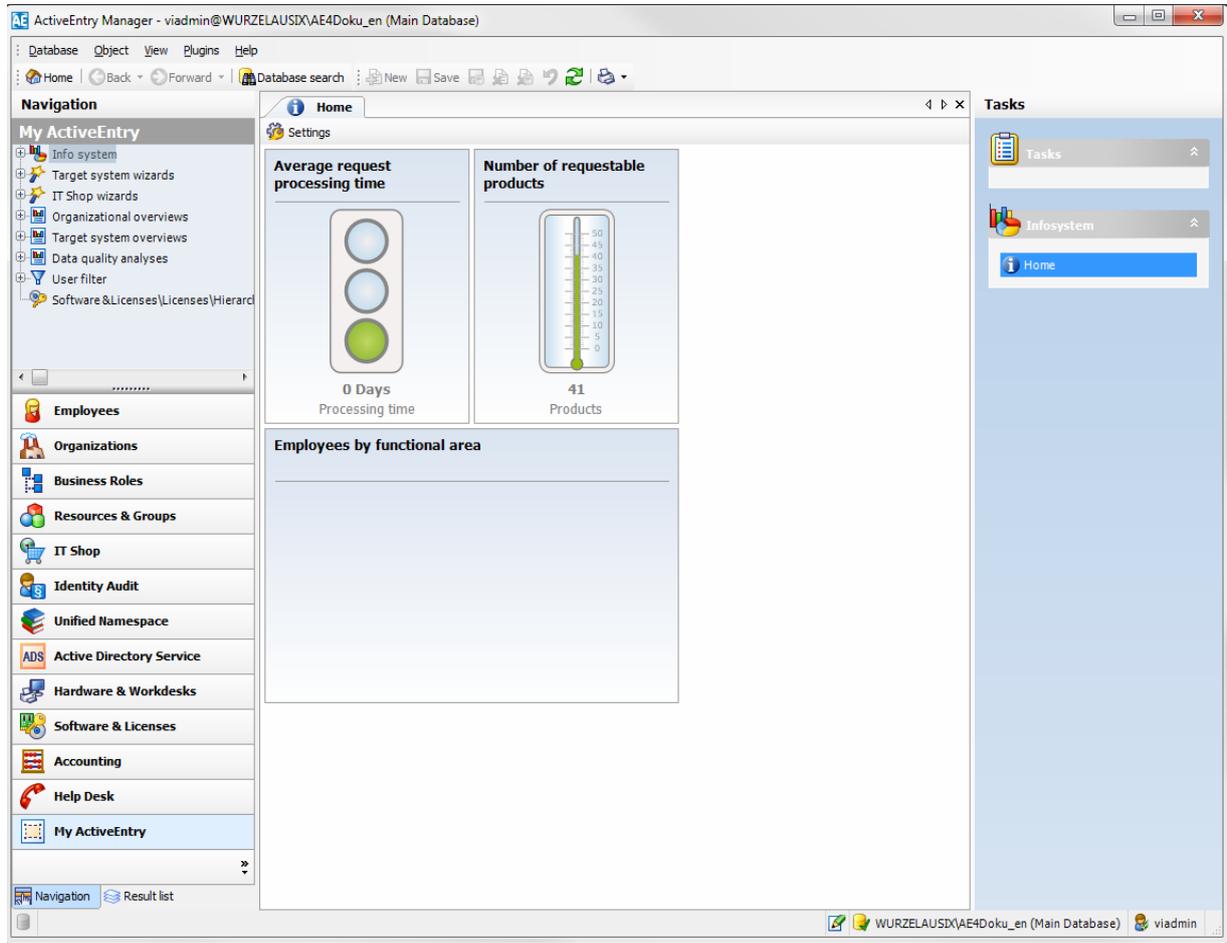


Figure 175: Default Program Layout in Simple Mode

In advanced mode, the position and size of the windows in the working area can be changed. The "Auto-Hide" mode can be activated or de-activated using the pin in the window's title bar. If this mode is activated, the selected window can slide in or out of its position. An icon in the navigation bar at the side allows a window in "Auto-Hide" mode to be selected.

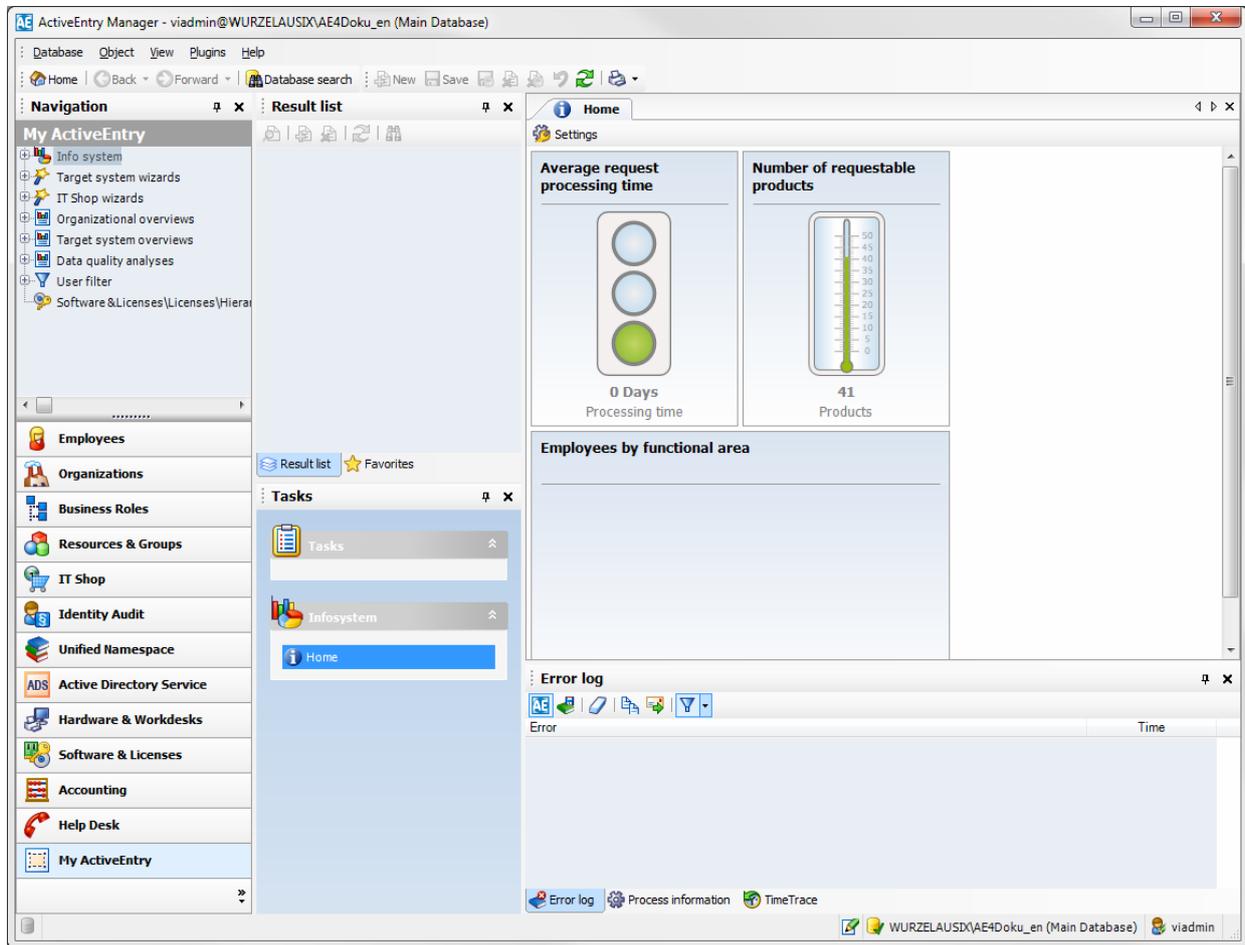


Figure 176: Standard Program layout in Advanced mode

The layout of the currently logged in user is saved in the user configuration, such that the last used layout is displayed the next time the program is started.

7.1.8 Searching for List Entries

The search dialog makes it possible to search for entries in a result list. The search dialog is opened using the context menu, the icon in the result list or <Ctl+F>.

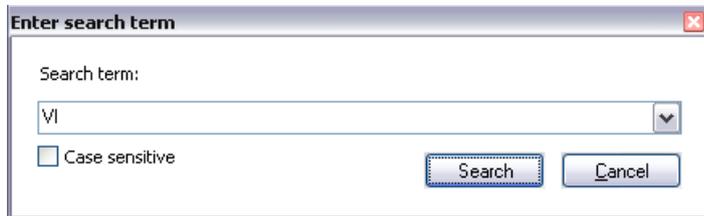


Figure 177: Search Dialog

You can also specify case sensitive for the given search string. Click on the <Search> button to jump to the first element that contains the search string and the search dialog is closed. <F3> continues the search. The button <Cancel> aborts the search.

Table 23: Key Combinations for the Search Dialog

Key Combination	Action
Strg + F	Open search dialog
Enter	Start search
Esc	End search
F3	Search next

For a wider ranging search, use the database search function. Refer to section [Implementing User Defined Filters for Database Searches](#) for more information.

7.1.9 Limiting the Number of Results

Table 24: Configuration Parameter for Limiting Results

Configuration Parameter	Effect
Common\DBConnection\ListLimit	This configuration parameter specifies the number of list entries above which the filter request becomes effective
Common\DBConnection\WebListLimit	This configuration parameter specifies the number of list entries above which the filter request in the web front becomes effective.

A filter dialog is implemented in the ActiveEntry Manager to limit the number of elements displayed in the result list or in control elements with list values (e.g. in pop-up lists). To use this, enable the option <Enable list limit> in the program settings. (see section [User Settings](#))

The number of entries above which the filter dialog comes into effect is globally fixed in the configuration parameter "Common\DBConnection>ListLimit". In addition, the logged in user can enter a limit in the program settings. This is saved in the user configuration and takes precedence over the global value.

If the number of results exceeds the limit, the filter dialog is opened.

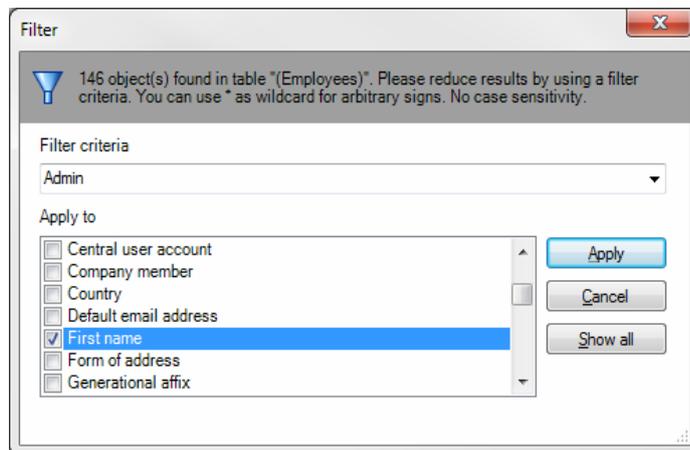


Figure 178: Filter Dialog

To limit the entries in the result list a <Filter criteria> can be entered. The wildcard * (star) may be used here. The filter is not case sensitive. In the dialog field <Apply> the properties that are applied by the filter criteria are shown. The desired entries can be activated and de-activated by clicking the check box. When you press the <Apply> button, the results are filtered and the number of results displayed is limited respectively. If you select the <Show all> button, all requested results are shown without taking the filter criteria into account. Choosing the <Cancel> button aborts the action and results in an empty result list. The option <Only show actual assignment> is shown additionally for list with object assignments. By setting this option, only those elements are shown that meet the filter condition and are already assigned to the base object.

Example:

This filter condition is used to search for all parts of an employee's full name. This is put together in the following manner:

Full name:
 surname, first name

Example:
 Little, Max
 Little-Maynard, Max
 Dolittle, Max

Filter condition: Little or *Little*
 Searches for all people whose full name contains "Little" anywhere in the string.

Finds, according to example:

Little, Max
 Little-Maynard, Max
 Dolittle, Max

Filter condition: Little*

Searches for all people whose full name starts with the string "Little".

Finds, according to example:

Little, Max
 Little-Maynard, Max

Filter condition: *Little

Searches for all people whose full name ends with the string "Little"

Finds, according to example:

no entry

7.1.10 Changing the Password for the Current User

Select the menu entry <Database>\<Change password...> to change the password for the currently logged on system user. Enter the old password and the new password, then re-enter the new password. The changes are put into effect with <OK>.



Figure 179: The Change User Password Dialog

7.1.11 Modifying the Program Settings

The following settings can be adjusted over the menu item <Database>\<Settings...>:

- User settings
- Special features for individual program components
- Standard Program settings
- Available plugins

The general configuration settings can be predefined in the configuration files "Manager.exe.config" and "IdentityManager.exe.config". In addition, global configuration settings can be defined using VI's own format in a configuration file. You can find examples of configuration files in the Configuration Manual in section [Administration Tool Configuration Files](#).

User Settings

Enter the program settings for the user on the <User> tab. These settings are stored in the ActiveEntry database user configuration.

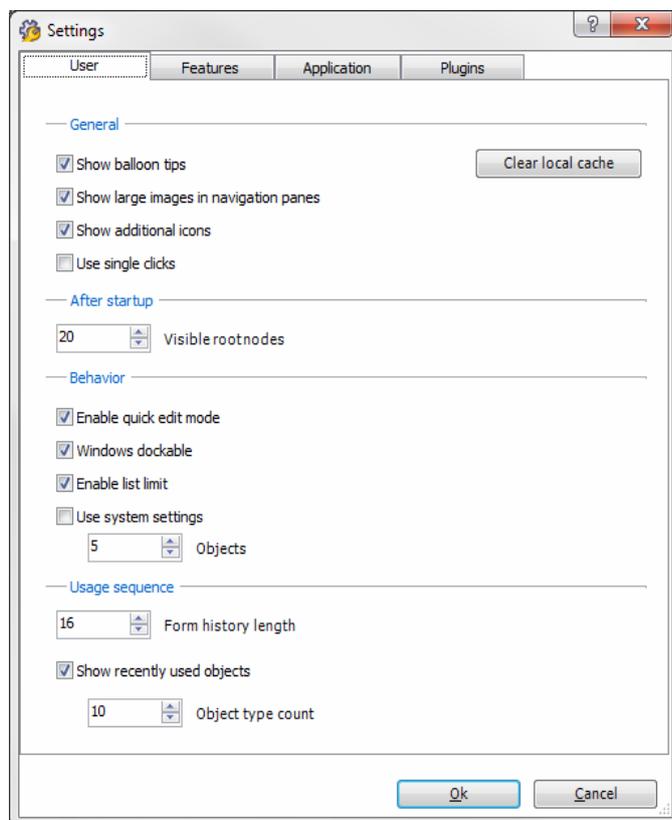


Figure 180: User Settings

- **Show balloon help**
Speech bubbles, which provide information about program functionality, are shown or not shown in the program depending on the settings.
- **Show large images in the navigation**
Large or small icons are shown on the navigation view title bars depending on the settings.
- **Show additional icons**
If this option is set, icons are shown in the task list in addition to the descriptions.
- **Use single clicks**
Objects are selected from the result list either with a double click or a single click depending on the setting.
- **Clear local cache**
The local cache directory "C:\Documents und Settings\\local settings\application data\Voelcker\Cache" is emptied.
- **Visible root nodes**
This is the number of category bars shown in the navigation view at start up. Changes to this item become effective after a restart.
- **Enable quick edit mode**
By default, an object's overview form is displayed first. To speed up data editing it is possible to show the first editable form instead. In order to do this quick edit mode has to be enabled. Quick edit mode is indicated by an additional icon in the program's status bar. This setting takes effect when the program starts up.
- **Dockable Windows**
Setting this option switches the program from standard view mode to advanced mode. The modes differentiate in the standard layout. In advanced mode, the position and size of the windows in the editable area within the user interface can be changed.
- **Enable list limit**
This setting can be enabled to limit the number of elements in a result list as well as control elements with list values (filter dialog).
- **Use system settings/objects**
If the list limit is set then the number of elements has to be entered. There is a choice between the global system setting or the user's own setting. If the number of results is greater than the defined number a filter dialog is opened.
- **Form history length**
Sets the number of forms that are available for viewing in the form history. You find the form history in the drop-down menu attached to the forward and back buttons on the standard toolbar.
- **Recently used objects/Number per object type**
When set, recently used objects are displayed in a separate node. At the same time, the number is set of objects per object type combined in the node.

Special Settings for Individual Program Components

Enter special program settings for individual components on the <Features> tab, which refers to the process view and is only displayed when this functionality is enabled. These settings are described in the section [Configuring the Process Display](#). The settings are stored in the ActiveEntry database user configuration.

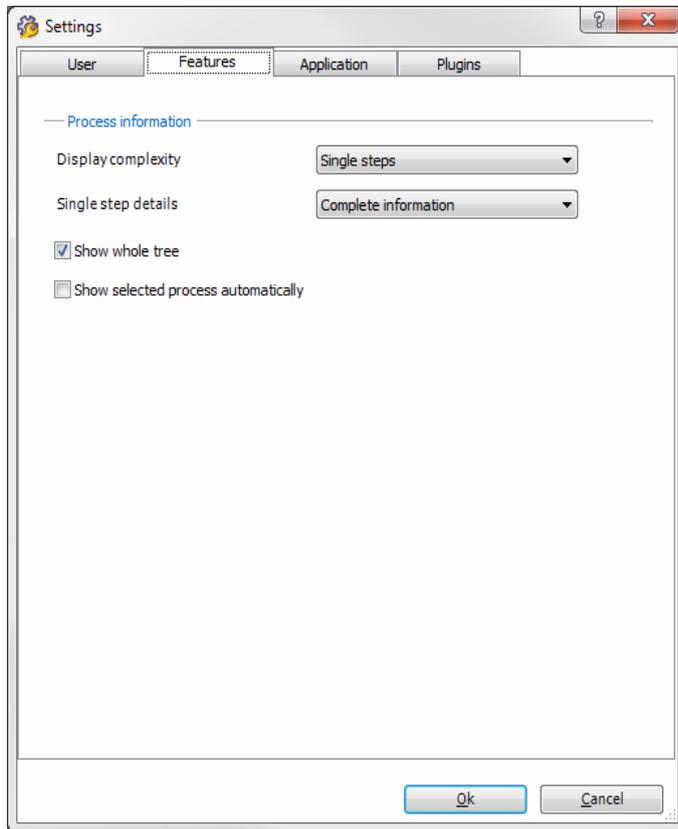


Figure 181: Special features of individual Program Components

General Settings

Enter general program settings on the <Application> tab. These settings are saved in the registry database on the workstation.

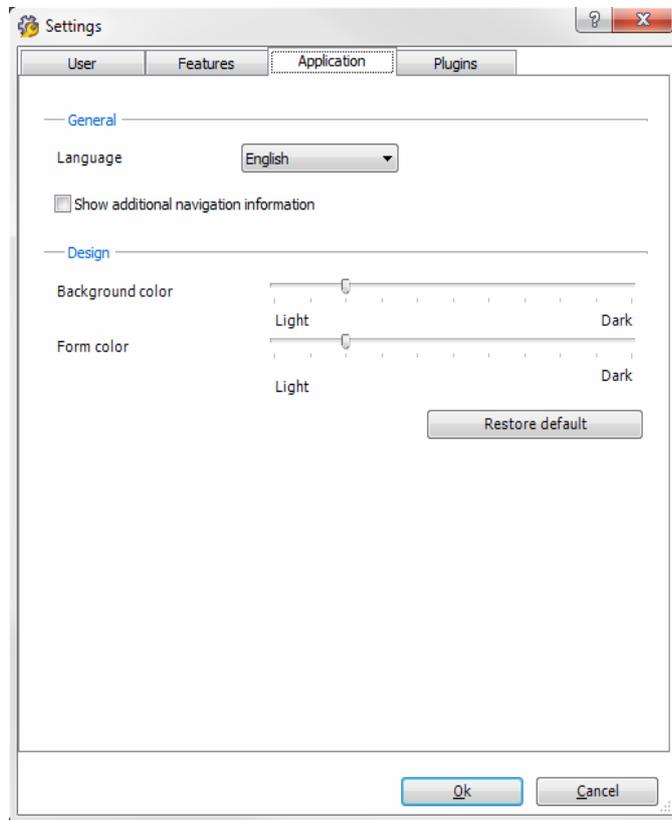


Figure 182: Default Application Settings

- Language**
 The initial program login uses the system language for the user interface. Changes to the language settings take effect after the program has been restarted. This sets the language globally for all ActiveEntry programs which means that the language setting do not have to be configured for each program individually. Refer to section [Languages for Displaying and Maintaining Data](#) in the Configuration Manual for more information.
- Show additional navigation information**
 If this option is set, additional navigation information for separate interface components is shown, for example the form name in the status bar and the definition of the menu item. Preprocessor configuration parameters that are connected to the objects are also shown here. Additionally in the programs's menu bar under the menu item <View> the item <Show loaded interfaces> is on offer.
 The option is not saved permanently. It has to be reset each time the program is started.
- Brightness of the background color and the form color**
 The brightness of the background color and the form can be set using the slide rule. The default can be reloaded over the corresponding control.

Available Plugins

The plugins that are available in the program are displayed on the <Plugins> tab with a short description of their function. You can enable or disable plugins. Plugins are displayed in the "Plugins" menu and can be started from there.

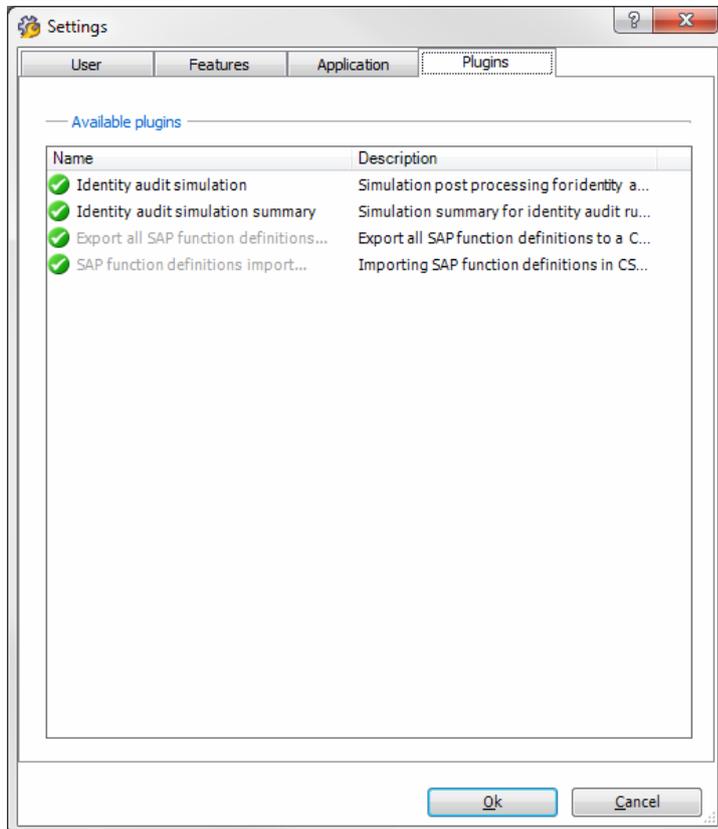


Figure 183: Available Plugins

7.2 Functions in the Navigation View

The navigation structure of the user interface is hierarchical and allows user oriented navigation right down to the selection of an object definition. The top level in the hierarchy, classifies ActiveEntry data into specified categories. You select the category by clicking on a bar in the category list.

