

# **AssetCenter<sup>TM</sup>**

**Version 3.5**

## **Reference Guide: Administration and Advanced Use of AssetCenter**

### **Volume I**

April 07, 2000

ITEM ACT-3.5X-EN-00698



The Infrastructure Management Company<sup>TM</sup>



© Peregrine Systems, Inc., 1999-2000. All Rights Reserved.

Sybase SQL Anywhere Runtime: © Sybase, Inc. 1992–1995; Portions © Rational Systems, Inc. 1992–1994.

Information contained in this document is proprietary to Peregrine Systems, Inc., and may be used or disclosed only with written permission from Peregrine Systems. This manual, or any part thereof, may not be reproduced without the prior written permission of Peregrine Systems, Inc.

This document refers to numerous products by their trade names. In most, if not all cases, these designations are claimed as Trademarks or Registered Trademarks by their respective companies.

**Peregrine Systems, ServiceCenter, AssetCenter, InfraCenter for Workgroups and InfraTools** are trademarks of Peregrine Systems, Inc.

The software described in this manual is supplied under license or nondisclosure agreement. The software may be used or copied only in accordance with the terms of the agreement.

Peregrine Systems reserves the right to modify the information contained in this document without notice.

The software is subject to modification and it is possible that the supplied documentation is not fully coherent with the version that you have. These modifications do not compromise proper understanding of the software. For further information on the most recent modifications, please refer to the **Readme.txt** file.

The names of companies and individuals used in the sample database and in examples in the manuals are fictitious and are intended to illustrate the use of the software. Any resemblance to actual companies or individuals, whether past or present, is purely coincidental.

### **AssetCenter and InfraCenter for Workgroups data integrity**

AssetCenter and InfraCenter for Workgroups are extremely rich in functionality. This richness relies on a complex database structure: The database contains a large number of tables, fields, links and indexes; certain intermediary tables are not displayed by the graphical interface; certain links, fields and indexes are automatically created, deleted or modified by the software.

Only the interfaces designed for AssetCenter and InfraCenter for Workgroups (graphical interface, APIs, import program, Web interface and gateways) are capable of modifying the database with respect to its integrity. **You must never modify the structure and/or the contents of the database by any means other than those intended for use with the software**; such modifications are highly likely to corrupt the database and bring about symptoms such as involuntary loss or modification of data or links, creation of "ghost" links or records, serious error messages, etc. Alterations to the database resulting from manipulations of this type void the guarantee and technical support provided by Peregrine Systems.

**Environments supported by AssetCenter and InfraCenter for Workgroups**

The list of environments supported by AssetCenter and InfraCenter for Workgroups can be found in the manual entitled "Installation and Upgrade Guide". Using AssetCenter or InfraCenter for Workgroups in environments other than those for which it is intended is done at the user's risk. Alterations made to the database resulting from using AssetCenter or InfraCenter for Workgroups in environments other than those for which it is intended void the guarantee and technical support provided by Peregrine Systems.

# Foreword

## ***Presentation of AssetCenter***

AssetCenter is a comprehensive IT infrastructure management system with integrated areas of functionality, based on a shared repository:

Asset Management.

Helpdesk Management

Procurement Management.

Leasing Management.

Cost Management.

Its "integrated" dimension allows you to access all functionality via a unified interface and use a single database, thus avoiding data redundancy.

Its "functional" aspect allows you to adapt the user interface according to your needs. You may simplify the interface by selecting only those areas of functionality you wish to use.

- Objectives of the reference manuals
- These manuals provide detailed, systematic explanations of AssetCenter's functionality.
- They are also available in the form of the on-line help, which has more or less the same contents.
- There is a document that contains a table of contents and an index for all of these manuals.

***Objective of this manual***

This manual is aimed at users and power users of AssetCenter. It explains in detail how to administer the AssetCenter database and user administrator-specific functions.