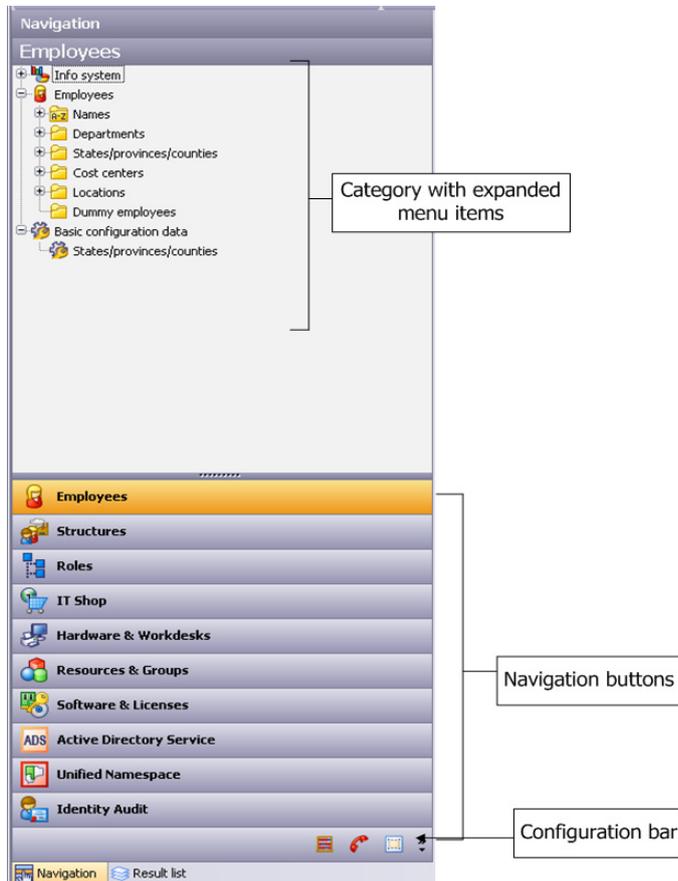


Figure 184: View of the Navigation Hierarchy Using the Category <Employee> as Example

The following types of menu items are used in a category:

- Fixed menu items, which sort the object definitions for a category by different criteria.
- Data dependent menu items for representing a hierarchical structure within a table.

If menu items are linked to preprocessor configuration parameters, the names of the configuration parameters are displayed after the menu item. Enable the option "Show additional navigation information" in the program settings in order to do this.

Special mouse behavior is integrated:

- An item in the navigation menu can be opened or closed by double clicking on the item name.

- With a single click on the item name a result list is displayed for the object, assuming that an appropriate object of this interface type is found. If no objects are found for a defined item, an empty result list is shown.

Table 25: Standard Key Combinations for the Navigation View

Key Combination	Action
Strg + Q	Enable navigation
Up arrow, down arrow, PgUp, PgDn, Home, End	Move with navigation structure
<+> or right arrow	Open menu item
<-> or left arrow	Close menu item
Enter	Select an item
F5	Refresh navigation view

The navigation view possesses a separate context menu which is shown by clicking with the right mouse button on the object. The menu items are shown or hidden depending on the selected object.

Table 26: Items in the Navigation View Context Menu

Item in Context Menu	Meaning
New	Insert a new object of the selected type. The existing preallocated insert values are taken over.
Search	Objects can be searched for within the navigation menu.
Add to my ActiveEntry	The selected item can be transferred into the category <My ActiveEntry>. This makes it possible to navigate faster with frequently used items. The information is saved in the user configuration.
Remove from my ActiveEntry	The selected item is deleted from the category <My ActiveEntry>. The information is saved in the user configuration.
Add to favorites	Insert selected object into the favorites.
Remove from favorites	Delete selected object from the favorites
Show process information	Shows process information for the selected object in a separate view.
Tasks	Show task list for the selected object. The required task can be run from the list.
Definition	Shows extended information about the configuration of the item. This menu item is only available in advanced mode.
Show deferred operations	Shows operations deferred in this category. This item is available if the configuration parameter "Common\DeferredOperation" is enabled.

You also see the following items in the <My ActiveEntry> category context menu::

Table 27: Items in the <My ActiveEntry> Category Context Menu

Item in Context Menu	Meaning
New filter	Save database query search criterion in a filter.
Filter bearbeiten	Edit existing filters.
Filter entfernen	Delete existing filters.
Filter exportieren	Export existing filters in XML format.
Filter importieren	You can import filters into ActiveEntry from XML files.

The navigation view has a configuration bar at the bottom with a configuration menu.

Table 28: Items in the Navigation View Configuration Menu

Item in Configuration Menu	Meaning
Show more Categories	Show left most category in the configuration bar in the navigation view. The icon is removed from configuration bar. When all categories are visible, the entry is deactivated in the configuration menu.
Show fewer Categories	Hide last category shown in the navigation view and iconize it in the configuration bar. When no more categories are visible, the entry is deactivated in the configuration menu.
Settings...	Shows programming settings in a dialog box.
Add or remove categories	Add or remove categories from the list in the navigation view (filter functions).

7.3 Functions in the Result List

When a menu item is selected in the navigation view, all objects that conform to the definition and conditions of the item are displayed in the result list. When an item is selected in the result list, the object is loaded and the first available form is shown in the document view. If there is no object assigned to the selected item, the last used form remains in the document view.

A list title is displayed in the title bar. The title depends on the items in the navigation hierarchy that were used to select the object. The list title also displays the number of elements in the list. The objects that were used last are displayed in the <Recently used> list depending on the program setting <Show recently used objects>. New objects are shown under the menu item <Newly added>.



Figure 185: Result List

Objects are loaded by double click as default. Set the option <use single clicks> in the program setting to load objects with one mouse click.

Use the key combination <shift> + 'select object' to open several forms for one or more objects in the same result list. This way you can quickly swap between objects without having to reload the object from the result list.

To edit several objects of the same type, select the ones you want in the result list (<Shift or Ctrl> + select) and open the master data form with <Enter> or using the context menu item <Tasks>/<Edit master data>. Refer to section [Multiple Object Edit](#) for more information.

Table 29: Standard Key Combinations for the Result List

Key Combination	Action
Ctrl + R	Enable result list
Up arrow, down arrow, PgUp, PgDn, Home, End	Move around in list
Enter, Shift + Enter	Select an entry
<Shift or. Ctrl> + select + enter	Select several entries
F5	Updating the result list

The result list features its own tool bar and context menu.

Table 30: Functions in the Toolbar

Icon	Function
	Shows the selected objects and allows multiple editing.
	Insert a new object
	Delete selected object(s)
	Reverse the deletion procedure
	Reloads the current object
	Searches within the result list
	Export result list (see Exporting Data)

Table 31: Entries in Result List context Menu

Entry in Context Menu	Meaning
New	Insert a new object of the selected interface type. The existing preallocated insert values are taken over.
Delete	Deletes selected object after the security alert is confirmed.
Undo delete	The deletion procedure for the selected object is revoked after confirming the security alert. This menu item is only available when the object is marked for deletion.
Search...	Search within the result list.
Add to favorites	Insert selected object into the favorites.
Remove from favorites	Delete selected object from the favorites
Show process information	Shows process information for the selected object in a separate view.
Tasks	Show task list for the selected object. The required task can be run from the list.
Properties	Show more properties for the current object. This menu item is only available in advanced mode.

7.3.1 Special Consistency Icons

Special icons are displayed in the result list for objects which are newly inserted, changed or marked to be deleted. This is valid for all database tables which contain a column "vi_consistent". If ActiveEntry Service has already declared the actions in the respective target system (vi_consistent = P), the icons are displayed that are specified during interface editing.

Table 32: Use of special consistency icons

Icon	Action	Display
	Insert	Insert new data set (vi_consistent =I)
	Delete	Mark data set to be deleted (vi_consistent=D)
	Change	Changes made in data set but not yet propagted in the target system (vi_consistent=U)

7.4 Functions in the Document View

When an object is selected in the result list, the corresponding form is shown in the document view. At least one form can be displayed for each object:

- Overview form
This form is only used to show selected information about an object.
- Master data
This form is editable and the object details can be changed here.

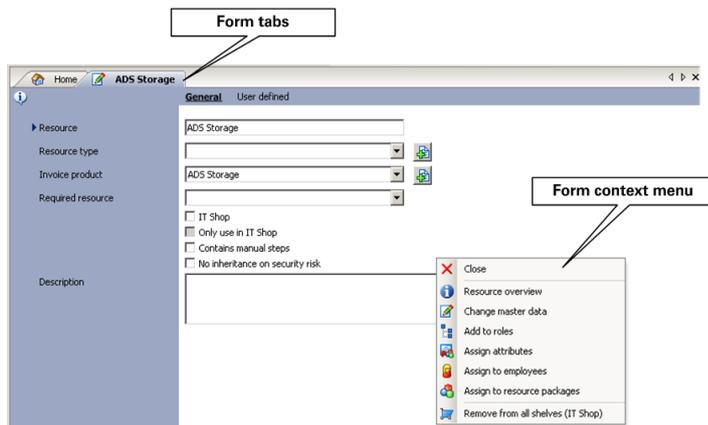


Figure 186: Master Data Form Layout

Each form features a form tab that displays the object as specified by a predefined format.

There may also be other forms. The forms context menu or the task view can be used to change between forms. The control elements on the forms are described in more detail in the section [Control Elements Used on Forms](#).

Use the key combination <shift> + 'select object' to open several objects in the same result list. This way you can quickly swap between objects within a category without having to reload the objects from the result list. When you change categories, the forms that are currently open in one category remain on display.

The overview form is shown first for all objects by default. However, to edit data more quickly it is possible to show the edit form for the object first. Use the combination <Alt> + 'select object' to open the edit form for the object first. Set the option <Enable quick edit> if you want to ensure that the edit form is always opened first.

On the forms context menu you can see the tasks that are available for the selected object. If you are in advanced mode, you can also see other properties of the current object by selecting the <Properties> item in the forms context menu.

7.4.1 Getting Form Help

Each form has dynamic help available. The mouse cursor changes into a help icon when the help icon on the form is clicked on. The next click on a field description causes a help description for the field to be displayed in a help box.

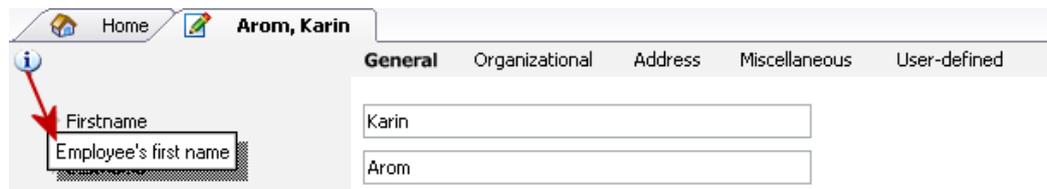


Figure 187: Help box for a data field

Use the F1-help to switch to the CHM-help, which is linked to the themes in the form. If more than one theme is relevant, they are shown in a selection dialog.

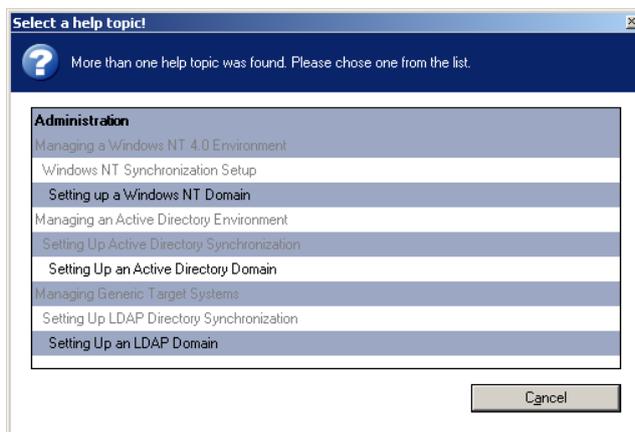


Figure 188: Selection Dialog showing Help Themes for a Form

7.4.2 Control Elements Used on Forms

There are different control elements for displaying and editing the current data. The most important control elements are described in the following sections.

Table 33: Standard Key Combinations for Control Elements

Key Combination	Action
Tab or Shift + Tab, left arrow, right arrow, up arrow, down arrow	Moving within the control elements
Up arrow, down arrow, PgUp, PgDn, Home, End	Moving within lists and hierarchical structures
<+> or right arrow	Open hierarchy level
<-> or left arrow	Close hierarchy level
Space or enter	Select entry
Shift + enter	Select several entries
F4 or Alt + down arrow	Open pop-up menu
Del	Delete entry from pop-up menu

Form Elements

Form elements are used to display information on an overview form. Form elements are mapped via a hierarchical structure of menu items.

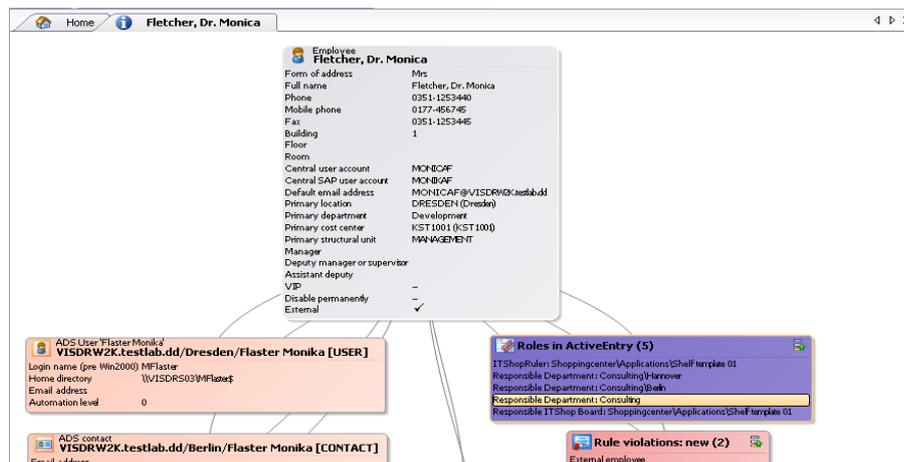


Figure 189: Example of Elements in an Overview Form

The display text of the menu item, the display text for the objects to be shown and the menu item icon are displayed in the header of a form element. Other data represents the object properties and values. There is a tooltip for each property showing a description for use. Some form element entries are highlighted in color when you click on them with the mouse. You can jump to the referenced object by clicking on the entry with the mouse.

If the form element is used for mapping lists, the items are displayed with their names. The number of items is shown in the form element header. There is also an icon in the form element header for showing and hiding items. There is no tooltip for list items.

Table 34: Form Element Icon

Icon	Meaning
	Show list items
	Hide list items

You can configure how the properties, color and position of the form elements of an object are displayed on the overview form using the menu item's layout information. The sections [Features of the Overview Form](#) and [Form Elements for Overview Forms](#) in the Configuration Manual explain how to create overview forms and configure menu items.

Input Fields

Input fields are used for editing strings, numbers and dates. A default context menu is available for input fields. Mandatory fields are marked with a triangle in front. Whether the field is mandatory or not, depends on the length of the input field (<Min. length>). Other Mandatory fields are already defined in the customizer.

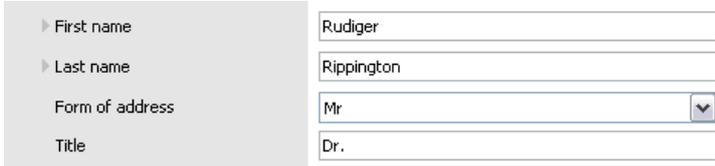


Figure 190: Mandatory Input Field

You can increase the size of input fields for inputting multiline text by using the key combination <Ctrl> + <Alt> + Enter.

Script Input Field

This input field is used when the input data needs to have a specified syntax (SQL, XML, VB.NET). The input field supports syntax highlighting. You can switch this input field into advanced edit mode. This mode allows additional actions.

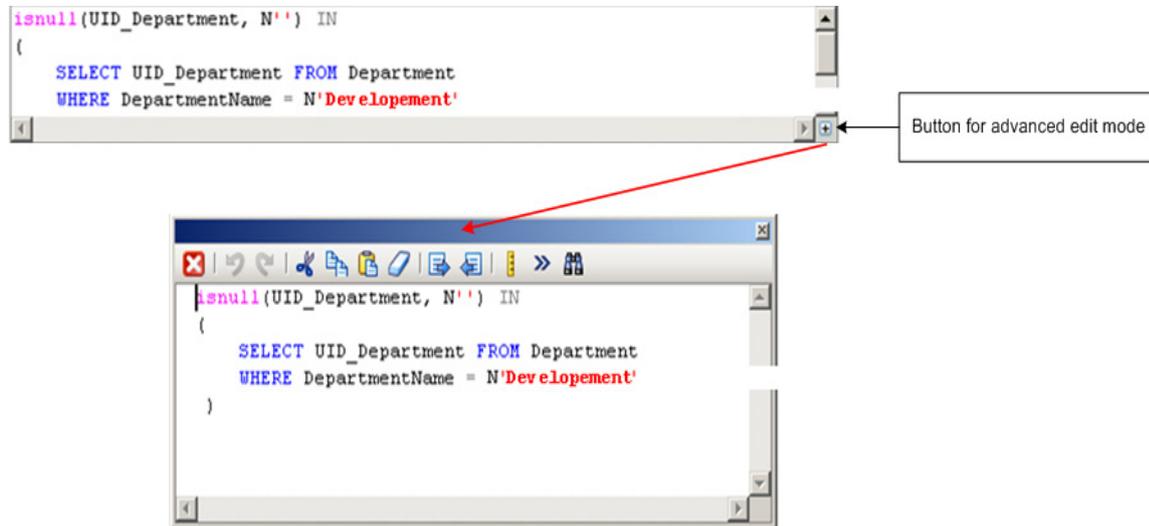


Figure 191: Entering a Database Query

Table 35: Meaning of Icons in Advanced Edit Mode

Icon	Meaning
	Quit advance edit mode
	Undo last change
	Redo last change
	Cut selected code
	Copy selected code into clipboard
	Insert code from clipboard
	Delete selected code
	Decrease insert
	Increase insert
	Show/hide line numbers

Table 35: Meaning of Icons in Advanced Edit Mode

Icon	Meaning
	Word wrap automatically
	Search within code

Permitted Value List Input Field

This input field is used if a list of permitted values can be specified for a column. The control element is displayed as a simple input field if no list is defined. If a list is defined, the control element is shown in the pop-up list. The control element is only available on the default form for columns predefined by Völcker Informatik or user defined columns (normally CustomProperty01 - CustomProperty02).



Figure 192: Input Field for List of Defined Values (with and without defined entries)

Multiple Value Properties Input Field

This input field is used when a column is suitable for several values. You can add and delete data with the buttons.

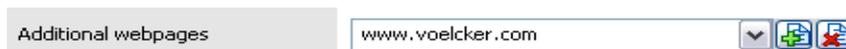


Figure 193: MVP Column Input Field

Language Dependent Input Field

This input field is used when a column is labeled as multi-language.

NOTE! Always enter the value in the default language.

Use the button next to the input field to open the translation dialog window. Enter the equivalent text in all the active languages. Use the button next to the translation text to delete a value. Use the <Save> button to save the input and <Cancel> to discard any changes. In both of these cases the dialog window is subsequently closed.

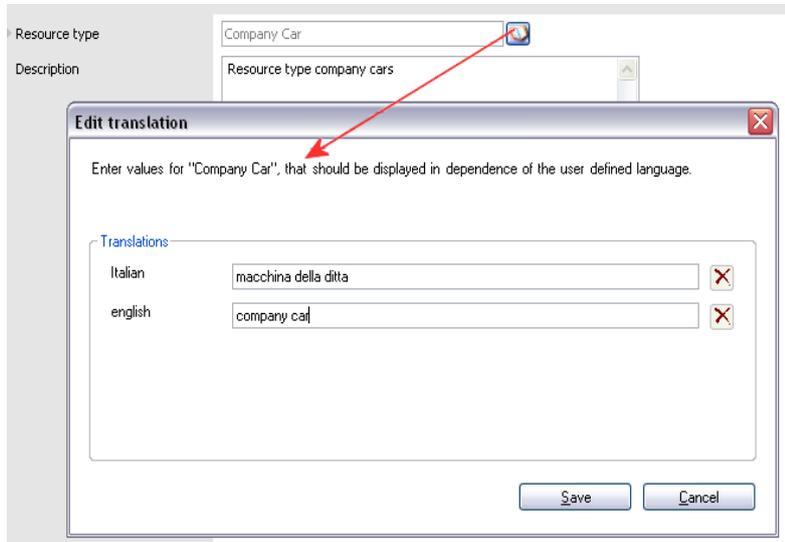


Figure 194: Multi-language Input Field

Language dependent text is only for displaying a value on a data form or in a list. Editable input fields still show the default language first.

Fakturaprodukt	Auto
Invoice product	Car

Figure 195: View Only Multi-language Field

▶ Fakturaprodukt	Auto	
▶ Invoice product	Auto	

Figure 196: Editable Multi-language Field

Pop-Up Menus

Pop-up menus or selection lists are used to choose an entry from a list. You can add values to some pop-up menus. An input label then appears in the menu.

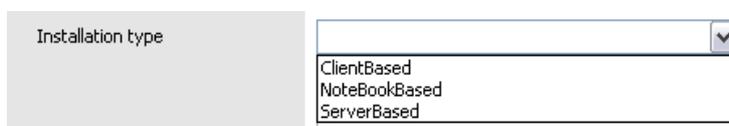


Figure 197: Pop-up Menu

Extended Pop-Up Menus

An extended pop-up menu points to elements in another database table. In order to represent hierarchical structures better, this pop-up not only supports selection from flat structures but also from hierarchical structures. To simplify object selection, you can resize the dialog window. This provides a better overview for selecting from large amounts of data.

You can also search for text strings within the selection data. Open the search dialog with the magnifying glass in the dialog window title bar. You can additionally specify case sensitive for a search. Use the <Search> button to jump to the first element that contains the search string. <F3> continues the search.

You enter the data in the input field by double-clicking on the icon in front of the object.

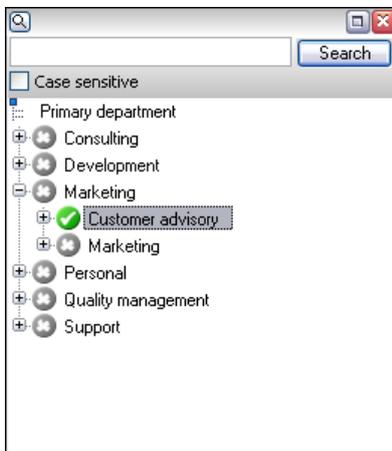


Figure 198: Extended Pop-Up

Check Boxes

Control boxes are used for specifying boolean values for an item. They are used to enable or disable an option separately from other options.



Figure 199: Check Box

Radio Buttons

Radio buttons are used to edit boolean values and numbers. They are used to mutually exclude options within a group.



Figure 200: Radio Buttons

Tables

A read-only table can be used to provide a overview of large amounts of information.

Profile name	Profile status (site)	Profile status
Framemaker - German - 7 - ...	Ready	EMPTY
Framemaker - German - 7.0...	Ready	EMPTY
Test - German - 1 - WXPd - ...	Ready	EMPTY
Word - German - 1.0 - NT40...	Ready	EMPTY

Figure 201: Table

You can insert or delete data sets into tables than can be edited. Select the table entry by holding down the mouse (or F2) and edit it.

Buttons

Use a button to start a defined action.

Action	MLS release (from FDS to MLS) ▼	Copy
--------	---------------------------------	------

Figure 202: Pop-up Menu with Adjacent Button

Tabs

Tabs are used for group control elements on separate pages of a form.

Rippington, Dr. Rudiger	
<ul style="list-style-type: none"> General Organizational Address Miscellaneous User defined 	
First name	Rudiger
Last name	Rippington
Form of address	Mr ▼

Figure 203: Form with Several Tabs

Setting the Date

You can either set the date by entering a value in the given field or you can choose the date from a control element provided for these means. Open the control element with the button next to the date field.

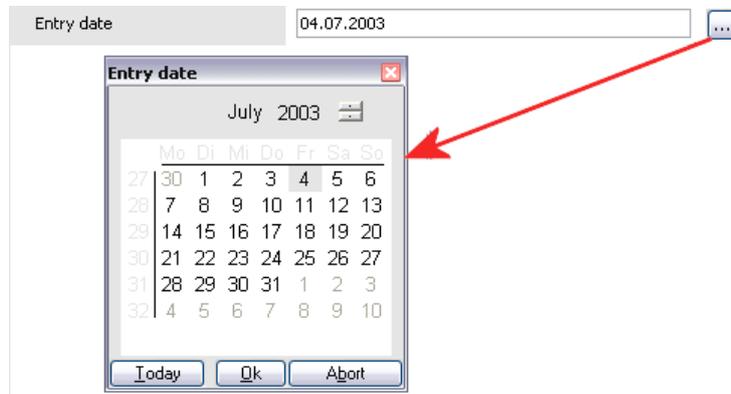


Figure 204: Control Element for Setting the Date

The description of the corresponding field is shown as title in the control's title bar. The control contains the following functions for selecting a date:

- Select a day
To do this select the day you want by clicking the mouse in the calendar. The selected day is highlighted.
- Select the month
It is possible to select a month by clicking on the name of the month in the list.
- Select a year
Use the arrow buttons to select the current year. You can insert the year using keys by marking the year in the control.

Press the <OK> button to accept the date. You can select today's date with the <Today> button. Use the <Abort> button to discard any changes. In all of these cases the window is subsequently closed.

Editing Login Times

This control element is used for configuring daily login times for a user account.

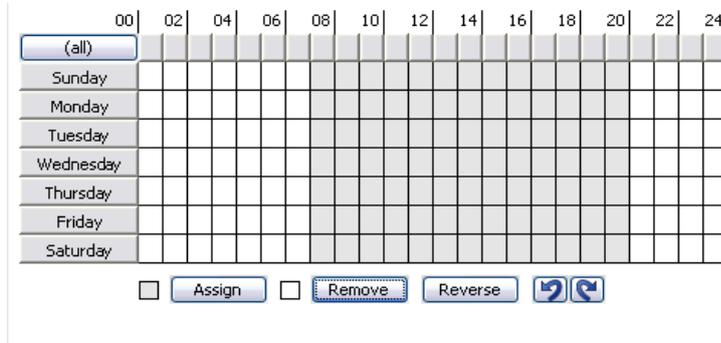


Figure 205: Control Element for Login Times

You can select a time period with the mouse or keys. Using the <Assign> and <Remove> buttons you can permit or deny logins at the specified times. Configured login times are highlighted in color respectively. Use the <Reverse> button to toggle the selected time period. The arrow keys can be used to undo or redo actions.

Displaying Object Relations

This control element displays the relations between ActiveEntry database objects (parent/child relation).

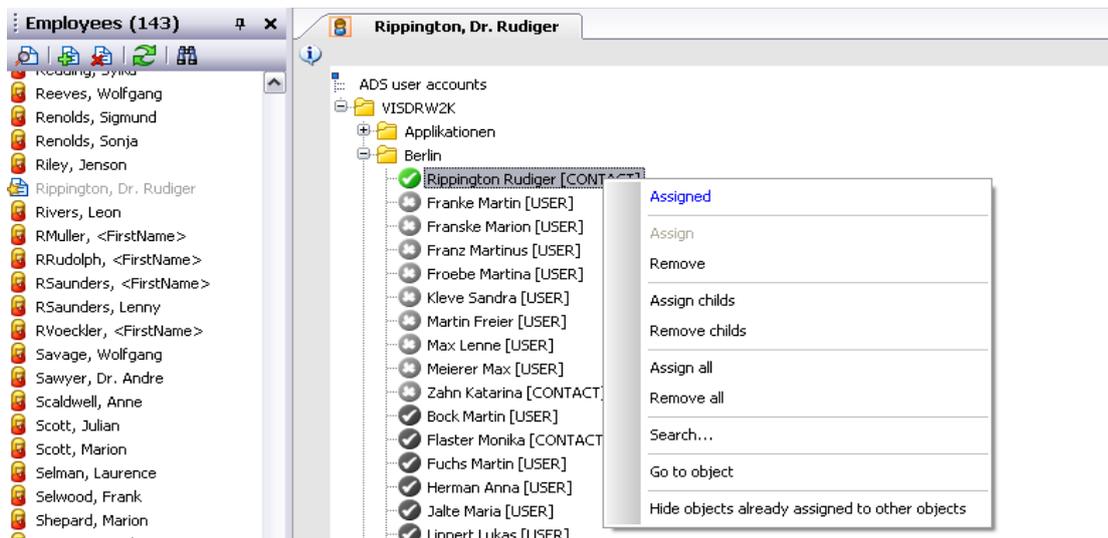


Figure 206: Graphical Representation of Assignments in a Control Element

Change the assignments with a double click on the icon or using the control's context menu.

Table 36: Meaning of Icons in the Control

Icon	Meaning
	The object is assigned to the selected root object.
	The object is not assigned to an object.
	The object is assigned to another object.

NOTE!

Assigned objects are displayed before non-assigned objects in the order for displaying object relations.

Table 37: Items in the Control's Context Menu

Item	Meaning
Assigned	The selected object is assigned to the root object.
Not assigned	The selected object is not assigned to an object.
Assigned to another object	The selected object is already assigned to another object.
Assign	Assign the selected object to the root object.
Remove	Remove the assignment of selected object to the base object.
Assign child objects	In a hierarchical structure, the selected object and its child objects are assigned to the root object.
Remove child objects	In a hierarchical structure, the selected object and its child objects are removed from the root object.
Assign all objects	All objects are assigned to the root objects.
Remove all assignments	All root object assignments are removed.
Search...	Opens a search dialog.
Go to object	Changes to selected object.
Go to assigned object	Change to root object that this object is assigned to.
Show objects already assigned to other objects	Shows objects that are assigned to another root object.
Hide objects already assigned to other objects	Hides objects that are assigned to another root object.

Displaying Assignments in a Member Tree

All direct and indirect assignments for a root object are displayed in a membership tree. Furthermore, the origin of the direct assignments is displayed. That means that the object was assigned to the root object through a dynamic role, for example, or by inheritance.

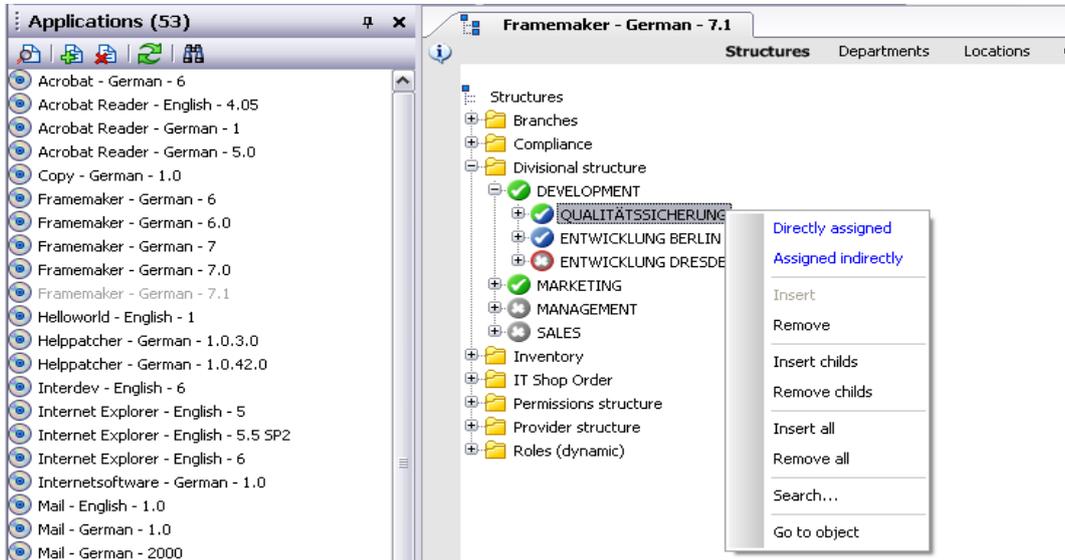


Figure 207: Displaying Assignments in a Membership Tree

You can change direct assignments by double-clicking on the icon or via the control's context menu.

Table 38: Meaning of Icons in the Membership Tree

Icon	Meaning
	The object is directly assigned to the selected root object.
	The object is indirectly assigned to the selected root object.
	The object is directly and indirectly assigned to the root object.
	The object is directly assigned to the selected to the root object. The assignment is not yet inserted into the corresponding *Total table.
	The object is not assigned to an object.
	Inheritance discontinued (only for assignment of objects in hierarchical structures)

NOTE!

Assigned objects are displayed before non-assigned objects in the membership tree.

Table 39: Items in the Membership Tree’s Context Menu

Menu item	Meaning
Not assigned	The selected object is not assigned to an root object.
Directly assigned	The object has been directly assigned to the root object.
Indirect assigned	Das Objekt wurde dem Basisobjekt indirekt durch Vererbung zugewiesen.
Indirectly via dynamic groups	These indirect assignments are made via the root object membership in dynamic groups.
Inheritance discontinued	This object has the property “Discontinue inheritance”.
Assign	Assign the selected object to the root object.
Remove	Remove the assignment of selected object to the base object.
Assign child objects	In a hierarchical structure, the selected object and its child objects are assigned to the root object.
Remove child objects	In a hierarchical structure, the selected object and its child objects are removed from the root object.
Assign all objects	All objects are assigned to the root objects.
Remove all assignments	All root object assignments are removed.
Search...	Opens a search dialog.
Go to object	Changes to selected object.
Extended properties	Changes to details form for the selected object where you can make further changes to the assignments. Prerequisite is that the selected objects is assigned and that the assignment has been saved.

You can use the <Search> function to search through the whole membership tree for occurrences of a particular text string. The search does not go through the tree in hierarchical structure order but by following the internal structure of the control. This may cause the search to appear to jump around arbitrarily within the tree. However, during the search all the data sets are taken into account and the object of the search will be found if it exists.

Entering Database Queries

Sometimes it is necessary to enter database queries. You can enter database queries directly as SQL queries or you can compose them using a wizard. Use the buttons <Edit SQL> and <Use wizard> to toggle between SQL and the wizard in the appropriate view.

Table 40: Meaning of the Icons

Icon	Meaning
	Direct input of database query as SQL query
	Wizard for entering a database query

Direct input of database queries takes place in a script input field. The functions described in the section [Script Input Field](#) are available for this.

Wizard for Entering Database Queries

The wizard helps you to formulate a condition (where clause) for database queries. The complete database query is composed internally. It always refers to the database table that is specified when you start the wizard.



Figure 208: Wizard for Entering Database Queries

There are predefined operators to make it easier to create conditions. The operators are underlined. Operators that are enclosed in brackets (<<operator>>) provide help for inputting the permitted values (e.g. select column, input date or random text input).

When you touch an operator with the cursor, the cursor changes shape into a hand icon. If an operator only has two permitted values you can swap between them with the mouse. When you click on an operator with more than two values a pop-up menu is opened with a list of permitted values. To select a value, click once on the entry you want and then confirm your choice using the button. Alternatively, you can simple double click on a value to select it. In both cases the pop-up is closed after the value has been selected.

You have the option to link several conditions together to create more complicated database queries. To do this, a new line with a logical link operator is displayed after each condition followed by another line for the next condition. Use the button in front of the condition if you want to delete it from the query.

Use the <Expert view> button to swap to the field for entering input directly in SQL syntax. To return to the simple input mode, use the button <Wizard view>. After you have entered all the conditions for the database query, use the <Fowards> button to take you forward to the preview. This shows you all the entries that satisfy the condition. If you continue again using the

<Forward> button the condition is shown as SQL query. Use the <Backward> button to return to the last view. The <Finished> button accepts the settings and the <Abort> button discard all the changes. In both cases the wizard is closed.

Rule Editor for Entering Rule Conditions

The Rule Editor helps you to formulate conditions for compliance rules. The complete database query is composed internally. Input of conditions is simplified by predefined condition types and limited permitted operators. You have the option to link several conditions together. There are two possible ways to define rule conditions:

- Simple definition
- Advanced mode

The Rule Editor is displayed in the respective design and with the respective functionality

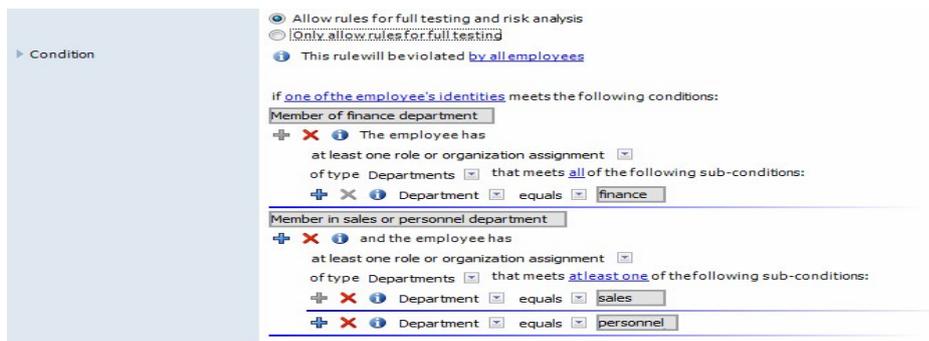


Figure 209: Rule Editor in Simple Definition Mode

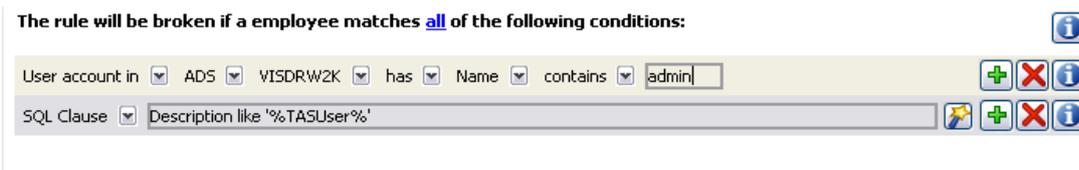


Figure 210: Rule Editor in Advanced mode

Table 41: Meaning of Icons in the Rule Editor

Icon	Meaning
	Add another partial condition or another rule block. A new line is displayed for entering the condition.
	Delete the partial condition or rule block. The line is removed.
	Opens the preview window. All affected employee objects are shown.

See section [Basics for Using the Rule Editor](#) in the Identity Management Manual for more information on the functionality of the Rule Editor.

Overview of All Assignments

The report „Overview of all assignments“ can be displayed for certain objects. The reports show all the employees have the selected base object is assigned to them. Both directly assigned objects as well as inherited objects the employees are taken into account. The report shows which employees are members of which roles of a role class.

Table 42: Meaning of Icons in the Report Toolbar

Icon	Meaning
	Show the legend with the meaning of the report control elements
	Saves the current report view as a graphic.
	Selects the role classe used to generate the report.

A simple mouse click on a role or the role class in the toolbar is all that is needed to display the report for this role.

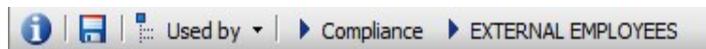


Figure 211: Toolbar for Report „Overview of all assignments“

The meaning of the report control elements is explained in a separate legend.

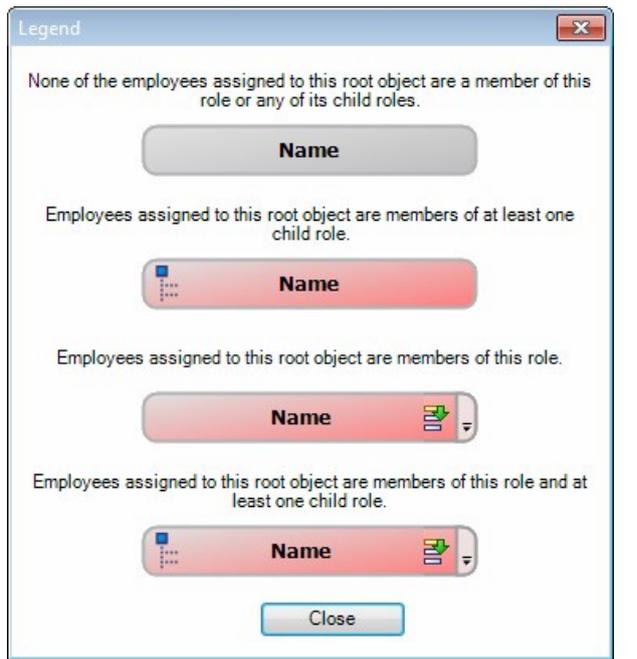


Figure 212: Legend for the Report „Overview of all assignments“

A simple mouse click on the control element is all that is needed to display all the employees that the base object is assigned to and that are members of the selected role. By double-clicking on the control element, all the child roles are displayed for the selected role.

7.4.3 Multiple Object Edit

It is possible to edit more than one object at the same time in the ActiveEntry Manager and the ActiveEntry Identity Manager. Select the required entries in the result list with <shift or ctrl> + select and open master data form.

The form tab of the main data form shows the number of the selected objects and the object type. Input fields that have different values are marked by an icon in front of the field. The values that are entered in the fields and changes that are saved are stored for all of the objects.

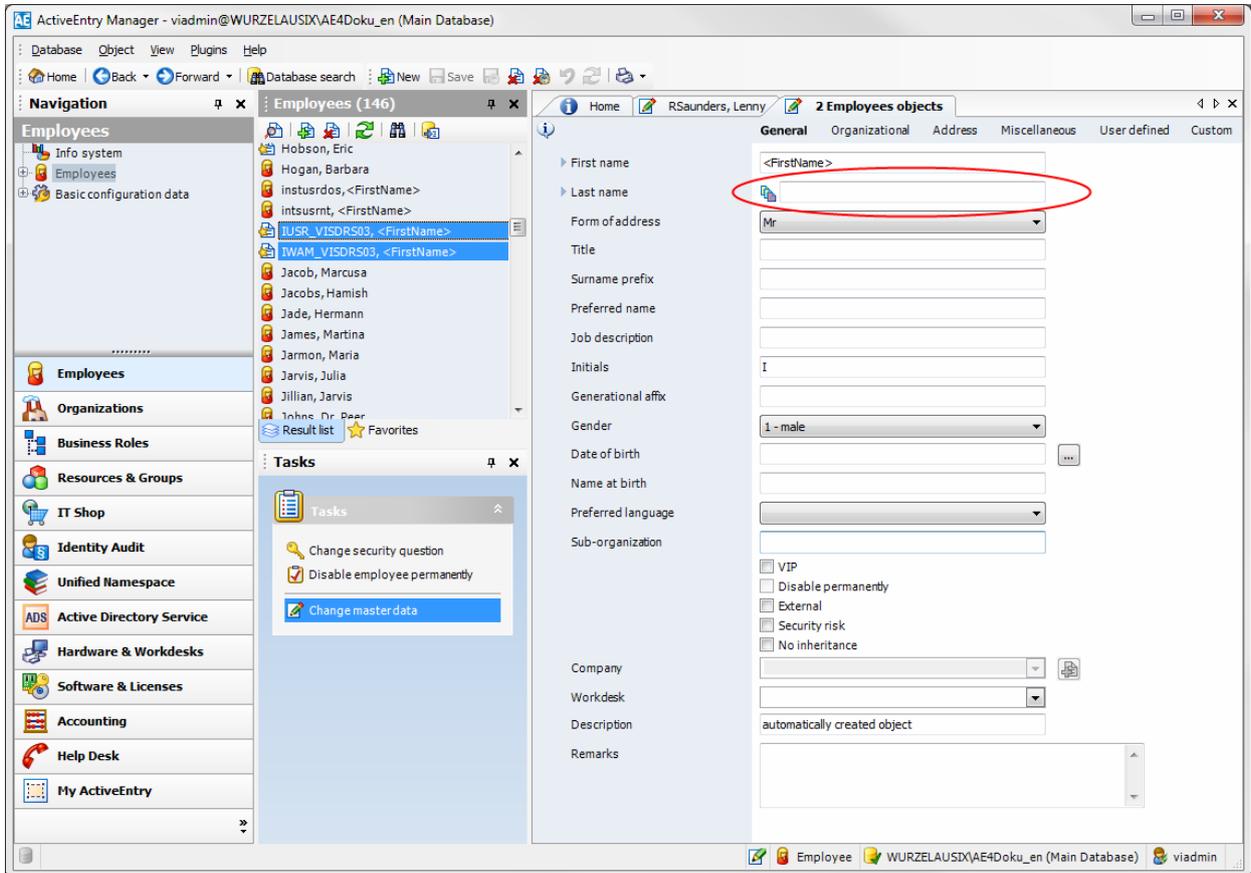


Figure 213: Master Data Form with Multi-Editing

7.4.4 Reusing Value Templates

In ActiveEntry value templates are used to specify a default value in a column or to take a value from one column and use it to define a value in another. Value templates can take effect within an object as well as between objects. Value templates take effect without taking into account the state of permissions. This means that the objects affected by the templates in use are all filled even if they are not displayed on the current form in the ActiveEntry Manager or the ActiveEntry Identity Manager.

The ActiveEntry Manager and the ActiveEntry Identity Manager support explicit updating the column values of objects mapped using value templates. The menu item <Object>\<Reuse template> allows you to reapply templates to the current object. This menu item is only available on the object's master data form.

NOTE! When the value templates are recalculated for an object, a large number of dependent objects may also change therefore generating processes.

7.5 Functions in the Tasks View

If an object is selected in the result list, the tasks to be performed for that object and the available reports are shown in the task view. Select the item in the respective list to execute a task or to open a report. You can see a more detailed description about a task or a report in the tooltip

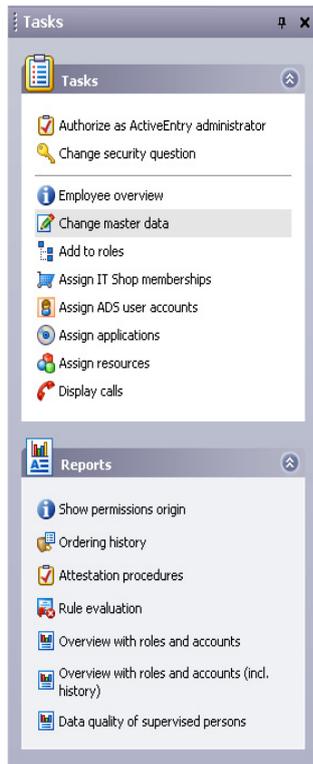


Figure 214: Task View

If tasks are linked to predecessor relevant configuration parameters, the names of the configuration parameters may be shown after the task. To do this, set the option "show additional navigation information" in the program settings.

7.6 Functions in the Favorites List

In the favorites list you can save links to the most frequently used objects. Use the menu item <Object>\<Add to favorites> to add the link to the favorites list. Use <Object>\<Remove from favorites> or the toolbar to delete the item again.

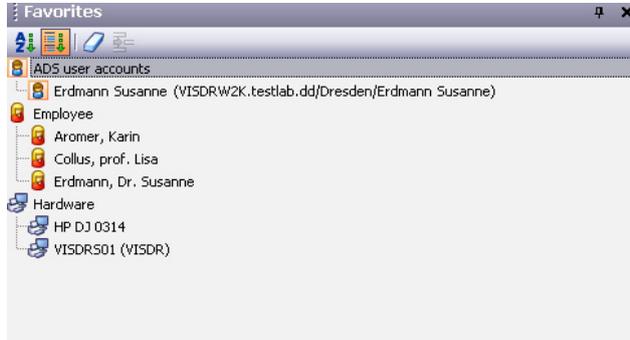


Figure 215: Favorites View

The favorites list features its own toolbar and a context menu.



Figure 216: Favorites Toolbar

Table 43: Meaning of the Entries in the Special Toolbar

Icon	Meaning
	Favorites sorted alphabetically
	Favorites sorted by object type
	Delete complete favorites list.
	Delete selected favorites

Table 44: Entries in Favorite List Context Menu

Item in Context Menu	Meaning
Find...	Searches for objects within the favorites list.
Tasks	All the tasks available for this object are shown in a drop-down menu. When you select one of the tasks the corresponding form is opened.
Properties...	Shows more properties for the current object. This menu item is only available in advanced mode.