---

# Contact Peregrine Systems

## World Headquarters

Peregrine Systems, Inc.  
3611 Valley Centre Drive  
San Diego, CA 92130  
USA  
Tel: +1 858 481 5000 or 800 638 5231  
Fax: +1 858 481 1751  
Web: <http://www.peregrine.com>  
  
Customer support:  
Tel: +1 858 794 7402 or 800 960 9998  
Fax: +1 858 794 6028  
Web: <http://support.peregrine.com>  
E-mail: [support@peregrine.com](mailto:support@peregrine.com)  
Open Monday to Friday 5:00 AM to 5:30 PM (PST)

## France, Spain, Greece, and Africa (except South Africa)

Peregrine Systems  
Tour Franklin - La Défense 8  
92042 Paris - La Défense Cedex  
France  
Tel: +33 (0)1 47 73 11 11  
Fax: +33 (0)1 47 73 11 12  
  
Customer support:  
Tel: +33 (0) 800 505 100  
Fax: +33 (0)1 47 73 11 61  
E-mail: [frsupport@peregrine.fr](mailto:frsupport@peregrine.fr)  
Open Monday to Friday 8:00 AM to 6:00 PM (local time)

## Germany and Eastern Europe

Peregrine Systems GmbH  
Bürohaus ATRICOM  
Lyoner Strasse 15

60528 Frankfurt  
Germany  
Tel: +49 (0)(69) 6 77 34-0  
Fax: +49 (0)(69) 66 80 26-26  
  
Customer support:  
Tel: 0800 2773823  
Fax: +49 (0)(69) 66 80 26-26  
E-mail: [psc@peregrine.de](mailto:psc@peregrine.de)  
Open Monday to Friday 8:00 AM to 5:00 PM (local time)

## **United Kingdom**

Peregrine Systems, Ltd.  
Ambassador House  
Paradise Road  
Richmond  
Surrey TW9 1SQ  
UK  
Tel: +44 (0)181 332 9666  
Fax: +44 (0)181 332 9533  
  
Customer support:  
Tel: +44 (0)181 334 5844 or 0800 834 7700  
Fax: +44 (0)181 334 5890  
E-mail: [uksupport@peregrine.com](mailto:uksupport@peregrine.com)  
Open Monday to Friday 8:00 AM to 6:00 PM (local time)

## **Denmark, Norway, Finland and Iceland**

Peregrine Systems A/S  
Naverland 2, 12th fl.  
DK-2600 Glostrup  
Denmark  
Tel: +45 43 46 76 76  
Fax : +45 43 46 76 77  
  
Customer support:  
Tel: +45 77 31 77 76  
Fax: +45 43 46 76 77  
E-mail: [support.nordic@peregrine.com](mailto:support.nordic@peregrine.com)  
Open Monday to Friday 8:30 AM to 4:00 PM (local time)



## **The Netherlands, Belgium, and Luxembourg**

Peregrine Systems BV  
Botnische Golf 9a  
Postbus 244  
3440 AE Woerden  
The Netherlands  
Tel: +31 (0) 348 43 7070  
Fax: +31 (0) 348 43 7080

Customer support:  
Tel: 0800 0230889 (The Netherlands)  
or 0800 74747575 (Belgium and Luxembourg)  
Fax: +31 (0) 348 43 7080  
E-mail: [benelux.support@peregrine.com](mailto:benelux.support@peregrine.com)  
Open Monday to Friday 8:00 AM to 6:00 PM (local time)

## **Singapore**

Peregrine Systems Pte.Ltd  
#03-16  
CINTECH III  
77 Science Park Drive  
Singapore Science Park  
118256  
Singapore  
Tel: +65 778 5505  
Fax: +65 777 3033

## **Italy**

Peregrine Systems, S.r.l.  
Via Monte di Pietà, 21  
I-20121 Milano  
Italy  
Tel: +39 (02) 6556931  
Fax: +39 (02) 65569390