7.7 Logging Error Messages

Errors that have occurred are logged via:

- Error messages in a message window (see section [Error Message Window](#))
- Error messages in the error log view
- The program’s error log file

7.7.1 Error Message Window

Error messages are shown in a separate window. Configure the amount of information to be displayed using the options in the error message window. The button <Send as mail> creates a new email message in the default mail program and copies the error text to it. Shut the window with the <OK> button.

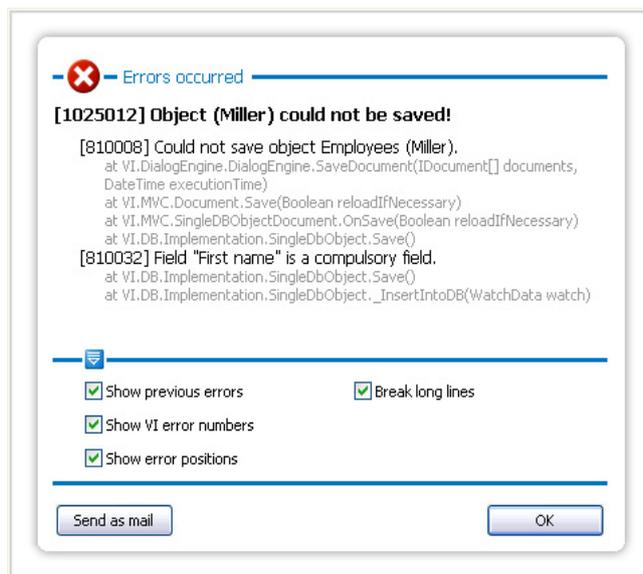


Figure 217: Example Error Message Window

7.7.2 Displaying Error and System Logs

Table 45: Configuration Parametes for Recording System Log

Configuration Parameter	Effect
Common\JournalLifeTime	Use this configuration parameter to specify the maximum amount of time (in days) that a system protocol entry can be stored in the database. Older entries are deleted from the database.

The <Error log> view shows you the program’s error log and the system log.

- Program's error log

The program's error log displays all warnings and error messages that have occurred since the program started up. The error log is reinitialized when the program is restarted. An icon in the program's status bar indicates new messages in the error log. Open the error log by double-clicking on the icon.
- System log

The system log displays information, warnings and error messages from different ActiveEntry components, such as DBScheduler, DBMigrator or ActiveEntry Service.

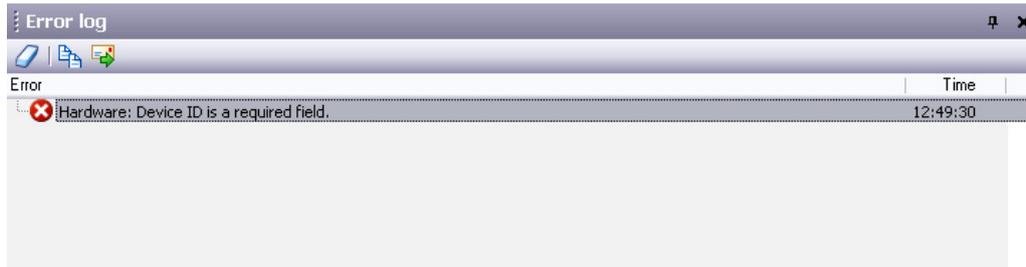


Figure 218: Logging Errors in the Error Log

The view has its own toolbar. Icons are shown or hidden depending on the type of log.

Table 46: Meaning of the Entries in the Special Toolbar

Icon	Meaning
	Display program's error log.
	Display the system log.
	Copy the selected entries into the clipboard
	Filter type of message to display (information, warning, error).
	Delete all entries from the error log.
	System log: refresh view.
Sender: DBScheduler	System log: select components that send the message.
	Send error as email

Table 47: Icons Used by Error Logging

Icon	Meaning
	Information is written to the error/system log.

Table 47: Icons Used by Error Logging

Icon	Meaning
	An error message has be entered in the error log.
	A warning has been entered in the error log.

7.7.3 Error Log File

All the errors that occur, independent of restarting the program, are written into an error log file. The log file is stored in the program directory. The log file is overwritten on a weekly basis.

The log files have the following naming convention:

```
ActiveEntry Manager_<weekday>_yyyy_MM_dd.log
```

where:

```
yyyy - year, 4-figures
MM - month, 2-figures
dd - day, 2-figures
```

Example:

```
ActiveEntry Manager_Monday_2004_02_09.log
```

7.8 Evaluating the Process Monitoring Information

ActiveEntry offers the possibility to log the history of changes to objects and their properties. A number of different mechanisms can be used within ActiveEntry to follow changes. Please refer to the chapter [Tracking Changes with Process Monitoring](#) in the Configuration Handbook.

The information generated by monitoring processes and process steps, process information for direct database actions as well as data changes are logged in a graphical form to the program's process view.

7.8.1 Standard Functions in the Process View

Table 48: Configuration Parameters for Recording Process Information

Configuration Parameter	Effect
Common\ProcessState	Records changes when the parameter is set.

The process view is only available when the configuration parameter "Common\ProcessState" is activated and you are authorized to use this program function. You get to the process view via the menu item <View>\<Process information>. The process view is subdivided into a process information form in which the recorded information from the monitoring process is represented, and a log to display the recorded data changes.

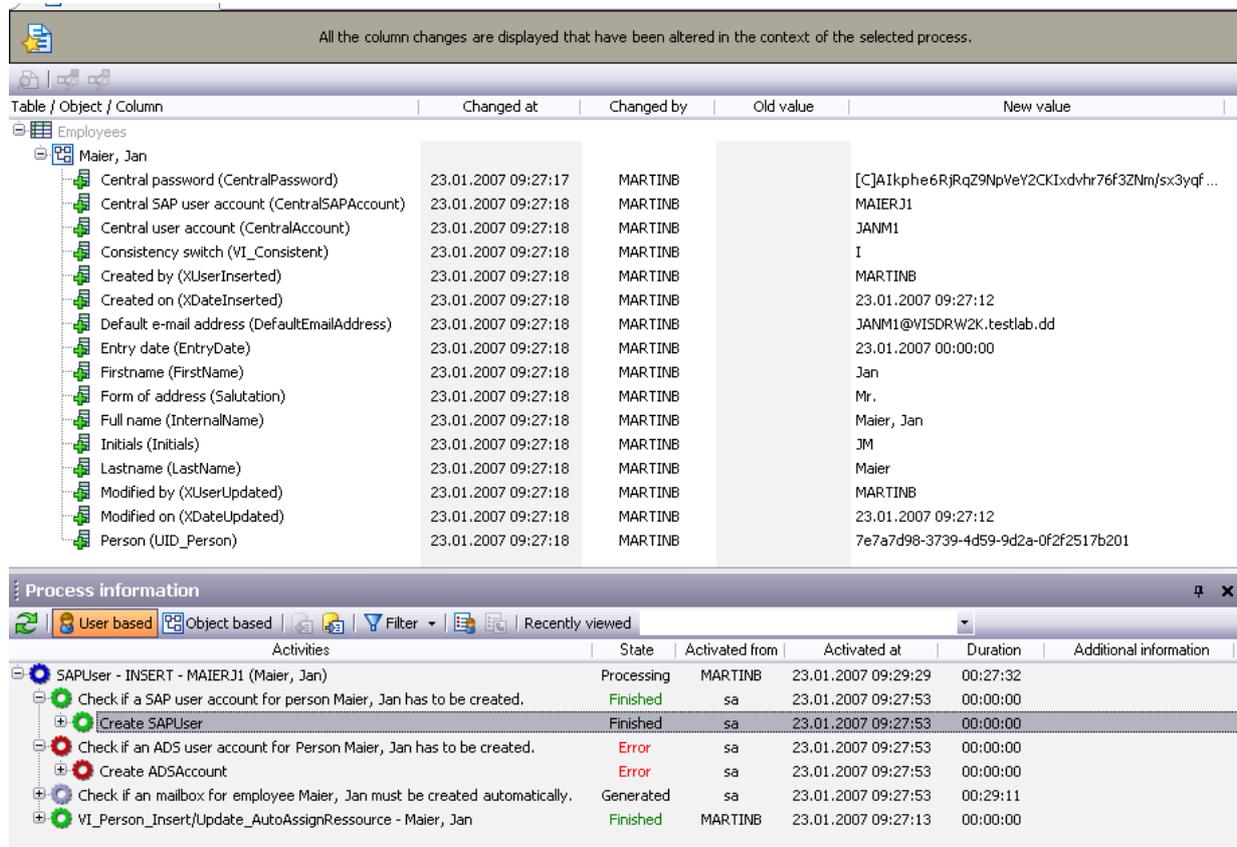


Figure 219: Process Information Form (below) and the Log (above)

Process Information Form Functions

The process information form features its own toolbar and a separate context menu. The menu entries are hidden or shown depending on the selected entry.



Figure 220: Toolbar for Viewing the Process Information

Table 49: Meaning of the Entries in the Special Toolbar

Icon	Meaning
	Reload process information.
	Show process information for the logged on user (user related process information).
	Shows process information for the selected object. (object related process information).
	Shows process dependent objects.
	Show substitution processes.
	Filter process information by status.
	Shows data changes made by the logged on user (user related changes).
	Shows data changes for the selected object. (object related changes).

Table 50: Entries in the Process Information View Context Menu

Entry in Context Menu	Meaning
Search...	Find objects within the process information view.
Add to favorites	Add selected object to the favorites.
Remove from favorites	Delete selected object from favorites
Show	Shows available form for this object. Switch to object if required.
Object type:<root object>	Root object of the process that was triggered
Show log	Show the log of the data changes for the selected process (process related changes).
Properties...	Shows more properties for the selected object. This entry is only available in advanced mode.

Functions in the Log

The process log is displayed in the document view. It is opened over the context menu or the respective buttons in the process information toolbar. The log has its own toolbar to display the data changes.



Figure 221: Process Log Toolbar

Table 51: Meaning of the Entries in the Special Toolbar

Icon	Meaning
	The chosen object is shown in the document view.
	Switches to the originally referenced (old) object and shows it in the window.
	Switches to the newly referenced object and displays it in the window.

Configuring the Process Display

Process information from process monitoring, for single activities and for details and progress steps, can be specified from different configuration possibilities for displaying information. Configuration is accessed over the menu <Database>\<Settings> on the tab <Features>.

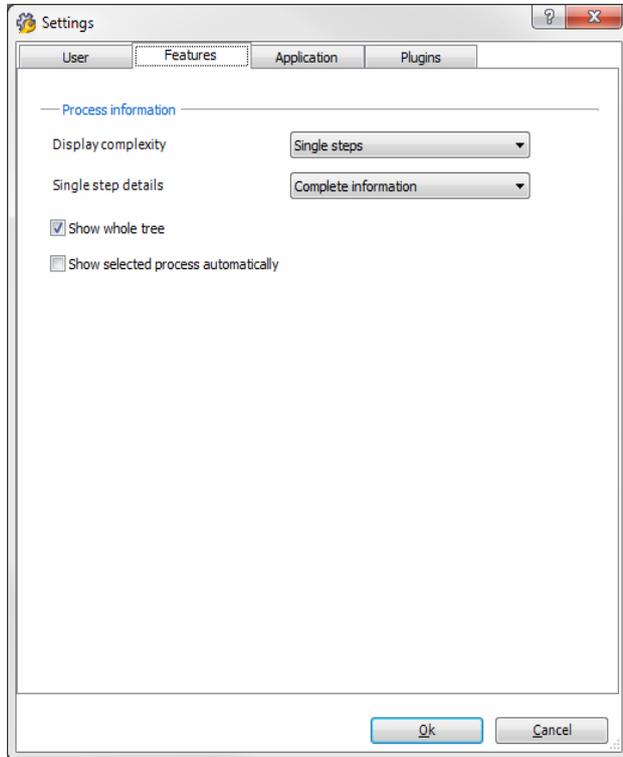


Figure 222: Configuration of the Process Information

You can decide on which level the process should be displayed (activities, details, individual steps). The information view for individual steps can be limited by defining the depth of the detail to be shown. The image of the process information can be represented in hierarchical tree form or in list form.

Table 52: Configuration of the Process Information View

Configuration	Allocation	Meaning
Display Range	Activities	Shows activity information (top hierachy level).
	Details	Shows information about activities and the related details.
	Single steps	Shows information about activities, details and individual steps at the select depth.

Table 52: Configuration of the Process Information View

Configuration	Allocation	Meaning
Single step details	Basic information	Shows single steps with a detail depth of "Basic information".
	Enhanced information	Shows single steps with a detail depth of "Enhanced information".
	Full information	Shows single steps with a detail depth of "Basic information", "Enhanced information" und "Full information" (technical view).
Show whole tree	active	Opens the whole tree hierachy automatically on loading the process information view.
	not active	Does not open the whole tree hierachy on loading the process information view.
Show selected processes automatically	active	Opens the whole tree hierachy automatically when a process is selected.
	not active	Does not open the whole tree hierachy when a process is selected.

7.8.2 Displaying Process Information

On the process information form in the process view, the logged in user can get an overview of events in the system that he or she has triggered and their results. In this way, information for the the entire process as well as for individual steps of a process are sent.

There are several different views implemented for displaying the process information. On the one hand, processes are shown that have been set up by the currently logged in system user (user related) and on the other hand, processes are displayed that have been set up for the currently selected object (object related).

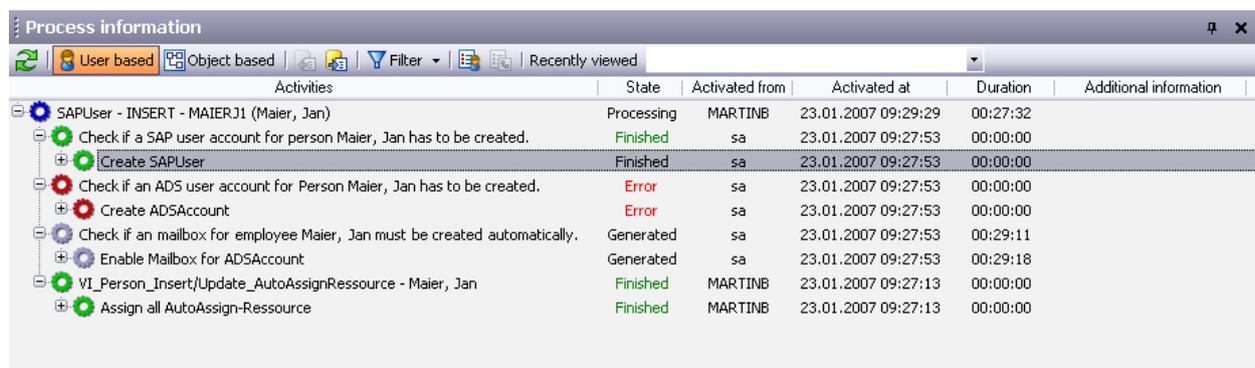


Figure 223: Displaying Processes

The following information is shown for a process.

Table 53: Process Information History

Information	Meaning
Activities	Display text for information for the process.
Status	Status of the process.
Started by	User that started the process.
Started on	Date on which the process was started.
Duration	Process duration.
Additional information	Additional information about the status, such as retries of single steps or start time for deferred step.
Process ID	Unique ID (GenProcID). Changes that can be traced back to the same origin, are given the same GenProcID and grouped together in this way. You can enable a process ID entry and copy it via the context menu.

The following icons are used to describe the state of the process.

Table 54: Meaning of Icons for Process State

Icon	Meaning
	Process successfully completed (state "Finished")
	The processing is in progress (state "Active")
	An error occurred during processing (state "Error")
	Status of process progress (state "waiting", "deferred", "frozen", "not complete")
	Process dependent on selected process
	Previous alternative process
	Next alternative process

7.8.3 Displaying the Change History

Table 55: Configuration Parameter for Recording Process Information

Configuration Parameter	Effect
Common\ProcessState	Records changes when the parameter is set
Common\ProcessState\PropertyLog	Records changes in individual values to the database when the parameter is set.

The data changes shown in the process information view are displayed in log form in the document view. The log is only available when the configuration parameter "Common\ProcessState\PropertyLog" is set and the logged on user has at the least, viewing rights for the tables "Dialogwatch*" and "DialogProcess*".

There are a number of different ways of representing the data changes. All the data changes that are carried out within a process are shown (process related) as well as all data changes that are carried out by the logged in user (user related). In addition, all data changes are represented that have been recorded for the currently selected object (object related).

The screenshot displays two windows from a software application. The top window shows a table of column changes for the object 'Maier, Jan'. The bottom window shows a 'Process information' view with a list of activities.

Table / Object / Column	Changed at	Changed by	Old value	New value
Maier, Jan				
Central password (CentralPassword)	23.01.2007 09:27:17	MARTINB		[C]A1kphe6RJqZ9NpVeY2CKIxdvhr76f3ZNm/sx3yqf ...
Central SAP user account (CentralSAPAccount)	23.01.2007 09:27:18	MARTINB		MAIERJ1
Central user account (CentralAccount)	23.01.2007 09:27:18	MARTINB		JANM1
Consistency switch (VI_Consistent)	23.01.2007 09:27:18	MARTINB		I
Created by (XUserInserted)	23.01.2007 09:27:18	MARTINB		MARTINB
Created on (XDateInserted)	23.01.2007 09:27:18	MARTINB		23.01.2007 09:27:12
Default e-mail address (DefaultEmailAddress)	23.01.2007 09:27:18	MARTINB		JANM1@VISDRW2K.testlab.dd
Entry date (EntryDate)	23.01.2007 09:27:18	MARTINB		23.01.2007 00:00:00
Firstname (FirstName)	23.01.2007 09:27:18	MARTINB		Jan
Form of address (Salutation)	23.01.2007 09:27:18	MARTINB		Mr.
Full name (InternalName)	23.01.2007 09:27:18	MARTINB		Maier, Jan
Initials (Initials)	23.01.2007 09:27:18	MARTINB		JM
Lastname (LastName)	23.01.2007 09:27:18	MARTINB		Maier
Modified by (XUserUpdated)	23.01.2007 09:27:18	MARTINB		MARTINB
Modified on (XDateUpdated)	23.01.2007 09:27:18	MARTINB		23.01.2007 09:27:12
Person (UID_Person)	23.01.2007 09:27:18	MARTINB		7e7a7d98-3739-4d59-9d2a-0f2f2517b201

Activities	State	Activated from	Activated at	Duration	Additional information
SAPUser - INSERT - MAIERJ1 (Maier, Jan)	Processing	MARTINB	23.01.2007 09:29:29	00:27:32	
Check if a SAP user account for person Maier, Jan has to be created.	Finished	sa	23.01.2007 09:27:53	00:00:00	
Create SAPUser	Finished	sa	23.01.2007 09:27:53	00:00:00	
Check if an ADS user account for Person Maier, Jan has to be created.	Error	sa	23.01.2007 09:27:53	00:00:00	
Create ADSAccount	Error	sa	23.01.2007 09:27:53	00:00:00	
Check if an mailbox for employee Maier, Jan must be created automatically.	Generated	sa	23.01.2007 09:27:53	00:29:11	
VI_Person_Insert/Update_AutoAssignResource - Maier, Jan	Finished	MARTINB	23.01.2007 09:27:13	00:00:00	

Figure 224: Logging the data changes withing a process

The following information is displayed in the data change log:

Table 56: Data Change Information

Information	Meaning
Table	Display text for the table that the data set belongs to. This input enables the objects to be grouped.
Object	Object effected by the change.
Column	Changed column.
Date changed	Date on which the change was initiated.
Changed by	The user that made the changes.
Old value	Column value before changes.
New Value	Column value after changes.

7.9 Analysis of Historical Data in TimeTrace

The TimeTrace function is used to trace changes to an object back to any point in the past. The TimeTrace function incorporates data changes from the ActiveEntry database as well as the records stored in the history database into the evaluation. This makes it possible to specify which employee had what permissions at which point in time. Historical data can be copied to the current object thus restoring its state before the changes were made.

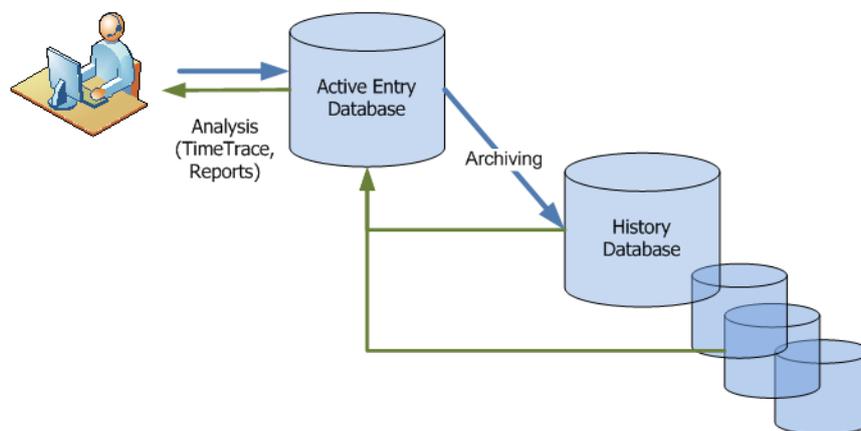


Figure 225: Historical Data Analysis

Prerequisite for using the TimeTrace function is that changes are logged within process monitoring. Read the section [Recording Data Changes](#) in the Configuration Manual for more information.

Changes that are saved in the ActiveEntry database can be immediately included in the evaluation. The history database must be declared in the ActiveEntry database if archive data are going to be integrated in the TimeTrace. For more information, read the section [TimeTrace Databases](#) in the Configuration manual.

7.9.1 Functions in the TimeTrace View

Table 57: Configuration Parameter for Recording Process Information

Configuration Parameter	Effect
Common\ProcessState	If this parameter is set, changes are recorded and the information is displayed in the process view.
Common\ProcessState\Property-Log	If this configuration parameter is set, all changes that are made to single objects are recorded in the database.

The TimeTrace view is only available when the configuration parameteress "Common\ProcessState" and "Common\ProcessState\PropertyLog" are set and you have sufficient permissions for the program function. Navigate to the view via <View>\<TimeTrace>.

The view has its own toolbar.

Table 58: Meaning of Entries in the Special Toolbar

Icon	Meaning
	Enables/Disables change history view
	Shows object properties after this change.
	Shows object properties before this change.
	Finds changes in the specified time interval.

7.9.2 Displaying Change Information

To display the change information for an object, proceed as follows:

- Select the object in the ActiveEntry Manager or ActiveEntry Identity Manager result list.
- Enable change history for this object using the corresponding icon in the TimeTrace view.
- Use the <Time interval> filter in the TimeTrace toolbar to specify the time period that the data should be loaded for. The changes are found in the ActiveEntry database and connected history databases.

The time and date of all changes loaded for this time period are displayed under the timeline in the overview. Click with the mouse on a time stamp to zoom in on that part of the timeline. Each change mark is displayed with time and date. If several change time stamps are close together, the selection is made via a context menu from which you can choose the time stamp you want. Each time stamp has a tooltip which shows you which data has changed.

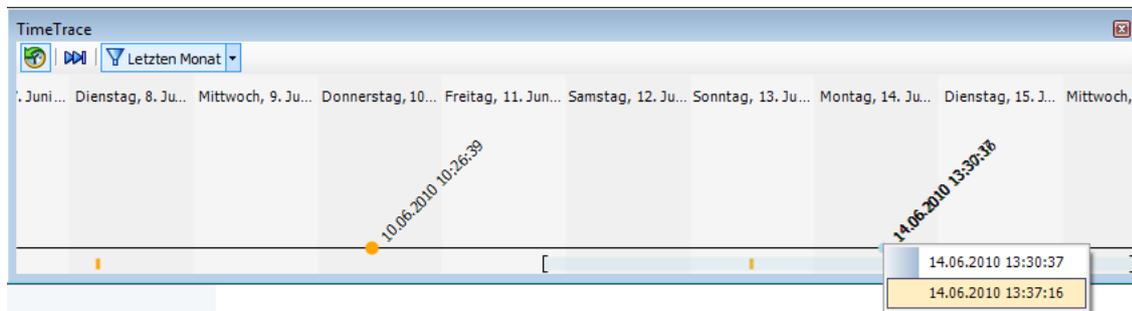


Figure 226: TimeTrace View Showing Change Time Stamps via a Context Menu

When you select a change time stamp, the master data form for that object is opened in the program's document view. Use the toolbar in the TimeTrace view to select whether the object properties should be shown before or after the change in the master data form. If a property shows a historical value, it is shown by a special icon. A tooltip displays the actual value of the property.

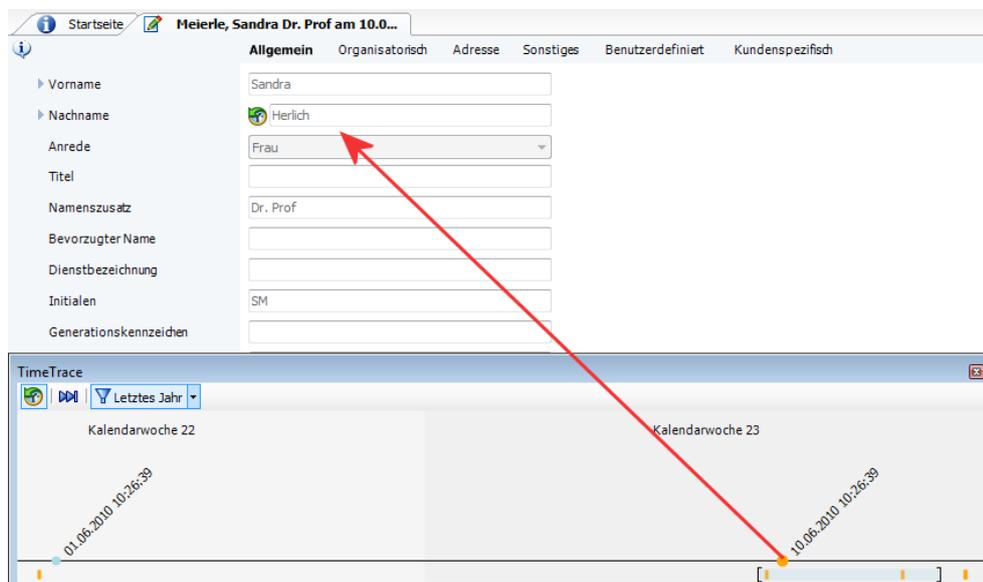


Figure 227: Stammdatenformular mit historischen Daten

You can copy historical data for the current object and then restore the object's state from before the change. To do this, click on the icon in front of a property to open a dialog window..

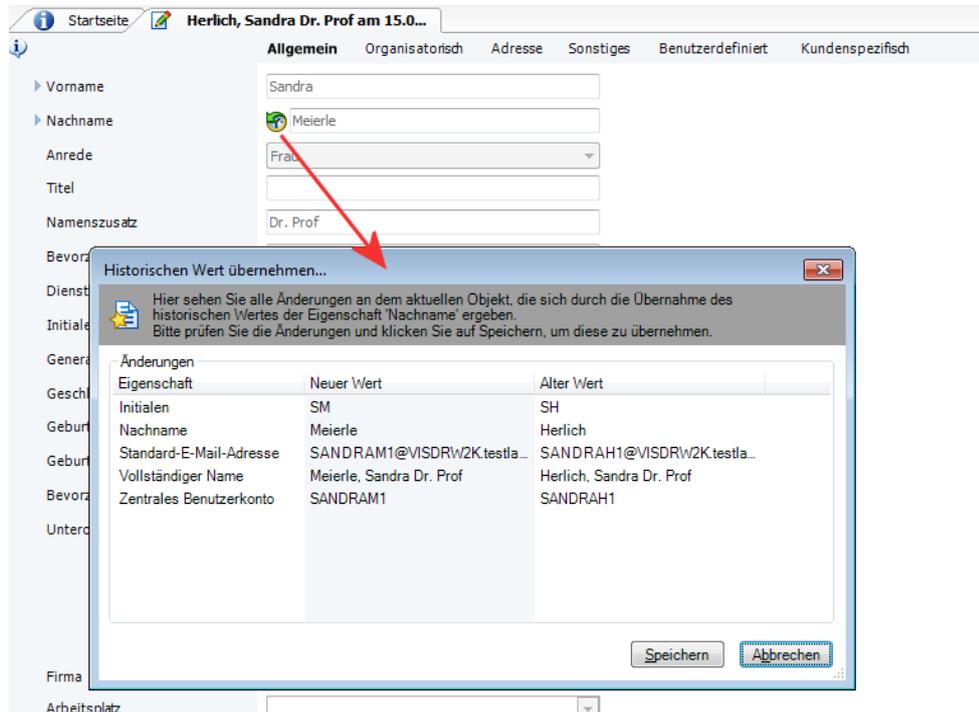


Figure 228: Transferring Historical Data

The following information is shown.

Table 59:

Information	Meaning
Property	These properties will be changed when the historical value is transferred. The changes take place either directly or via templates.
New value	Value of the property once the historical value has been saved.
Old value	Shows the current value of the property. This value is overwritten when the historical value is saved. Angezeigt wird der aktuelle Wert der Eigenschaft. Dieser Wert wird überschrieben, wenn der historische Wert gespeichert wird.

Use the <Save> button to accept the changes. Use the <Cancel> button to discard the changes. In both cases the dialog window is closed.

7.10 Working in Simulation Mode

The ActiveEntry Manager and the ActiveEntry Identity Manager can be switched into a simulation mode. This mode allows you to analyze the effects of changes made on large amounts of data before actually carrying them out.

NOTE!

Simulation mode should only be used in exceptional cases! During simulation, the objects are locked for other users. It can lead to restrictions in working with the administrative tools. In certain circumstances, ActiveEntry Service cannot process further jobs in the simulation phase. Depending on the breadth of the changes, the whole ActiveEntry network can come to a stand still during a simulation.

Simulation mode is only available when you are authorized to use this functionality. Activate the simulation mode from the menu item `The simulation mode is activated over the menu <Database>\<Start simulation>`. The active simulation mode is indicated by an icon in the program's status bar which is displayed in red.

Table 60: Icon representing Active Simulation Mode in the Statusbar

Icon	Meaning
	The program is in simulation mode.

To exit the simulation mode and return the program to a normal work mode, select the menu item `<Database>\<Stop simulation>`. After the simulation has finished you can save the changes directly or delay them. For this you need to use the appropriate menu item in the program. To avoid blocking the whole system for a long period, a timer is implemented that exits the simulation mode after 5 minutes without saving the data.

Saving changed data in simulation mode records the following information:

- Computational tasks for the DBscheduler that result from the changes.
- Trigger modifications that result from the change.
- Processes that are generated due to the change.
- Objects that are effected by the change.
- Recalculation of Identity Audit rules that result from the change
To do this, enable the plugins "Identity audit Ssmulation" and "Identity audit simulation summary" in the program settings.

7.10.1 Simulation Data Overview

When the simulation is ended the changes that have been recorded are loaded and displayed in log form in the document view of the program. You can either execute the changes directly after you close the log (and after accepting the security prompt) or you can discard the simulation data. Here you can get an overview of which action has been triggered by the changes that have taken place.

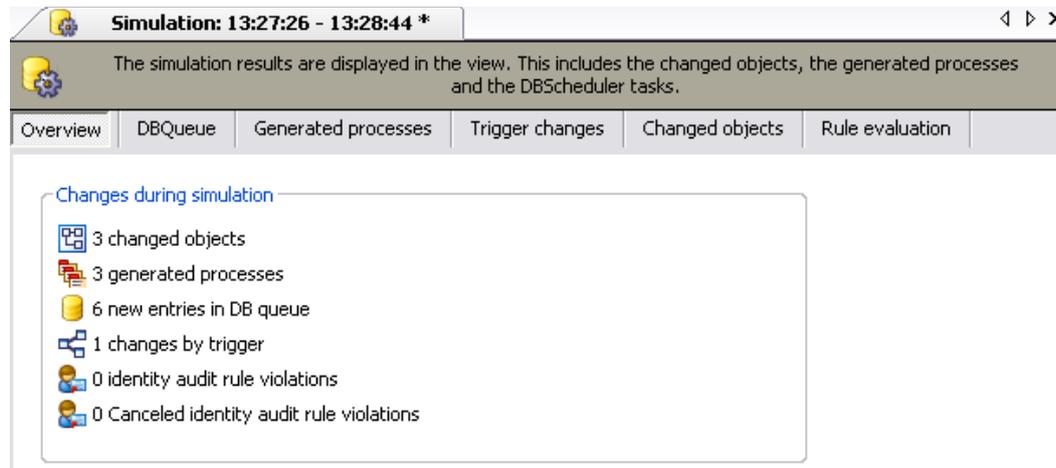


Figure 229: Simulation Data Overview

Analyzing the DBQueue

Calculation tasks for the DBScheduler resulting from simulation changes are shown in this view.

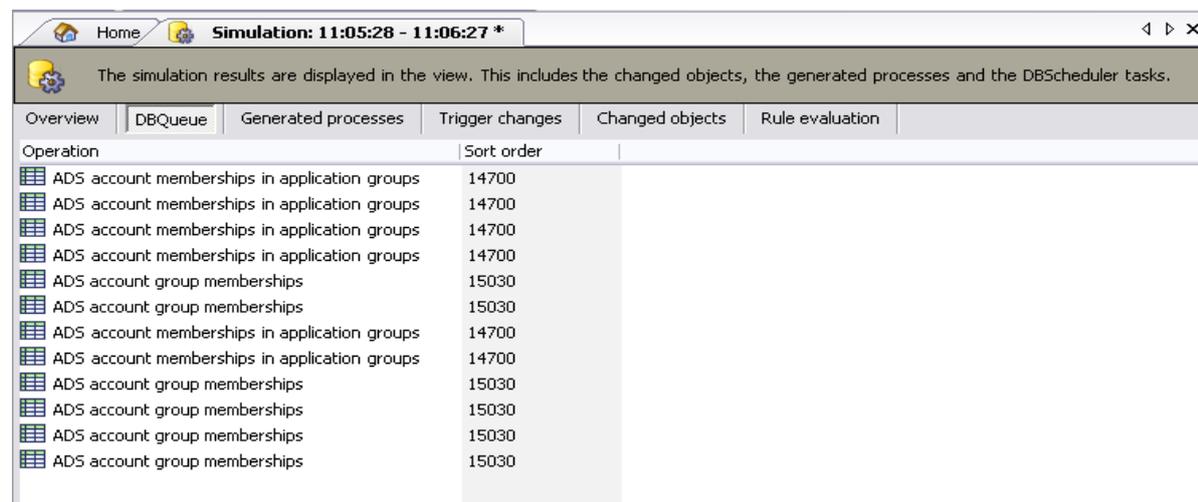


Figure 230: Logging the Calculation Tasks for the DB Scheduler

The following information is displayed in the DBQueue log. You can hide and show additional information using the context menu.

Table 61: DBScheduler Calculation Task Information

Information	Meaning
Operation	Calculation tasks to be carried out.
Sort order	Order in which the calculations task to be carried out, are sorted.

Analyzing Generated Processes

Processes and process steps are shown that are generated based on changes made during the simulation. In addition, individual properties of processes and process steps are shown with concrete values.

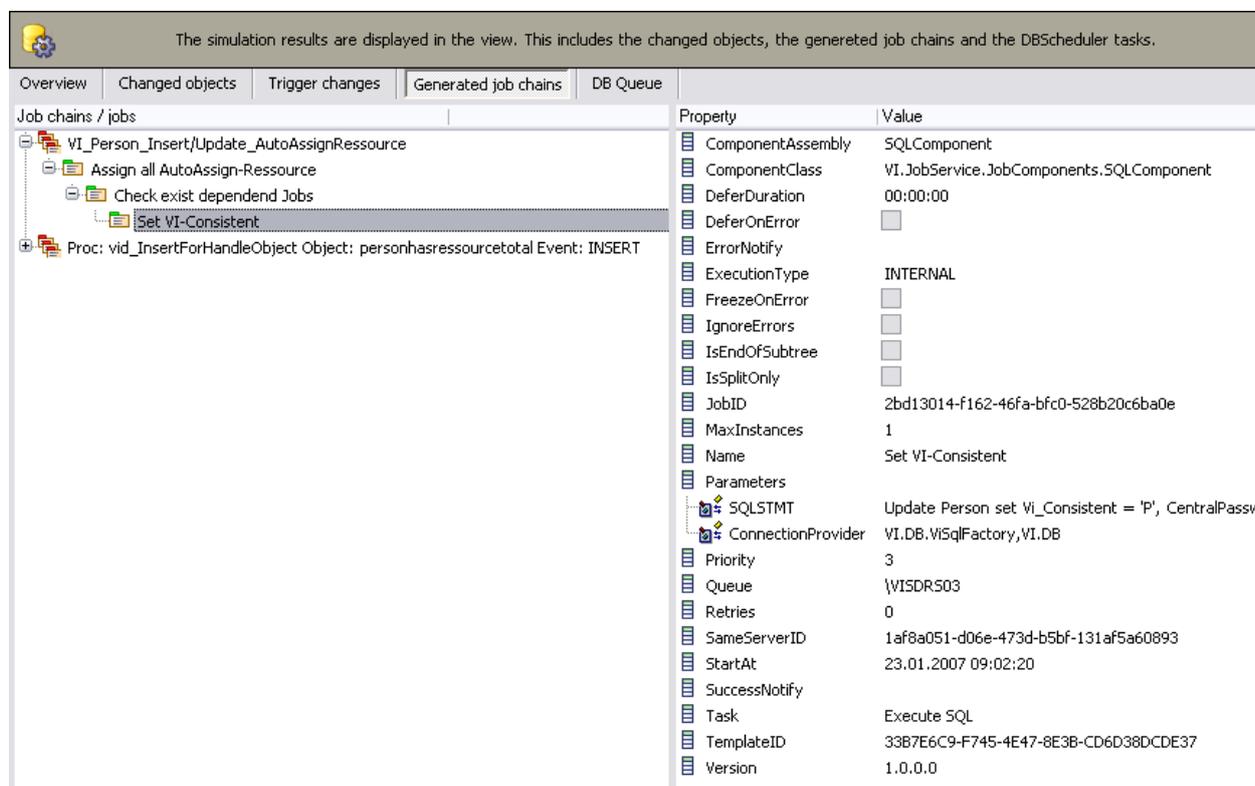


Figure 231: Logging Generated Processes

The following information is displayed for processes and process steps.

Table 62: Generated Process Information

Information	Meaning
Process	Name of the generated process.
Process step	Name of the generated process step.

Table 62: Generated Process Information

Information	Meaning
Property	Property of the process or process step.
Value	Attribute's value.

Analyzing Trigger Changes

All changes to objects that have been setup by triggering during the simulation are shown here.



Figure 232: Logging Trigger Changes

The following information is displayed.

Table 63: Changed Object Information

Information	Meaning
Table	Display text for the table that the data set belongs to. This input is used for grouping objects.
Object	Object that is affected by the changes.
Column	Column to be changed.
Old value	Column value before the change.
New value	Column value after the change.

Analyzing Modified Objects

All objects and their properties that are affected by the changes during simulation are displayed.

Table / Object / Column	Old value	New value	Change Level
Employees (Person)			
Raabe, Sonja			
City		Berlin	1
Primary department		Berlin	0
Employee number		4242	0
Remote access permitted	<input type="checkbox"/>	<input checked="" type="checkbox"/>	0
Preferred language		German	0
Primary structure unit		BERLIN	0
Primary location		BERLIN (Berlin)	0
Notebook user	<input type="checkbox"/>	<input checked="" type="checkbox"/>	0
Available field no. 01		1	0

Figure 233: Logging Changed Objects

The following information is shown for all modified objects.

Table 64: Information for Changed Objects

Information	Meaning
Table	Display text for the table that the data set belongs to. This input enables the objects to be grouped.
Object	Object effected by the change.
Column	Changed column.
Old Value	Column value before changes.
New Value	Column value after changes.

Analyzing the Rules

All rules affected by the changes are recalculated during simulation. Simulation and evaluation of rule violations only takes place if the relevant plugins are enabled in the program settings. New rule violations and violations that have been revoked as a result of recalculating the rules are displayed.

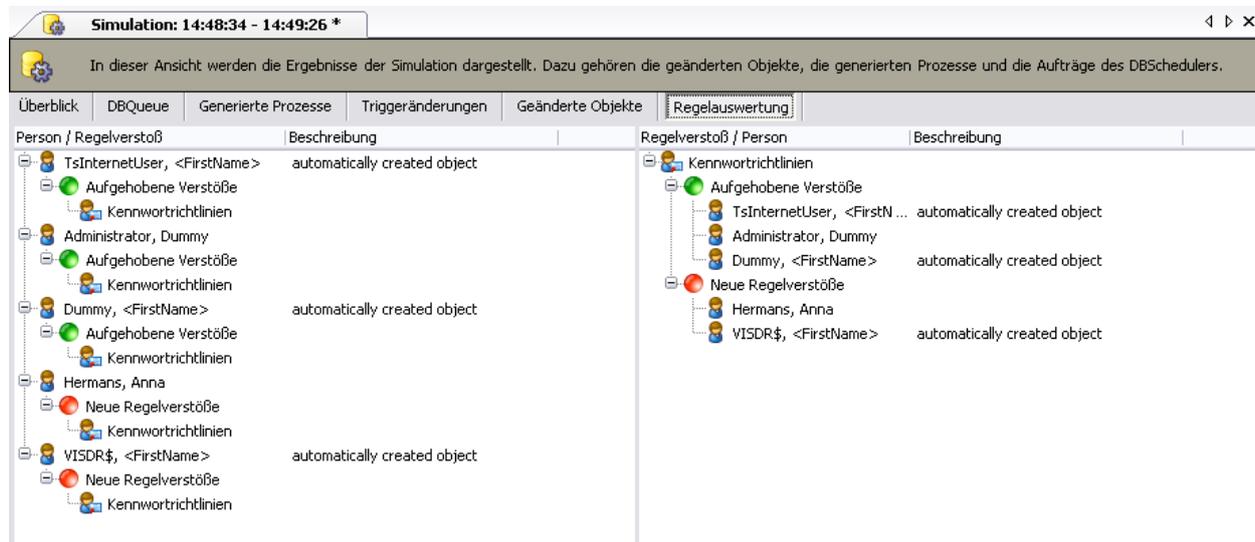


Figure 234: Logging Rule Violations

The following information is displayed.

Table 65: Rule Violation Information

Information	Meaning
Employee	Employee that has violated the rule or has stopped violating the rule.
Rule violation	Type of change (new violation or revoked violation) and the affected rule.
Description	Description of the rule violation

7.11 Planing the Execution Times of Operations

You can specify a time schedule for executing individual operations in ActiveEntry Manager and im ActiveEntry Identity Manager if you do not want to execute them immediately. Schedules may be planned for different operations. You can execute custom tasks and events as well as default operations such as adding, changing and deleting objects. The DBScheduler checks whether the planned operations exist and executes them at the specified point in time.

7.11.1 Specifying the Execution Time Schedule

Table 66: Configuration Parameter for Deferred Operation Execution

Configuration Parameter	Meaning
Common\DeferredOperation	Preprocessor relevant configuration parameter for recording deferred operations. If the parameter is enabled, tasks can be set up for deferred processing of operations. The database has to be recompiled after changes have been made to the parameter.

Planing of execution times is only possible if the configuration parameter "Common\Deferred-Operation" is enabled. The database has to recompiled after changing this parameter. Refer to the sections [Compiling the ActiveEntry Database](#) and [Preprocessor Relevant Configuration Parameters](#) in the Configuration Manual for more information.

Use the menu item <Object>\<Define execution time...>. To schedule an object for deletion, use the menu item <Object>\<Schedule delete...>. A dialog window opens where you can set the date and time of execution.

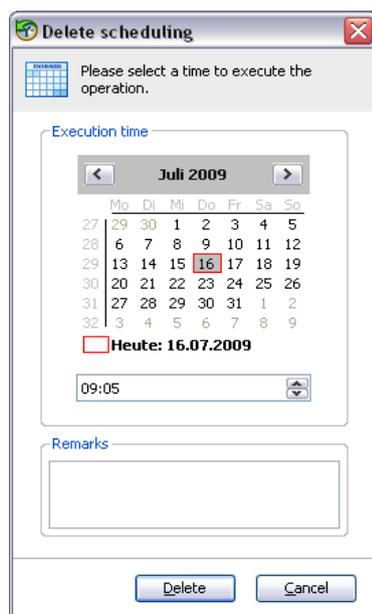


Figure 235: Planning an Execution Schedule

The name for the operation to be executed is displayed in the title bar of the dialog window. The dialog window contains the following functions for scheduling execution:

- Select a day
To do this select the day you want by clicking the mouse in the calendar. The selected day is highlighted.
- Select the month
It is possible to select a month by clicking on the name of the month in the list or on the arrow buttons.
- Select a year
Click the mouse on the year to display the arrow buttons. Use the arrow buttons to select the year or use the up and down arrows keys on your keyboard.
- Select today's date
Use the <Today> entry to select today's date. This date is marked with a red frame in the calendar.
- Select a time
Mark the hours or minutes and use the arrow keys or click with the mouse on the arrow buttons to change the time.

Use the <Save> button to save the selected time of execution. Use the <Cancel> button to discard the changes. In both cases, the dialog window is closed.

7.11.2 Displaying Scheduled Operations

Use the menu item <Database>\<Show deferred operations> to open a form, which displays all deferred operations and their scheduled execution times.

Table / Object	Operation	Execution time	Remarks
Departments	Insert Object	17.07.2009 09:11	Insert of " in table Department.
Catering	Insert Object	17.07.2009 09:13	Insert of 'Catering' in table Depa
Employees	Delete object	19.07.2009 09:05	Delete object 'Dummy, Adele' fr

Figure 236: Overview of Planned Operations

The following information is displayed.

Table 67: Data Modification Information

Information	Meaning
Table	Display text for the table that the data set belongs to. This input is used for grouping objects.
Object	Object that is affected by the changes.
Operation	Operation to be executed for this object.
Time of Execution	Time at which operation should be executed.

A double mouse click on an entry loads the associated object and displays it. If the time of execution has elapsed and an error occurred, the corresponding entry is marked in red.

The formular has its own tool bar and a separate context menu. The menu items are enabled or disabled depending on which entry is selected.

Table 68: Meaning of Entries in the Special Tool Bar

Icon	Meaning
	Change to selected object
	Löscht das ausgewählte Objekt
	Reload the data
	Filter information

Use the predefined filter form to limit the amount of information that is displayed by using a filter condition. You can filter by 'pending operation' or by 'state of pending operation'. Use the arrow button next to the <Filter view> icon in the tool bar to open the predefined filter menu. You can combine filters. The filter is applied to the list until you reset it. To do this, click on <Filter view> in the tool bar or use the menu item <Show all>.

Table 69: Predefined Filters

Filter		Meaning
State	Pending operations	Pending operations are hidden/shown.
	Obselete operations	Operations that are beyond the time of execution are shown/hidden.
Operation	Add object	Shows/hides all entries for the operation "Add object".
	Change object	Shows/hides all entries for the operation "Change object".
	Delete object	Shows/hides all entries for the operation "Delete object".
	Generate event	Shows/hides all entries for the operation "Generate event".
	Call method	Shows/hides all entries for the operation "Call method".
Alles anzeigen		Shows all planned operations.

7.12 Implementing User Defined Filters for Database Searches

User defined filters allow the logged in user to display specific data according to the user's own selection criterion. Filters should always relate to objects with respect to the object definition. The main components of a user defined filter are:

- A search in the object's display value
- A self-formulated search condition (Where clause)
- A full-text search with various options

You have the option to set up adhoc filters and permanent filters. Adhoc filters are used for one-off searches. These filters are not saved and are applied to the data immediately. We recommend setting up a permanent filter if you want to reuse it more frequently. Permanent filters are saved in the user configuration and therefore are always available for use. Permanent filters are shown in the category <My ActiveEntry> and can be edited and run from there.

7.12.1 Setting Up User Defined Filters

Use the menu item <View>\<Database search> to open the input dialog for setting up user defined filters. This is shown as an extension to the navigation view.

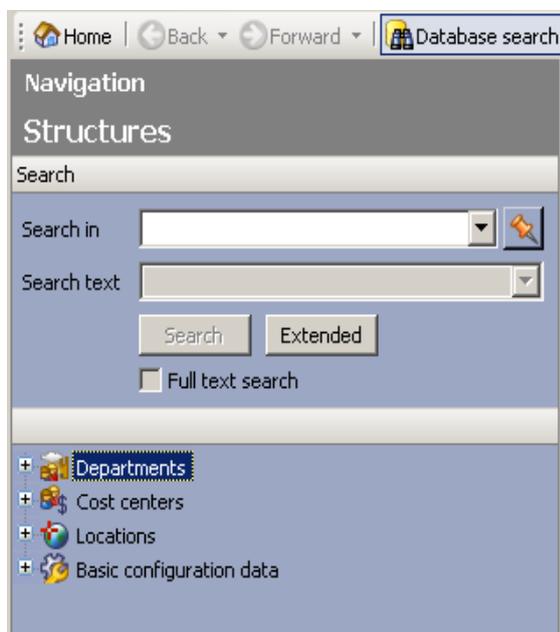


Figure 237: Database Search View

Here you have the option to start a simple adhoc search. The object definitions that are available in the selected navigation category are shown in the pop-up menu <Search in>. You may use the wildcard '*' (star). The string is searched for within the display values of the selected object definition. The search results are displayed in the result list.

TIP



This button allows you to specify the current search as default for further searches.

The <Advanced> button allows you to open an input dialog in advanced edit mode. This allows you to enter a search condition by defining a valid database query WHERE clause instead of a display text search or to use a full-text search. Apart from running a search request as an adhoc search, you can also save a permanent user defined filter in advanced edit mode.

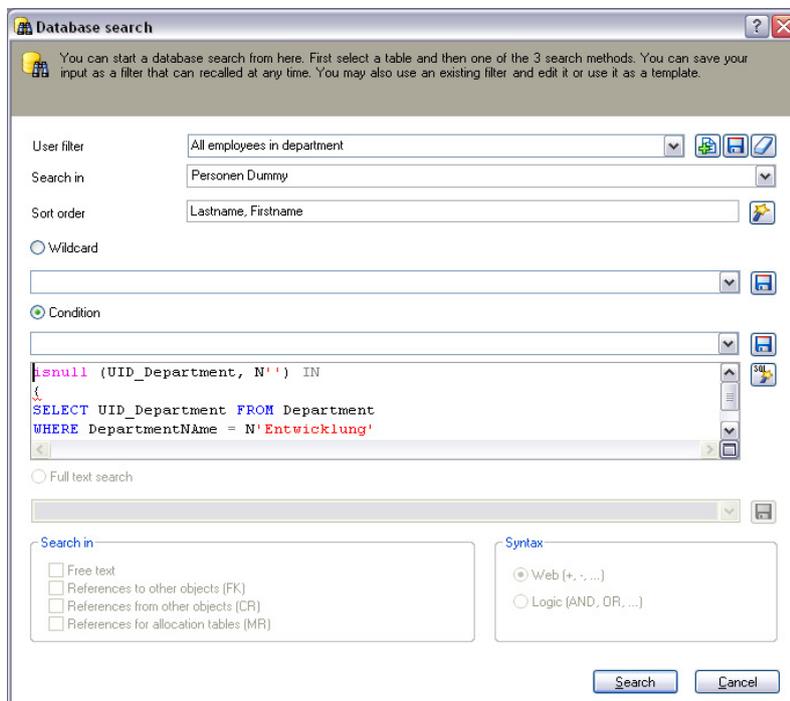


Figure 238: Advanced Database Search

Use the <Search> button to start the search. The results are displayed in the result list. Use the <Cancel> button to discard any changes. In both cases the dialog box is closed.

You can use the following data in advanced mode to model your search:

- User defined filter
Enter a name for the search so it can be saved and used as a permanent filter. Permanent user defined filters are displayed with these names in the category <My ActiveEntry> where they can be selected for editing or execution.

- **Search in**
The object definitions that are available in the selected navigation category that the database search was started in are shown in this pop-up menu.
- **Sort order**
Enter the properties (column) that the search results should be ordered by. Use the button next to the input field to show all properties available according to the object definition. Each identifier that you click on creates an entry in the input field.
- **Search using wildcard, a condition or full-text search**
Chose the option you want to specify how the search should be carried out. Each of these options is described in more detail in the following.

Searching with Wildcards

Use the <Wildcard> option to search for a string in the display values of the selected object definitions. Use the button next to the input field to save the search string. The search string is now available in the pop-up menu for further searches. You can use the wildcard '*' in your search string.

Example:

```
abc* - searches all entries with a display value starting with "abc"
*abc - searches all entries with a display value ending in "abc"
abc - searches all entries with a display value containing "abc"
```

Searching with Conditions

Use the <Condition> option to search all enties according to the selected object definition that satisfy the given condition. Create the condition with valid database query WHERE clauses. You can enter the SQL request directly or you can use a wizard to create it. Open the wizard with the button next to the input field. You can save the condition with the button next to the pop-up menu. The condition is therefore available to you over the pop-up menu for further searches.

Searching with Full-Text

This search method is only available if the configuration parameter "Common\Fulltext" is enabled and the database is set up for full-text search. Read the section [Setting Up an ActiveEntry Database for Full-Text Search](#) for more information. The <Full-text search> option starts a full-text search for the string entered in search field. You can save the search string with the button next to the pop-up menu. The sstring is therefore available to you over the pop-up menu for further searches. To search for a string in all the entries that match the selected objects, select the option <Free text> in the <Search in> options box.

Example:

abc - search all entries that have the exact string "abc"
 "abc*" -search all entries that have the properties beginning with "abc".

If you are searching for more than one string, you need to specify the method in the <syntax> options box. You can chose between a logical operator or a web search expression. The following operators are permitted:

Table 70: Syntax for Linking Search Strings

Web Seach Expression	Logical Expression	Meaning
+	AND	String must exist
-	AND NOT	String may not exist
~	NEAR	String infront is similar to string behind.
String without operator prefix	OR	String must exist
"expression"		Brackets an expression with spaces and special characters

For example, in order to find an employee with the name "Frank" (irrespective of whether is it a first name or surname) you simply search with the <free text> option. In certain cases, however, it makes sense to search within an object's foreign relations. If you want to find all the employees at the "Dresden" location that are assigned the application "Microsoft Office 2003"; for example, the object "employee" will not be found because the location is a foreign reference and the application is saved as an assignment. In order to get a result anyway in such cases, you can include child relations, foreign references and assignment in the object search. To do this select the options in the <Search in> options box.

7.13 Exporting Data

You may use the ActiveEntry Manager and the ActiveEntry Identity Manager to export data from the application data model. Export is only available when you are authorized to use this functionality. The program supports data export to files in CSV format that you can subsequently edit with Microsoft Office Excel. You can export all the data from a base table including all tables that can be referenced from this table by foreign key relations.

7.13.1 Creating an Export

Open the export form from the menu item <Database>\<Export data>. The form is divided in to several criterion. Specify the export criterion <Column selection>, <Column sort order> and <Condition> boxes. A preview of the export is created in the <Export data> box. Then the export is run.

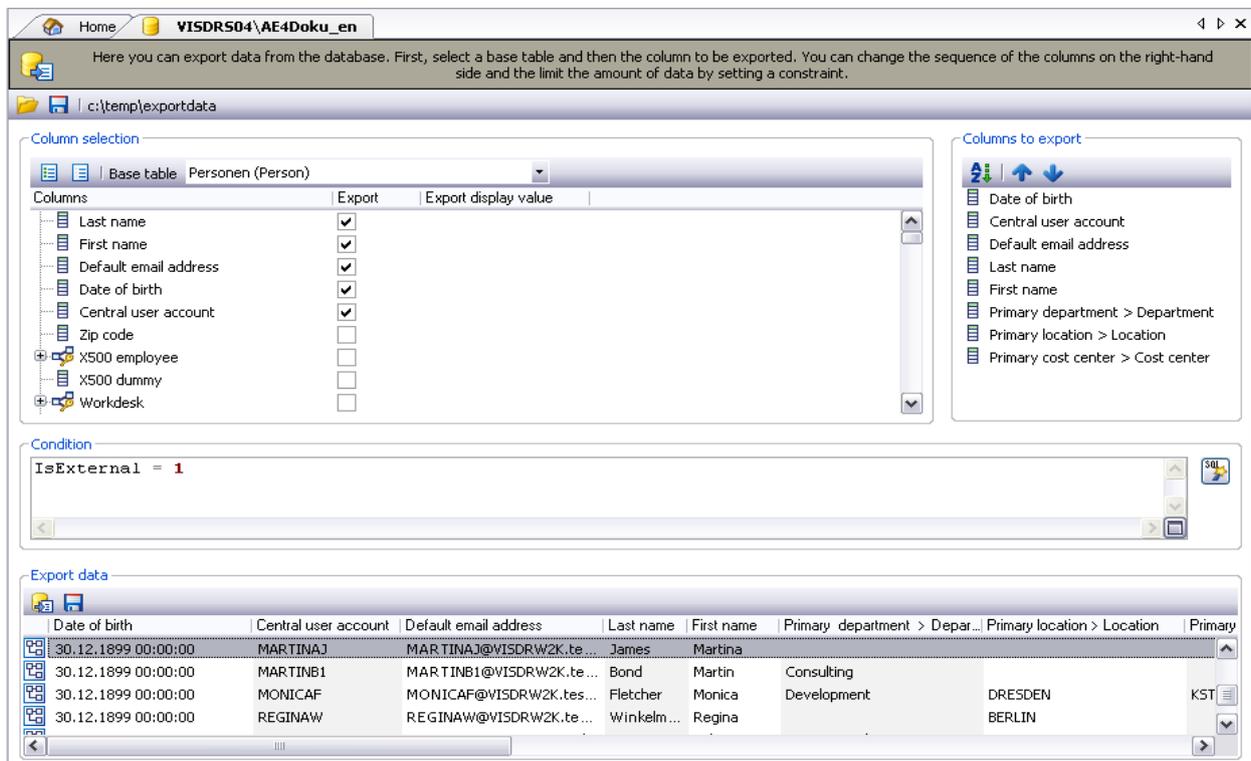


Figure 239: Data Export Form

Selecting a Base Table and Columns for Export

Decide on the database table and the database columns required in the <Column selection> box. Select the export database table in the <Base table> pop-up menu. After selecting the database table, the exportable database columns are loaded and shown in table form. Click on a icon in the table header of the result list to sort by the selected column.

The columns in the selected base table and all tables with their columns that can be available via FK relations are displayed in the <column> column. Enable the associated check box in the column <Export> to include the database column in the export. You can add all base table columns or reference tables for export using the tool bar or you can delete them from the export data.

Table 71: Meaning of Entries in the Column Selection Tool Bar

Icon	Meaning
	Select all columns in the table for export
	Remove all columns

For certain columns you can specify whether the actual column value or the display name should be exported. This may be necessary for columns with special formats such as multi-language or with a fixed number of decimal places. For this, you use the check box in the column <Export display name>.

Changing the Column Order

The selected columns are displayed in the box <Column to export>. You have the option here to change the column order for exporting. To do this use the entries in the tool bar.

Table 72: Meaning of the Entries in the Column Sort Order Tool Bar

Icon	Meaning
	Sort column alphabetically
	Move column up
	Move column down

Limiting Export Data Sets

You can formulate a condition in the <Condition> box to limit the export data sets. The condition is defined as a valid database query WHERE clause. You can enter this directly as an SQL query or you create it with a wizard.

Previewing and Exporting the Data

Create a preview of the export in the <Export data> box and export the data to a CSV file. The box has its own tool bar to do this.

Table 73: Meaning of Entries in the Data Export Tool Bar

Icon	Meaning
	Create an export preview
	Export the data to a CSV file

The data sets that satisfy the export criterion are show in the preview in table form. Click with the mouse in a column in the result list table header to sort by the selected column. This has no effect on the order that the data is exported.

The data is exported in the same order as displayed in the preview. Use the button on the tool bar to start the export. This opens a dialog window where you can enter a path and file for storing the CSV file.

7.13.2 Saving and Loading Export Definitions

You can save the export definition in the user configuration or in an XML file and load it again from there. If you save export definitions in the user configuration they are only available to yourself. To make them available to other users, save the definitions in files. You can read in and save the export definitions with the icon buttons in the form header. There is also a pop-up menu for the buttons.

Table 74: Meaning of Item in the Standard Toolbar

Icon	Meaning
	Load export definitions
	Save export definitions

A dialog is opened where you can save or load the files by entering a path and file name for the XML file or the name of the export in the user configuration. You can also select export definitions from your user configuration and delete them.

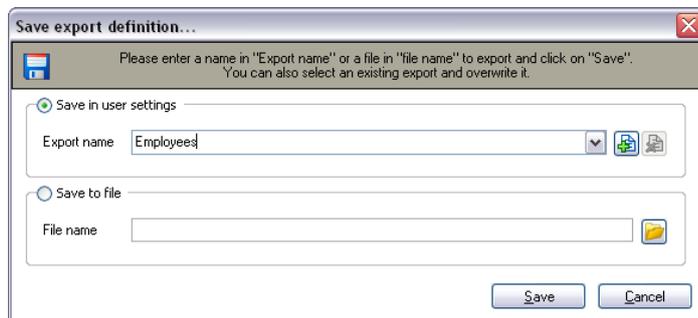


Figure 240: Saving Export Definitions

7.14 Information about DBScheduler Calculation Tasks

Changes to inheritance relevant data resulting from assignment changes, changes to system data or user interface modifications for a system user, make it necessary to recalculate the data. These calculation tasks are set up in the DBQueue and processed by the DBScheduler. The DBScheduler is run at regular intervals by a database schedule "VID_DBScheduler".

You can start processing calculation tasks manually if required and if you have the necessary administrative rights. As long as you have the required authorization for this functionality you can open the dialog window by right clicking on the database status icon in ActiveEntry Manager.

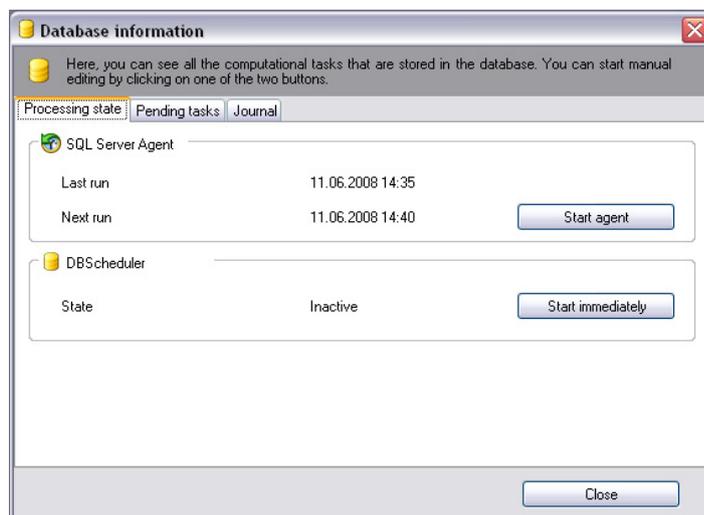


Figure 241: Extended DBScheduler Information

On the tab <Processing state> you can see all the information about the state of the SQL Server Agent and the state of the DBScheduler. You can start the DBScheduler on the server side with the SQL Server Agent (<Start agent> button) or directly with the logged on user's connection (<Start immediately> button). Use the <Close> button to close the dialog window. All the tasks

that are waiting in the DBQueue are displayed on the tab <Pending tasks>. They will be processed the next time the DBScheduler is run. The most recent DBScheduler entries are displayed on the <Journal> tab (see Process Orchestration, section [ActiveEntry Service Logging](#)). User the <Close> button to close the window.

GLOSSAR

A

Active Directory (AD)

LDAP based directory server from Microsoft that was introduced with Window 2000.

Active Directory Service (ADS)

Directory service implementation from Microsoft.

ActiveEntry

Product for provisioning IT and other company resources.

ActiveEntry Service

A server system service. ActiveEntry Service handling processing.

ActiveEntry Designer

Main configuration interface for ActiveEntry.

ActiveEntry Identity Manager

Main administration tool for managing employees, user accounts and permissions within an ActiveEntry network.

ActiveEntry IT Shop

Web based application that provides various workflows.

In ActiveEntry IT Shop you can change employee master data, edit employees, request company resources in the IT Shop, delegate roles, modify approvals, attestations or rule violations.

ActiveEntry Manager

Main administration tool for displaying and editing all the information in an ActiveEntry network.

ActiveEntry Service Updater

Program for updating ActiveEntry Service on Job servers.

ActiveEntry Transporter

Program for exporting objects and custom changes from an ActiveEntry database to an ActiveEntry database.

ActiveEntry Unified Namespace (UNS)

ActiveEntry Unified Namespace (UNS) is a virtual target system for mapping various target system along with their containers, user accounts, target system groups and associated memberships. The data for all target systems that are connected to ActiveEntry is mapped in the ActiveEntry Unified Namespace. This allows other core ActiveEntry functions, such as compliance testing, attestation or IT Shop to be used across target system. The target systems Active Directory, Lotus Notes, SAP R/3, LDAP and Windows NT can also mapped like your own applications, e.g. a telephone system.

ALE

see Application Link Enabling (ALE)

Application

An application is a configuration neutral component assigned for a particular use.

Application group

A global group for assigning applications to users.

Application Link Enabling (ALE)

SAP technology for integrating and running distributed applications on different SAP systems. Refer to your SAP system documentation for further details.

Application role

ActiveEntry application roles are functional roles that you use to specify entitlements to ActiveEntry functions which result from ActiveEntry user tasks from within the company structures. Application roles take administration and approval processes into account. Application roles are preset but may be changed and extended.

Application Server

Servers that manage a directory structure which represents the source for automatic software installation on the connected client PC's.

ActiveEntry Service Configurator

Program for configuring ActiveEntry Service.

Approval procedure

Method for granting approval for customer requests withing the IT Shop. An approval procedure is made up of approval policies that can contain several approval levels. Several approval steps can be defined for each approval level. A different approver can be specified for each approval step.

Approver

The approver is an employee that can grant or deny approval in a procedure for approving a request (renewal or cancellation).

Approval policy

Specifies which approval workflow should be in used IT Shop for an attestation instance or a request (renewal or cancellation).

Approval procedure

Finds the attestor for the current attestation instance or the approver for the current request (renewal or cancellation) in IT Shop.

Approval workflow

Specifies which approval procedure will be used in which order in attestation instances or requests (renewals or cancellations) in IT Shop. An approval workflow contains at least one approval level with at least one approval step.

Assignment request

Requests for company resources, employees, hardware or workdesk for roles. You can request assignments for departments, cost centers, locations or business roles via the ActiveEntry IT Shop, then they are authorized via approval process.

Attestation

A method for authorizing data or internal rules. Attestation functionality in ActiveEntry is used by manager or other in authority to certify the correctness of editing permissions, entitlements, requests or exception approvals on a regular or manual basis.

Attestation instance

Objects that are created as soon as attestation is automatically or manually started. When attestation is triggered, ActiveEntry creates an attestation instance for each attestation object. Attestation data is saved in the attestation instance. This includes the attestation object, status (open, approved, denied), date of attestation, the attestor.

Attestor

The person that will carry out the attestation. Attestors approve data that is presented in an attestation instance, or they deny it.

Authentication module

Authentication modules are used to define how user should log onto ActiveEntry tools. Users can log in as, for example employees, ADS users or system users. The authentication module determines which system user is directly or indirectly assigned to the logged in user. This assigns user permissions for the user interface elements of the administration tool that has been started and for the database objects.

Authorization definition

Group of transactions and authorization objects in ActiveEntry to be tested by an SAP function.

Authorization Editor

Tool for edition the authorization definition for an SAP function.

Authorization Field

An object in an SAP system. The smallest unit that can be granted authorizations. To do this, authorization fields are given fixed values (activities of data). Up to 10 authorization fields are grouped together in one authorization that only result in a valid authorization when together.

Authorization Object

An object in an SAP system. Makes the definition of authorizations possible in an SAP system. Comprises of upto 10 authorization fields that are connected with an AND link.

C

Cancellation workflow

Entscheidungsworkflow, durch den die Entscheider ermittelt werden, wenn ein bestelltes Produkt abbestellt wird.

Cart

This is used to collect products in the IT Shop that can be requested at any time. A customers can add as many carts as they want. The cart is deleted as soon as the requests contained in it have be carried out.

Cart item

A product that is assigned to a shopping cart.
A cart item makes it evident which product should be requested by whom for whom.

Central User Administration (CUA)

Function in SAP for administrating user in a central system rather than maintaining all clients separately.

Clients in different SAP system are grouped together in a system network. Users of these clients are maintained in a central system and the data is distributed to client systems. Therefore, users that own permissions in different clients do not have to individually maintained. Roles and profiles are administrated in client systems but can only be assigned to users in the central system.

Refer to your SAP system documentation for more details.

Common Information Model (CIM)

System administration protocol for monitoring and configurin system components and attributes. CIM represents a consistant and uniform view of all types of logical and physical objects in a system management environment.

Company resource

Umbrella term for all objects that are assigned to employees, hardware, workdesks or roles or that canbe requested via the IT Shop and not roles themselves. Company resources are: applications, drivers, system entitlements, resources, target system groups, PXE menus, inventory items, system roles.

Configuration parameter

Parameter for configuring the basic settings for ActiveEntry system administration.

Preprocessor relevant configuration parameter are configuration parameters that are connected to a preprocessor condition. If a preprocessor relevant configuration parameter changes, the database has to completely recompiled.

Configuration Parameter Editor

ActiveEntry Designer editor for customizing configuration parameters

Crypto Configuration

A program for encrypting the database contents of an ActiveEntry database.

CUA

see Central User Administration (CUA)

CUA Status

Labels an SAP client for use as central system or client system in the central user administration. Clients that should be excluded from the Central User Administration are labeled with the CUA status "None".

Customer

A company employee that is entitled to request items from the IT Shop. An employee becomes a customer when assigned to a shop.

Customers form an IT Shop solution in combination with shelves, products, shops and shopping centers.

D

Database schema

A logical description of data that are saved in a database. The schema not only defines names for individual data items, their size and other characteristics, but also identifies the relation between the data. The ActiveEntry data model differentiates between reference data and metadata. Reference data is described by the application data model, the metadata by the interface data model.

DBCompiler

Program for compiling the ActiveEntry database after changes have been made.

DBMigrator

Program for migrating an ActiveEntry database.

DBScheduler

The DBScheduler is used to calculate processing task from the DBQueue (table "DialogD-BQueue). The DBScheduler comprises of a combination of saved procedures and triggers. The DBScheduler also controls recurring tasks on a cyclical basis such as daily maintenance tasks for calculating statistics or indexing the database.

Default user

The default user for a workstation is the user account assigned to the workstation. The result being that one PC can have several default users.

Default PC

An employee's default PC is determined by the workdesk entered in the employees data. This means that an employee can only have one default PC.

Delegation

Special assignment request form

In the case, an employee passes any number of role assignments to another employee for a limited period of time. Delegations can be authorized via an approval procedure.

Desktop Management Interface (DMI)

Industry standard for management and control of desktop PC, notebook or server components. DMI was the first desktop management standard and today is part of the Web-Based Enterprise Management (WBEM) Initiative.

Desktop Management Task Force (DMTF)

Standards organization that develops and maintains standards for systems management of IT environments in enterprises and in the internet.

Discontinue inheritance

The property "Discontinue inheritance" indicates that the option "End of inheritance" is set in the master data of any role marked in this way.

Distribution model

Relationships between logical systems are defined in the SAP distribution model. It is used by Application Link Enabling to control data distribution amongst others.

Refer to your SAP system documentation for more details.

Domain Name System (DNS)

The Domain Name System (DNS) is a distributed database that manages namespaces in the internet.

Driver

The driver is a configuration neutral component assigned for a particular use. The "User" in this case is the workstation.

Dynamic Host Configuration Protocol (DHCP)

Standard for administration of dynamic settings and addresses in a network. DHCP makes it possible to dynamically assign an IP address with the help of a DHCP server and other configuration parameters on computers in a network.

E

Enterprise Resource Planning (ERP)

Identifies the company task for planning the use of existing company resources in the most efficient way for daily operations.

Exception approver

A person that can approve rule exceptions.

Exception approvers are only those employees that are assigned to at least one compliance rule as exception approver via the application role <Identity Audit>\<Exception approver>.

F

First Distribution Server (FDS)

Application servers that serve the administration of shared applications, drivers and machine profiles are labeled as First Distribution Servers (FDS). The FDS is the at the top level of the application server hierarchy.

Function Instance

Function definition that is given values for a specific application. A specific SAP client to be used in the SAP function is given in the function instance. Furthermore, variable that are allocated to authorization fields are given fixed values. Function instances can only be set up for active SAP functions.

Function Element

A general term for transactions, authorization objects and authorization fields that are displayed in an authorization definition as a tree structure in the Authorization Editor.

G

Business Role

Business roles represent customized functions in ActiveEntry. You can use them to model approval workflows, assignments or approval procedures according to the needs of you organization structure. All business roles are specified by your company.

Global shelf template

Template that you can use to automatically generate shelves in all IT Shop shops. A global shelf can be assigned company resources (as products) and approval policies.

H

HistoryDB

Archiving system for data changes.

HistoryDB Manager

Administration tool for displaying and editing all the information in the HistoryDB archiving system.

HistoryDB Service

System service on the servers. The HistoryDB Service imports log entries into the HistoryDB archiving system.

Hypertext Transfer Protocol (HTTP)

Protocol for transferring data.

I

IT Shop

Program component for providing employees with company resources via a defined approval procedure. IT Shop solutions are setup in ActiveEntry Identity Manager and can then be used in the ActiveEntry IT Shop.

IT Shop Structure

Role classes are used to group the components of an IT Shop solution i.e. shopping center, shop, shelf, customer.

J

Job Queue Info

Programs for monitoring the current state of the services running in an ActiveEntry network.

Job destination

ActiveEntry Service component. The Job destination handles the process steps and returns the result back to the Job provider.

Job provider

ActiveEntry Service component. A Job provider delivers process steps to the Job destination and evaluates the results.

Job queue

Central storage for process component generated actions to be executed.

Job server

Server running elementary tasks.

Job Server Editor

ActiveEntry Designer for editing Job server properties.

Job service

See ActiveEntry Service.

L

Language Editor

ActiveEntry Designer Editor for translating text captions.

Lightweight Directory Access Protocol (LDAP)

Network protocol that permits queries and modifications to directory service's information (a hierarchical database distributed on a network).

List Editor

ActiveEntry Designers basic editor with which list can be displayed and edited.

License Wizard

Wizard for creating a license request for using ActiveEntry and installing the ActiveEntry license.

Lotus Notes

Document oriented distributed database system with a very tight email connection.

M

Main Library Server (MLS)

The Main Library Server stores profiles that should be available to several sites. Only one MLS can be defined in an ActiveEntry database. This is set up on the site's FDS.

Managed Information Format (MIF)

MIF defines the managed objects and attributes associated with them. Each instance of a managed object has to have its own MIF file that describes its manageable aspects. The MIF file is divided into blocks.

Managed Object

Name for an resource that is represented through an object in a network management system (WMI).

Managed Object Format (MOF)

MOF is a compiled language developed by DMTF and based on Interface Definition Language (IDL). MOF can be used to define statistical or dynamic classes or class instances.

Machine profile

See Machine type.

Machine type

Machine types are used in the administration of configuration parameters for Unattended Setup (UAS) of machines. Since machine types are dealt with in the same way as application and driver profiles in ActiveEntry, they are also called machine profiles.

Mitigating Control

A control that should be carried out if a compliance rule is violated or an SAP function matches. Mitigating controls are independent of ActiveEntry functions. For example, the risk that is connected with a rule violation can be reduced by regular manual checking of prohibited authorizations.

N

Namespace Mapping Editor

Program for creating and editing a mapping file for extending templates for target system specific process components.

NetBIOS

Network Basic Input Output System - Programmers interfaces developed by IBM to make communication between two network programs possible. NetBIOS allow 16 character for a NetBIOS name. Microsoft limited NetBIOS names to 15 characters because the 16th character is used as a NetBIOS suffix.

O

User Interface Editor

ActiveEntry Designers editor for editing the administration tool's user interface.

Object definition

Object definitions create a view for database objects that can be differated by their properties and therefore allow an additional control function.

Object Editor

ActiveEntry Designers basic editor for displaying and editing all objects.

Organization

The company structures department, cost center and location are called organizations in ActiveEntry.

Org level

An object in an SAP system that defines fixed values for authorization fields. Org levels are, for example, custom accounting codes, functional areas or account types.

P

Patch

Software update.

Permissions Editor

ActiveEntry Designer editor used to grant table and column permissions to permissions groups and system users.

Plugin

Additional software module.

Preprocessor condition

Condition for posing restrictions on program code during compilation. Conditional compilations allows parts of the program code to be included but excludes other parts. Preprocessor conditions are defined via configuration parameters and their options.

Process

Stringing together process step into a sensible order. The process has the task of mapping live processes.

Process Editor

ActiveEntry Designer editor for handling process steps and processes.

Process function

Task executed by a process.

Process parameter

Parameter permitted for a single process component task.

Process step

Separate parts of a process. A process step represents one work procedure.

Process component

Elementary component available for use in process steps.

Product

Company resource that is assigned to an IT Shop shelf and therefore can be requested. Products form an IT Shop solution in combination with shelves, customers, shops and shopping centers. Only company resources that are assigned to a service item and labeled with the option <IT Shop> can be added as products to the IT Shop.

Production Application Server (PAS)

A Production Application Server (PAS) provides the user profiles. The PAS lies under the FDS in the application server hierarchical structure. The PAS is used to reduce the load on the FDS and represents a 1:1 mapping of the FDS. ActiveEntry Service is responsible for maintaining continuity between PAS and FDS.

Profile

A software application (application, driver, patch) prepared for automatic installation. A profile is normally operating system dependent.

Profile Editor

Program for editing software profiles.

Profile Scanner

Program for creating software profiles.

Provider client

The provider client is a completely configured ActiveEntry customer environment with a database, ActiveEntry Service and possibly ActiveEntry front-ends. The provider client actively administers a network. In addition to the usual ActiveEntry environment, the provider client can process its own ActiveEntry Service requests that are executed on the provider master.

Provider master

The provider client is a completely configured ActiveEntry provider environment with a database, ActiveEntry Service and possibly ActiveEntry front-ends. The provider master does not necessarily administer its own network but does however, contain additional information about the provider clients in its administration. The provider master keeps a queue for provider clients requests.

Provider mode

Provider mode is a model that stores and changes information in a central ActiveEntry environment. The information is transferred into mainly independent ActiveEntry environments and take effect there.

R

Renewal workflow

Approval workflow that finds the approver if a requested product needs to be renewed.

Replication Info

Program for monitoring replication of software profiles.

Request template

Template for a cart containing cart items that are often requested together. Public request templates are available to all ActiveEntry users the moment they are shared. Non-public request templates can only be used by the request template owner.

Resource

An existing item for solving a particular task.

Resource type

Objects that are used to sort resources corresponding to usage.

Processing steps for resource types can be defined that need to be run when a resource is successfully assigned to an employee.

Role

The term "role" is an umbrella term for the company structures departments, cost centers, locations and business roles. Roles in ActiveEntry are all objects through which employees, hardware or workdesks can be assigned company resources. Therefore IT Shop structures are also roles in the ActiveEntry sense of the word.

Examples of roles are the department: "Development," location "Prague," product "FrameMaker - German - 9.0".

Role classes

Objects that group together similar roles.

Role classes are defined in ActiveEntry to differentiate between various company structures. Role classes regulate inheritance behavior in these company structures. Furthermore, they specify which company resource assignments are possible through a role in a role class.

Examples of role classes are: "departments," "location" or "IT Shop structure." Define custom role classes in order to create business roles.

Role type

Company specific criteria for allocating roles.

Role types are mainly used to regulate inheritance of approval policies within an IT Shop structure. To do this you define role types that you assign to the approval policies and IT Shop rules. In addition, you can use role types to structure business role or shops in the IT Shop by criteria.

S

SAM Database

Security Accounts Manager - secure account administration under Windows.

Administration of user accounts and encoded password is done in the SAM database.

SAP Authorization

Authoriation permissions that that SAP users obtain on the basis of the SAP roles assigned to them in the SAP system.

SAP Function

An object in ActiveEntry that can be used to test which SAP authorizations an SAP user in an SAP client has effectively.

SAP function category

An object for grouping SAP functions.

SAP menu

Element for guiding users through the SAP GUI. Authorizations are linked to fixed menu items with authorization objects. Authorization objects can be linked into authorization objects via the choice of SAP menu in the ActiveEntry Authorization Editor.

SAP R/3

Product from the company SAP AG.

Schedule

Task to be performed on a cyclical basis.

Schedule Editor

ActiveEntry Designer editor for setting up scheduled tasks for procedures that are to be executed cyclically.

Schema Extension

Program for extending the ActiveEntry database schema with custom tables and columns.

Schema Editor

ActiveEntry Designer editor for customizing database schema table and column definitions.

Secure Sockets Layer (SSL)

Transfer protocol that enables encoded communication.

Service catalog

Displays all requestable service items grouped by service category.

Service items for products that are assigned to IT Shop shelves are displayed in the service catalog.

Service category

Grouping criteria for service items

A product's service item must be assigned to a service category in order to select the product from the service catalog.

Service item

These are objects that are necessary to book company resources internally.

Service items must be assigned to company resources so that they can be requested and booked internally as products in the IT Shop. A service item contains an exact product definition, assignment to a cost center, price information.

Service Provisioning Markup Language (SPML)

Service Provisioning Markup Language is an XML based description language that is used as an exchange format for user and resource information between provisioning systems. The standardization of SPML has been driven by the OASIS consortium (Organization for the Advancement of Structured Information Standards, www.oasis-open.org) which includes some well-known software companies. The latest version (2.0) was released in April 2006.

Shelf

An IT Shop structure that is part of a shop and can be assigned products.

Shelves form part of a hierarchical IT Shop solution along with customers, shops, shopping centers and products.

Shelf template

Template that you can use to automatically generate shelves in IT Shop and fill them with company resources.

You can use shelf templates when you want to setup shelves in several shops with identical products. ActiveEntry differentiates between global shelf templates, special shelf templates and shopping center templates.

Shop

An IT Shop structure that is assigned shelves and customers.

Shops form a hierarchical IT Shop solution along with customers, shelves, shopping centers and products. Each shop contains a shelves that the shop customer can request items from.

Shopping cart

See Cart.

Shopping center

IT Shop structure for group shops together.

Shops form a hierarchical IT Shop solution along with customers, shops, shelves and products.

Shopping center template

Template that you can user to replicate a shelf from a special shelf template in all the shops in a shopping center. To do this, the shopping center template must be assigned to at least one special shelf template.

Site

A group of servers, workdesks and user for means of software distribution.

Software Loader

Program for loading new or changed files in the ActiveEntry database. They can then be distributed in the ActiveEntry network through automatic software updating.

Software profile

See Profile.

Special shelf template

Template that you can use to automatically generate shelves in selected shops in the IT Shop. A special shelf template can be assigned company resources as products and approval policies. The shops that should be replicated by the shelf template are selected individually.

System role

A system role is a resource in which any number of company resources can be grouped together.

System roles are used to simplify assignment of different company resources. If a system role is assigned to an employee or a workdesk, they receive all the company resources that are assigned to the system role. This might be system permissions, applications or non-IT Shop resources.

System roles can be assigned directly to employees or workdesks, requested via the IT Shop or assigned to roles.

System user (1)

A predefined user that contains several entitlements to ActiveEntry functions. The system user obtains these entitlements via his or her permissions groups assignments. A system user is assigned to user during the administration tool login procedure. Entitlements for the ActiveEntry functions are passed onto the user from this system user.

Certain system users are included in the ActiveEntry installation. Further system users can be defined in ActiveEntry Designer.

System user (2)

An authentication module for logging onto ActiveEntry tools. See Authentication module.

System user ID

The user ID that a user enters to log onto an ActiveEntry tool.

The system user ID is independent of the selected authentication module. It can be a login name for an ADS domain or a system user e.g. a central user account.

T

Target system

A system in which employees under ActiveEntry administration have access to network resources.

Example: Active Directory, SAP R/3, Lotus Notes

Target system area

Administration unit in a target system for user accounts, user groups and machine accounts.

Example: Active Directory domain, SAP R/3 client, Lotus Notes domain.

Template

Rule for mapping object properties. Templates can be used within an object as well as across objects.

Test Application Server (TAS)

A Test Application Server (TAS) is used for creating and testing application, driver and machine profiles. If the profiles are tested successfully they can be replicated on the First Distribution Server.

Text comparison

A procedure in SAP which mirrors names of roles and profiles from a CUA client system in the central system. The roles and profiles in the central system are only made known when the text comparison has been run at least once. Then they can be assigned to user.

Roles and profiles from client systems cannot be synchronized with ActiveEntry until the text comparison has been run in SAP.

Refer to your SAP system documentation for more details.

Transaktion

An object in an SAP system that start an ABAP program.

U

UID

The UID is a artificial primary key that is created by the operating system as soon as the object is inserted in the database. The UID is a unique value which does not alter even when changes are made to the object properties. An object is labeled with a UID and can be uniquely referenced with it.

Unattended Setup

A structure that helps in the installation and configuration of workstations using minimal resources.

UNS

See ActiveEntry Unified Namespace (UNS)

User & Permissions Group Editor

ActiveEntry Designer editor for editing permissions groupa and system users.

V

Variable set

A group of all variables and their values that can be used in the authorization definition of an SAP function. Variable sets are used to set up fucntion instances for one and the same function definition.

VI Client

Software for automatically installing software (applications, drivers) on client PC's.

VI Access Permissions

The VI access permissions allow system users to change VI specific objects. The VI access permissions are limited to a set time period and must be applied for separately.

W

Web-Based Enterprise Management (WBEM)

Suggested protocol for system management and system administration using WWW interfaces.

Windows Internet Name Service (WINS)

The Windows Internet Naming Service (WINS) is a software service developed by Microsoft that dynamically assigns IP addresses to computer names (NetBIOS names).

Windows Management Instrumentation (WMI)

Microsoft Implementation of Web-Based Enterprise Management (WBEM) technology for entering management information.

Windows NT

An operating system developed by Microsoft.

Workflow Editor

An editor that you can use to create workflows for attestation instances or approval processes. In the Workflow Editor, approval levels and steps from an approval workflow are inserted via a special graphical control. Approval levels can be arranged in any way and connected to each other.

FIGURES

Figure 1:	Upgrading ActiveEntry if Older Versions Exist.	23
Figure 2:	Startup Screen for "DBMigrator"	29
Figure 3:	Creating the Database Connection	30
Figure 4:	Connection Data for the Database Login.	30
Figure 5:	Selecting the Migration Files.....	31
Figure 6:	Displaying the Migration Steps	31
Figure 7:	Alert Box	32
Figure 8:	Database Backup Alert Box.....	32
Figure 9:	Migration.....	32
Figure 10:	Display Current User	33
Figure 11:	Migration Completed	33
Figure 12:	Logging Processes Changes.....	34
Figure 13:	Task Set Up for the DBScheduler.....	34
Figure 14:	Creating a Dialog Login	35
Figure 15:	Connection data for the Login.....	35
Figure 16:	License Status Advice.....	36
Figure 17:	Completing the Database Connection Data.....	36
Figure 18:	Compiling the Database.....	37
Figure 19:	Software Update Alert.....	38
Figure 20:	Importing Files to the Database	38
Figure 21:	Enable Automatic Software Update	38
Figure 22:	Ending the Program	39
Figure 23:	Reset System Update Option	39
Figure 24:	License Wizard Startup Screen	41
Figure 25:	Preparing a License Request	42
Figure 26:	Creating a Database Connection.....	42
Figure 27:	Login	43
Figure 28:	Selecting Product and License Type	43
Figure 29:	License Data Input	44
Figure 30:	Saving and Sending the License Request.....	44
Figure 31:	Exiting the License Wizard.....	45
Figure 32:	License Wizard Startup Screen	45
Figure 33:	Preparing the License Installation	46
Figure 34:	Connecting to the Database	46
Figure 35:	Login	47
Figure 36:	Installing the License	47
Figure 37:	Completing the License Wizard	48
Figure 38:	ActiveEntry Service Operating Mode.....	49
Figure 39:	Example Configuration for FileJobGate.....	51
Figure 40:	Startup Window for the Program "ActiveEntry Service Configurator"	52
Figure 41:	Selecting a Module Type	54
Figure 42:	SQLJobProvider Configuration Data	56
Figure 43:	OracleJobProvider Configuration.....	57

Figure 44:	FileJobProviders Configuration Data	58
Figure 45:	FTPJobProvider Configuration Data	60
Figure 46:	HTTPJobProviders Configuration.....	61
Figure 47:	Example Configuration for Web Service Job Processing	62
Figure 48:	WebServiceJobProvider Configuration Data	62
Figure 49:	JobServiceDestination Configuration Data	64
Figure 50:	FileJobDestination Configuration	66
Figure 51:	FTPJobDestination Configuration Data	67
Figure 52:	HTTPJobDestination Configuration Data.....	68
Figure 53:	General Configuration Settings	69
Figure 54:	Log Writer Configuration Data	71
Figure 55:	FileLogWriter Configuration File.....	72
Figure 56:	Dispatcher Configuration Data	73
Figure 57:	Dispatcher Configuration Example	74
Figure 58:	Connection Configuration Data	75
Figure 59:	HTTPLogPlugins Configuration Data	77
Figure 60:	StatisticsPlugins Configuration	78
Figure 61:	ScheduleCommandPlugin Configuration Data	79
Figure 62:	DBSchedulerWatchDogPlugins Configuration.....	80
Figure 63:	RequestWatchDogPlugins Configuration Data.....	81
Figure 64:	PerformanceCounterPlugins Configuration Data.....	82
Figure 65:	ActiveEntry Service Entry	83
Figure 66:	Example of a Cluster Solution	84
Figure 67:	Setting the JobDestination in the ActiveEntry Service Configuration File	86
Figure 68:	Cluster Administrator	87
Figure 69:	Adding a new Cluster Resource.....	87
Figure 70:	Adding Physical Nodes.....	88
Figure 71:	ActiveEntry Services Dependencies	88
Figure 72:	ActiveEntry Service, Service Name	89
Figure 73:	Additional Registry Key	89
Figure 74:	Cluster Resource Success Alert.....	89
Figure 75:	Cluster Administrator	90
Figure 76:	Adding a New Cluster Resource	91
Figure 77:	Adding the Physical Nodes.....	91
Figure 78:	Cluster Resource Dependencies.....	92
Figure 79:	Cluster Resource Parameters	92
Figure 80:	Access Rights to the Share	93
Figure 81:	Cache Settings	93
Figure 82:	Cluster Resource Creation Success Alert	93
Figure 83:	Changing the Logwriter in the ActiveEntry Service Configuration File	94
Figure 84:	"ActiveEntry Transporter" Startup Screen.....	97
Figure 85:	Creating a Database Connection.....	97
Figure 86:	Selecting the Transport Package	98
Figure 87:	Importing the Data	98
Figure 88:	Tasks Set Up for the DBScheduler	99

Figure 89:	Compiling the Database	99
Figure 90:	Quitting the Program	100
Figure 91:	"ActiveEntry Transporter" Startup Screen	101
Figure 92:	Creating a Database Connection	101
Figure 93:	Transport Package Name	102
Figure 94:	Data Export	102
Figure 95:	Specifying Export Criteria	103
Figure 96:	Exporting Transport Data	103
Figure 97:	Quitting the Program	104
Figure 98:	Exporting Change Labels	104
Figure 99:	Contents of a Change Label	105
Figure 100:	User Selection for Transport by Change Information	106
Figure 101:	Selecting a Date for Transport by Change Information	106
Figure 102:	Selecting Tables for Transport by Change Information	107
Figure 103:	Transport Data Display	107
Figure 104:	Displaying Schema Extensions	108
Figure 105:	Selecting Single Objects	108
Figure 106:	Selecting System Files	110
Figure 107:	Login	114
Figure 108:	Updating the Job Server with ActiveEntry Service Updater	115
Figure 109:	"Software Loader" Startup Screen	116
Figure 110:	Connecting to the Database	116
Figure 111:	Login	117
Figure 112:	Selecting the Transfer Direction	117
Figure 113:	Specifying the Severity Level for New Files	118
Figure 114:	Change Labels for New Files	118
Figure 115:	Selecting a Change Label	119
Figure 116:	Selecting Which Files to Load	120
Figure 117:	Question Loading New Files	120
Figure 118:	Loading the Files into the Database	121
Figure 119:	DBScheduler Computational Tasks	121
Figure 120:	Program Exit	122
Figure 121:	"Software Loader" Startup Screen	122
Figure 122:	Connecting to the Database	123
Figure 123:	Login	123
Figure 124:	Selecting the Transfer Direction	124
Figure 125:	Selecting the Application Group	124
Figure 126:	Selecting the Directory	125
Figure 127:	File Selection	125
Figure 128:	Exporting the Files	126
Figure 129:	Program Exit	126
Figure 130:	Startup screen for the DBCompilers	127
Figure 131:	Creating the Database Connection	127
Figure 132:	Login	128
Figure 133:	Completing the Database Connection Information	129

Figure 134:	Displaying Outstanding DBScheduler Tasks before Starting the Compilation ..	129
Figure 135:	Specifying which Database Parts to Compile.....	130
Figure 136:	Selecting Which Scripts to Compile	131
Figure 137:	Selection of processes to Compile	131
Figure 138:	Selecting processes to be Compiled.....	132
Figure 139:	Option for Testing SQL Procedures	132
Figure 140:	Selecting the Conversion Scripts	133
Figure 141:	Compiling	133
Figure 142:	Error Message Output	134
Figure 143:	Warning Message Output.....	134
Figure 144:	Quitting the DBCompiler.....	135
Figure 145:	Startup screen for the Program "Crypto Configuration"	135
Figure 146:	Creating the Database Connection	136
Figure 147:	Login	136
Figure 148:	Creating a Database Key	137
Figure 149:	Creating a New Private Key (1).....	138
Figure 150:	Generating a New Private Key (2)	138
Figure 151:	Creating a New Private Key (3).....	139
Figure 152:	Database Tables and Columns with the property "Encrypted"	139
Figure 153:	Database Conversion Prompt	140
Figure 154:	Backup Prompt.....	140
Figure 155:	Conversion Progress Bar.....	140
Figure 156:	Quitting the Program.....	141
Figure 157:	Database ID Verification	144
Figure 158:	Complete Database Connection Informationen	145
Figure 159:	Encrypted Database Alert	146
Figure 160:	Input Mask for Connection Data under Microsoft SQL Server	146
Figure 161:	Input Mask with Oracle Connection Data	146
Figure 162:	Request to Login Again to the Database	147
Figure 163:	Connection Dialog at Program Startup.....	159
Figure 164:	Input Mask for Connection data under Microsoft SQL Server.....	160
Figure 165:	Database Connection Advanced Options.....	160
Figure 166:	Input Mask for Connection Data under Oracle.....	160
Figure 167:	Selecting the Connection.....	161
Figure 168:	Connecction Dialog with Login to Administration Tools	166
Figure 169:	Error Message Logging in with a System User ID	167
Figure 170:	Program's Title Bar	170
Figure 171:	The Program's Status Bar - Simple	170
Figure 172:	The Program's Status Bar - Extended.....	170
Figure 173:	The Program Menu Bar	171
Figure 174:	Toolbar	175
Figure 175:	Default Program Layout in Simple Mode	178
Figure 176:	Standard Program layout in Advanced mode	179
Figure 177:	Search Dialog	180
Figure 178:	Filter Dialog	181

Figure 179:	The Change User Password Dialog.....	182
Figure 180:	User Settings	183
Figure 181:	Special features of individual Program Components.....	185
Figure 182:	Default Application Settings	186
Figure 183:	Available Plugins.....	187
Figure 184:	View of the Navigation Hierarchy Using the Category <Employee> as Example ... 188	
Figure 185:	Result List	191
Figure 186:	Master Data Form Layout	193
Figure 187:	Help box for a data field	194
Figure 188:	Selection Dialog showing Help Themes for a Form	194
Figure 189:	Example of Elements in an Overview Form.....	195
Figure 190:	Mandatory Input Field.....	196
Figure 191:	Entering a Database Query	197
Figure 192:	Input Field for List of Defined Values (with and without defined entries)	198
Figure 193:	MVP Column Input Field	198
Figure 194:	Multi-language Input Field.....	199
Figure 195:	View OnlyMulti-language Field.....	199
Figure 196:	Editable Multi-language Field	199
Figure 197:	Pop-up Menu.....	199
Figure 198:	Extended Pop-Up	200
Figure 199:	Check Box	200
Figure 200:	Radio Buttons.....	200
Figure 201:	Table	201
Figure 202:	Pop-up Menu with Adjacent Button	201
Figure 203:	Form with Several Tabs	201
Figure 204:	Control Element for Setting the Date.....	202
Figure 205:	Control Element for Login Times	203
Figure 206:	Graphical Representation of Assignments in a Control Element	203
Figure 207:	Displaying Assignments in a Membership Tree	205
Figure 208:	Wizard for Entering Database Queries.....	207
Figure 209:	Rule Editor in Simple Definition Mode	208
Figure 210:	Rule Editor in Advanced mode	208
Figure 211:	Toolbar for Report „Overview of all assignments“	209
Figure 212:	Legend for the Report „Overview of all assignments“	210
Figure 213:	Master Data Form with Multi-Editing.....	211
Figure 214:	Task View	212
Figure 215:	Favorites View	213
Figure 216:	Favorites Toolbar	213
Figure 217:	Example Error Message Window	214
Figure 218:	Logging Errors in the Error Log	215
Figure 219:	Process Information Form (below) and the Log (above).....	218
Figure 220:	Toolbar for Viewing the Process Information.....	218
Figure 221:	Process Log Toolbar	219
Figure 222:	Configuration of the Process Information	221
Figure 223:	Displaying Processes	222

Figure 224:	Logging the data changes withing a process	224
Figure 225:	Historical Data Analysis.....	225
Figure 226:	TimeTrace View Showing Change Time Stamps via a Context Menu	227
Figure 227:	Stammdatenformular mit historischen Daten	227
Figure 228:	Transferring Historical Data	228
Figure 229:	Simulation Data Overview.....	230
Figure 230:	Logging the Calculation Tasks for the DB Scheduler.....	230
Figure 231:	Logging Generated Processes	231
Figure 232:	Logging Trigger Changes.....	232
Figure 233:	Logging Changed Objects.....	233
Figure 234:	Logging Rule Violations	234
Figure 235:	Planning an Execution Schedule	235
Figure 236:	Overview of Planned Operations	236
Figure 237:	Database Seach View.....	238
Figure 238:	Advanced Database Search.....	239
Figure 239:	Data Export Form	242
Figure 240:	Saving Export Definitions	245
Figure 241:	Extended DBScheduler Information.....	245

TABLES

Table 1:	ActiveEntry Service Configurator - Menu Commands and Key Combinations	52
Table 2:	Meaning of the Icons for the Module Properties	54
Table 3:	Verification Test Error Output	55
Table 4:	Permitted Notification Methods.....	59
Table 5:	Message Types.....	71
Table 6:	Message Types.....	73
Table 7:	Polling Interval Guidelines for ActiveEntry Service	74
Table 8:	Calling Parameters for the Service Startup	83
Table 9:	Meaning of Icons for Object Selection	109
Table 10:	Entries in the Context Menu for Object Selection	109
Table 11:	Severity Level Meaning	110
Table 12:	Permitted Configuration Parameter Values.....	111
Table 13:	Meaning of Status.....	119
Table 14:	Meaning of Status.....	125
Table 15:	Configuration Parameters for Full-text Search	141
Table 16:	Authentication Module and ActiveEntry Tools	161
Table 17:	Icons in the Status Bar	170
Table 18:	General Key Combinations in the Menu Bar	171
Table 19:	The Meaning of Menu Bar Entries	172
Table 20:	Functions in the Standard Toolbar	175
Table 21:	Functions in the Forms Related Toolbar	175
Table 22:	Standard Key Combinations for the Context Menu	176
Table 23:	Key Combinations for the Search Dialog	180
Table 24:	Configuration Parameter for Limiting Results	180
Table 25:	Standard Key Combinations for the Navigation View	189
Table 26:	Items in the Navigation View Context Menu	189
Table 27:	Items in the <My ActiveEntry> Category Context Menu	190
Table 28:	Items in the Navigation View Configuration Menu	190
Table 29:	Standard Key Combinations for the Result List	191
Table 30:	Functions in the Toolbar.....	192
Table 31:	Entries in Result List context Menu.....	192
Table 32:	Use of special consistency icons	193
Table 33:	Standard Key Combinations for Control Elements.....	195
Table 34:	Form Element Icon	196
Table 35:	Meaning of Icons in Advanced Edit Mode	197
Table 36:	Meaning of Icons in the Control.....	204
Table 37:	Items in the Control's Context Menu	204
Table 38:	Meaning of Icons in the Membership Tree.....	205
Table 40:	Meaning of the Icons	206
Table 39:	Items in the Membership Tree's Context Menu	206
Table 41:	Meaning of Icons in the Rule Editor.....	208
Table 42:	Meaning of Icons in the Report Toolbar.....	209
Table 43:	Meaning of the Entries in the Special Toolbar	213

Table 44:	Entries in Favorite List Context Menu	213
Table 45:	Configuration Parametes for Recording System Log	214
Table 46:	Meaning of the Entries in the Special Toolbar	215
Table 47:	Icons Used by Error Logging	215
Table 48:	Configuration Parameters for Recording Process Information	218
Table 49:	Meaning of the Entries in the Special Toolbar	219
Table 50:	Entries in the Process Information View Context Menu.....	219
Table 51:	Meaning of the Entries in the Special Toolbar	220
Table 52:	Configuration of the Process Information View	221
Table 53:	Process Information History	223
Table 54:	Meaning of Icons for Process State.....	223
Table 55:	Configuration Parameter for Recording Process Information	224
Table 56:	Data Change Information	225
Table 57:	Configuration Parameter for Recording Process Information	226
Table 58:	Meaning of Entries in the Special Toolbar	226
Table 59:	228
Table 60:	Icon representing Active Simulation Mode in the Statusbar.....	229
Table 61:	DBScheduler Calculation Task Information.....	231
Table 62:	Generated Process Information	231
Table 63:	Changed Object Information.....	232
Table 64:	Information for Changed Objects.....	233
Table 65:	Rule Violation Information.....	234
Table 66:	Configuration Parameter for Deferred Operation Execution	235
Table 67:	Data Modification Information	237
Table 68:	Meaning of Entries in the Special Tool Bar	237
Table 69:	Predefined Filters.....	237
Table 70:	Syntax for Linking Search Strings	241
Table 71:	Meaning of Entries in the Column Selection Tool Bar	243
Table 72:	Meaning of the Entries in the Column Sort Order Tool Bar	243
Table 73:	Meaning of Entries in the Data Export Tool Bar.....	244
Table 74:	Meaning of Item in the Standard Toolbar	244

INDEX

A

Access rights 60, 62
 Active Directory (AD) 247
 Active Directory Service (ADS) 247
 Active Entry
 installation 23
 ActiveEntry 247
 application role 249
 licensing 41
 ActiveEntry Analyzer 157
 ActiveEntry Designer 155, 247
 ActiveEntry Identity Manager 151, 247
 ActiveEntry Installation
 hotfix 94
 service pack 94
 updating 94
 version change 94
 ActiveEntry IT Shop 152, 247
 post 40
 share 40
 ActiveEntry Manager 152
 ActiveEntry Service 247
 cluster 84
 configuration file 51
 configure 51
 event display 71
 install 83
 language 69
 log file 72
 process components 48
 setup 48
 statistic information 64
 update 112, 114
 user account 21
 ActiveEntry Service Configurator 52, 154
 module list 53
 module type 54
 validity test 55
 ActiveEntry Service Updater 114, 154, 248
 ActiveEntry tools
 login 161
 update 111
 ActiveEntry Transporter 153
 ActiveEntry Unified Namespace 248
 ActiveEntry.Net Setup Wizard 24
 ALE
 see Application Link Enabling
 Application 248
 Application group 248
 Application Link Enabling 248
 Application role 249
 Application server 249
 Approval method 249
 Approval policy 249

Approval procedure 249
 Approval workflow 250
 Approver 249
 Assignment request 250
 Attestation 250
 Attestion instance 250
 Attestion
 attestor 250
 Authentication module 250
 ADS user 164
 ADS user (dynamic) 164
 ADS user (manual input) 165
 ADS user (manual input/role based) 165
 ADS user (role based) 165
 employee 163
 employee (dynamic) 163
 employee (role based) 163
 LDAP user (dynamic) 166
 system user 162, 268
 tool authenticator 162
 web ADS user 164
 Authorization definition
 see SAP function > Authorization definition
 Authorization editor
 see SAP function > Authorization editor
 Authorization field
 see SAP function > Authorization definition >
 Authorization field
 Authorization objekt
 see SAP function > Authorization objekt
 Autoupdate.log 114

B

Base group 153
 Basegroup 18, 27
 Business role 256

C

Cancelation workflow 251
 Cart 251
 Cart item 251
 Central user administration 252
 Cluster 84
 Cluster resource
 ActiveEntry Service 87
 log file 90
 setup 87, 90
 Combined Log Format 77
 Company resource 252
 Compiling
 database 126
 error message 126
 Configuration
 ComponentDebugMode 69
 DebugMode 69
 HTTPAddress 69

- HTTPPort 69
 - Language 69
 - SSLCertificate 69
 - SSLKey 69
 - UseSSL 69
 - Configuration file 51
 - Configuration parameter 252
 - Configuration Parameter Editor 252
 - Connection
 - JobGenLogDir 75
 - NoReloadBeep 75
 - ObjectLogDir 75
 - SQLLogDir 75
 - Consistency icon 190
 - Contact data
 - system user 161
 - Crypto Configuration 135, 154, 252
 - CUA
 - see Central User Administration
 - CUA status 253
 - Customer 253
 - Customer configuration package 95
 - create 100
- D**
- Data Import 156
 - Data transport
 - database history 101
 - export criterion 101
 - Database
 - compiling 126
 - connection data 126
 - ConnectionProvider 126
 - ConnectionString 126
 - customer name 126
 - customer number 126
 - customer prefix 126
 - database ID 143
 - encryption 135
 - full-text search 141
 - login 159
 - migration 27
 - procedures 39
 - reference database 143
 - single user mode 27, 29
 - update 95
 - version status 29
 - viDynamicGroupCheck 39
 - Database history 100, 101
 - Database login
 - under Microsoft SQL Server 159
 - under Oracle 159
 - Database role
 - basegroup 18
 - Database schedule 39
 - change request cycle 39
 - procedures 39
 - set up 143
 - setup 39
 - VID_CompressJobQueueStats 39
 - VID_DBScheduler 39
 - VID_JobSchedule 39
 - viInventoryHistoryTruncate 39
 - Database schema 253
 - Database search 238
 - Database system
 - Microsoft SQL Server 17
 - Oracle 19
 - Database user
 - Microsoft SQL Server 18
 - Oracle 19
 - DBCompiler 126, 153, 253
 - DBMigrator 29, 153, 253
 - migration 29
 - DBScheduler 39, 253
 - start 245
 - system log 245
 - DBSchedulerWatchDogPlugin 80
 - Interval 80
 - ProviderID 80
 - Default PC 254
 - Default user 254
 - Delegation 254
 - Desktop Management Interface 254
 - Desktop Management Task Force 254
 - Dispatcher
 - IsProxy 73
 - ProxyIntervall 73
 - Distribution model 254
 - Domain Name System 255
 - Driver 255
 - Dynamic Host Configuration Protocol 255
- E**
- Encoding 154
 - Encryption 135
 - encryption file 135
 - encryption information 135
 - key
 - add 135
 - change 135
 - generate 135
 - key file 135
 - PrivateKey 64
 - Encryption file 64, 135
 - Enterprise Resource Planning 255
 - Error log 214
 - EventLogLogWriter 71
 - LogSeverity 71
 - Exception approver 255
- F**
- File

- export 122
 - import 116
 - FileJobDestination 50, 66
 - AutoUpdateSubDirectories 58
 - BackupFiles 58
 - CheckInputIndex 58
 - EventTypes 58
 - Hostname 58
 - InputDirectory 58, 66, 67
 - MaxListCount 58
 - OutputDirectory 58, 67
 - Port 58
 - ProviderID 66, 67
 - SubDirectories 58
 - TimerInterval 58
 - UseEncryption 58
 - FileJobGate 50
 - FileJobProvider 49, 58
 - AutoSubDirectories 58
 - BackupFiles 58
 - CheckInputIndex 58
 - EventTypes 58
 - HostName 58
 - InputDirectory 58
 - MaxListCount 58
 - OutputDirectory 58
 - Port 58
 - SubDirectories 58
 - TimerInterval 58
 - UseEncryption 58
 - FileLogWriter 72
 - HistorySize 72
 - LogLifeTime 72
 - LogSeverity 72
 - OutPutFile 72
 - ParamMaxLength 72
 - Filter
 - adhoc 238
 - permanent 238
 - Filter designer 206
 - First Distribution Server 255
 - FTP Server 49, 60
 - FTP user
 - access rights 60, 62
 - FTPJobDestination 50, 67
 - AutoUpdateSubDirectories 58
 - BackupFiles 58
 - CheckInputIndex 58
 - EventTypes 58
 - Hostname 58
 - InputDirectory 58
 - MaxListCount 58
 - OutputDirectory 58
 - Port 58
 - SubDirectories 58
 - TimerInterval 58
 - UseEncryption 58
 - FTPJobProvider 49, 60
 - AutoSubDirectories 58
 - BackupFiles 58
 - CheckInputIndex 58
 - EventTypes 58
 - FTPPassword 60
 - FTPPort 60
 - FTPServer 60
 - FTPUser 60
 - HostName 58
 - InputDirectory 58
 - MaxListCount 58
 - OutputDirectory 58
 - Port 58
 - SubDirectories 58
 - TimerInterval 58
 - UseEncryption 58
 - Full text search 238
 - Full-text catalog
 - mark database column 142
 - prerequisite 141
 - setup 143
 - update 143
 - Function element
 - see SAP function > authorization definition > Function element
 - Function instance
 - see SAP function > Function instance
- H**
- Hardware inventory
 - conversion script 126
 - HistoryDB 256
 - HistoryDB Manager 157, 256
 - HistoryDB Service 257
 - Hotfix package 95
 - HTTP Server 69
 - HTTPJobDestination 50, 68
 - ChildPort 68
 - ProviderID 68
 - Retries 68
 - RetryDelay 68
 - HTTPJobGate 50
 - HTTPJobProvider 49, 61
 - ParentPort 61
 - ParentServer 61
 - Retries 61
 - RetryDelay 61
 - HTTPLogPlugin
 - log file 77
 - LogFile 77
 - HTTPStatusPlugin 76
 - Hypertext Transfer Protocol (HTTP) 257
- I**
- Inheritance

- discontinuing 205
- VID_DBScheduler 39
- installutil.exe 83
 - calling parameters 83
- Inventory history
 - delete 39
 - storage period 39
- IT Shop 257
- IT Shop structure 257
- Item 262

J

- Job destination 48, 257
 - FTPJobDestination 67
 - HTTPJobDestination 68
- Job gate 48
- Job provider 48, 257
 - FTPJobProvider 60
 - HTTPJobProvider 61
 - MSSQLJobProvider 56
 - OracleJobProvider 57
 - WebServiceJobProvider 61
- Job queue 258
 - queue name 64
 - statistic 39
- Job Queue Info 154
- Job server 258
- Job Server Editor 258
- Job service 258
- Jobdestination
 - FileJobDestination 50, 66
 - FTPJobDestination 50
 - HTTPJobDestination 50
 - JobServiceDestination 50, 69
- Jobgate
 - FileJobGate 50
 - HTTPJobGate 50
- JobGenLogDir 75
- Jobprovider
 - FileJobProvider 49, 58
 - FTPJobProvider 49
 - HTTPJobProvider 49
 - MSSQLJobProvider 49
 - OracleJobProvider 49
 - WebServiceJobProvider 49
- JobQueueInfo 257
- Jobservice.cfg 51
- JobServiceDestination 50, 69
 - ExternalSlots 64
 - InternalSlots 64
 - MaxExternalSlotReuse 64
 - PrivateKey 64
 - ProviderID 64
 - Queue 64
 - RequestTimeout 64
 - StartInterval 64
 - StatisticInterval 64

K

- Key file 135

L

- Language Editor 258
- License Wizard 155, 258
- Lightweight Directory Access Protocol (LDAP) 258
- List Editor 258
- Login
 - database user 159
 - system user 161
- LogWriter
 - EventLogLogWriter 71
 - FileLogWriter 72
- Lotus Notes 258

M

- Machine profile 259
- Machine type 259
- Main Library Server 259
- Managed Information Format 259
- Managed Object 259
- Managed Object Format 259
- Manager 247
 - consistency icon 190
 - controls 195
 - data export 242
 - database search 238
 - document view 193
 - error log 214
 - favorites list 213
 - form help 194
 - help 193
 - language 186
 - multiple editing 210
 - navigation view 188
 - process information pane 217
 - program settings 183
 - quick edit mode 186
 - result list 190
 - search dialog 180
 - simulation mode 228
 - system log 214
 - tasks view 212
 - TimeTrace 225
- Migration
 - initial migration 29
 - migration packet 29
 - migration steps 29
 - under Microsoft SQL Server 29
 - under Oracle 29
 - version status 29
- Migration file
 - common installation part 29
 - product installation part 29

Migration package 95
Mitigating control 259
Module
 configuration 69
 connection 75
 Dispatcher 73
 log writer 70
 Plug-in 76
 process collection 55
 processing and forwarding 63
MSSQLJobProvider 49, 56
 ConnectionString 56
 RequestQueueLimit 56
 ResultQueueLimit 56

N

Namespace Mapping Editor 260
NetBIOS 260
NSNames.ora 146

O

Object
 meta object 169
 multiple editing 210
 user interface 169
Object definition 260
Object Editor 260
ObjectLogDir 75
OracleJobProvider 49, 57
 ConnectionString 57
 RequestQueueLimit 57
 ResultQueueLimit 57
Org level
 see SAP function > org level
Organisation 260

P

Patch 261
PerformanceCounterPlugin 82
 CounterType 82
 PollingInterval 82
Plug-in
 DBSchedulerWatchDogPlugin 80
 RequestWatchDogPlugin 81
Plugin 261
Plugins
 HTTPLogPlugin 77
 HTTPStatusPlugin 76
 PerformanceCounterPlugin 82
 ScheduleCommandPlugin 79
 ShareInfoPlugin 77
Preprocessor condition 261
Process 261
Process component 262
Process components 48, 64
Process Editor 261

Process function 261
Process information
 process information pane 217
Process parameter 261
Process step 262
Production Application Server 262
Profile 262
Profile Editor 262
Profile Scanner 262
Provider client 263
Provider master 263
Provider mode 263
Proxyserver 73

R

Renewal workflow 263
Replication Info 263
Report Editor 156
Request template 263
RequestWatchDogPlugin 81
 Interval 81
 MinRequests 81
Resource 263
Resource type 264
Rights Editor 261
Role
 base group 153
 basegroup 27
 business role 256
 user defined 264
Role class 264
Role type 264
Rolle
 Organisation 260

S

SAM database 264
SAP
 distribution model 254
SAP authorization 265
SAP function 265
 Authorization definition
 Function element 256
 authorization definition 251
 authorization feld 251
 Authorization Editor 251
 authorization objekt 251
 Function instance 256
 org level 260
 SAP menu 265
 transaction 269
 variable set 270
SAP funktion category 265
SAP menu
 see SAP function > SAP menu
SAP R/3 265

- Schedule 265
 - Schedule Editor 265
 - ScheduleCommandPlugin 79
 - Command 79
 - Interval 79
 - LogSeverity 79
 - OutputToLog 79
 - StartCommand 79
 - StopCommand 79
 - Schema Editor 266
 - Schema Extension 155, 265
 - Script
 - compile 126
 - Script Debugger 157
 - Secure Sockets Layer (SSL) 266
 - Server
 - installation 21, 48
 - setup 48
 - Service catalog 266
 - Service category 266
 - Service item 266
 - Service Provisioning Markup Language 266
 - ShareInfoPlugin 77
 - Shelf 266
 - Shelf template 267
 - global 256
 - shopping center template 267
 - special 268
 - Shop 267
 - Shopping cart
 - see Cart
 - Shopping center 267
 - Single user 27, 29
 - Site 267
 - Software Loader 116, 156, 267
 - Software profile 268
 - Software update 110
 - ActiveEntry Service 112, 114
 - ActiveEntry tools 111
 - Autoupdate.log 114
 - software revision.viv 110
 - update.log 110
 - SQL Server Agent
 - database schedule 27
 - SQLLogDir 75
 - Statistic information 78
 - StatisticsPlugin
 - BufferSize 78
 - CollectTo 78
 - Columns 78
 - FileName 78
 - Interval 78
 - Separator 78
 - ShowDates 78
 - ShowHeaders 78
 - StudioProcessor.exe 64
 - System log
 - show 214, 245
 - System role 268
 - System user 268
 - authentication module 268
 - viadmin 34, 35, 162
 - System user ID
 - definition 268
- T**
- Target system 269
 - Target system zone 269
 - Template 269
 - Test Application Server 269
 - Text comparison 269
 - TimeTrace 225
 - TNS alias 159
 - TNSNames.ora 159
 - Transaction
 - see SAP function > Transaction
 - Transport
 - change information 105
 - change label 104
 - database history 100
 - full transport 109
 - schema extension 107
 - select dates 105
 - select objects 108
 - select users 105
 - system configuration 109
 - system file 110
 - Transport package
 - create 100
 - customer configuration package 95
 - export 100
 - export criterion 101
 - hotfix package 95
 - import 95
 - migration package 95
- U**
- UID 270
 - Unattended Setup 270
 - Unified Namespace
 - see ActiveEntry Unified Namespace
 - UNS 248
 - see ActiveEntry Unified Namespace
 - User account
 - ActiveEntry Service 21
 - User Interface Editor 260
 - Users and Permissions Group Editor 270
- V**
- Variable set
 - see SAP function > Variable set
 - Version control 153
 - VI access permissions 270

VI Client 270
VI editing permissions 170
viNetworkService.exe 51
viNetworkService.exe.config 51

W

Web applicaiton
 post 40
Web application 152
 share 40
Web Based Enterprise Management 271
Web Designer 157
WebServiceJobProvider 49, 61
 CheckResponseInterval 61
 Domain 61
 Password 61
 RequestQueueLimit 61
 server name 61
 URL 61
 user 61
Windows Internet Name Service 271
Windows Management Instrumentation 271
Windows NT 271
Workflow Editor 271
Workstation
 setup 25

ActiveEntry
now you can.