## **Sweden**

Peregrine Systems AB  
Frösundaviks Allé 15, 4th floor  
S-169 70 Solna  
Sweden

Tel: +46 (0)8-655 36 04  
Fax : +46 (0)8-655 26 10

Customer support:  
Tel: +45 77 31 77 76  
Fax: +45 43 46 76 77  
E-mail: [nordic@peregrine.com](mailto:nordic@peregrine.com)  
Open Monday to Friday 8:30 AM to 4:30 PM (local time)

## Japan

Peregrine Systems K.K.  
Level 32, Shinjuku Nomura Building  
1-26-2 Nishi-shinjuku, Shinjuku-ku  
Tokyo 163-0532  
Japan

Tel: +81 (3) 5322-1350  
Fax: +81 (3) 5322-1352

Customer support:  
Tel: +81 (3) 5322 1350  
Fax: +81 (3) 5322 1352  
E-mail: [glipper@peregrine.com](mailto:glipper@peregrine.com)

---

## Conventions

The following notation is used for commands:

[ ]	Square brackets denote an optional parameter. Do not type them in your command. Exception: In BASIC scripts, square brackets are used to denote the data access path and must be included in the script: <b>[Link.Link.Field]</b>
< >	Brackets denote a parameter in plain language. Do not type them. Substitute the text with the appropriate information.
{ }	Curly brackets denote a series of parameters. Only one of these parameters may be used. Do not type these curly brackets in your command.
	A pipe is used to separate a series of parameters contained within curly brackets.
*	An asterisk added to the right of square brackets means that the formula shown can be repeated several times.

The following text formats have given meanings:

<b>Fixed width characters</b>	DOS command.
<div>Example</div>	Example of code or command.
...	Code or command omitted.
<b>Object name</b>	The names of fields, tabs, menus and files are shown in bold.
<div>Note</div>	Important note.

---

## Send us your comments

We want to deliver the most accurate documentation possible.

Any comments would be greatly appreciated.

Send any remarks to **documentation@peregrine.com**.



# Table of contents

---

<b>Chapter 1 - Database options</b>	<b>1</b>
-------------------------------------	----------

---

<b>Chapter 2 - Structure of the AssetCenter database</b>	<b>5</b>
Foreword	5
Definition of a database	6
Structure of the database.txt and tables.txt files	7
Description of the tables	8
Identifying AssetCenter tables	8
Table description strings	9
Description of fields	10
Identifying AssetCenter fields	10
Field types and data entry formats	11
dtLastModif fields	15
"FullName" fields	16
Description of links	17
Identifying of links	17
Type of a link	18
Cardinality of a link	18
Intermediary tables	19
Contextual links	21
Description of indexes	21
Identifying indexes	21
Uniqueness of indexed field values	21
Customizing the database	22
Customizing tables	23
Customizing fields and links	23
Customizing the indexes	24
Default values for fields and links	24
Counters in field default values	25
Errors in the description of default values	26
HTML tags recognized in help on fields and links	26

---

<b>Chapter 3 - Creating, customizing and describing the AssetCenter database</b>	<b>29</b>
Using AssetCenter Database Administrator	30
Launching AssetCenter Database Administrator	31

Description of the File menu	31
Description of the Edit menu in AssetCenter Database Administrator	34
Description of the Action menu in AssetCenter Database Administrator	34
Creating a database	41
Customizing a database	41
Repairing a database	47
Description of an AssetCenter database	47
Database description parameters	48
Syntax for description templates	50
Fixed text	50
Comments	51
Including another template	51
Browsing, sorting and filtering components	51
@if...\$else...\$elseif...\$endif conditions	54
Functions available in description templates	54
Processing the value of a property using a function defined in a template	58
Deleting the end-of-paragraph mark	59
Counting the number of browsed components	60
Defining a global variable for the template	60
Information on certain database description parameters	61
Database instance	62
Table instance	62
Field instance	63
Link instance	67
Index instance	68
Script instance	69
Global variables	70
Values generated during the export process	71
<hr/>	
<b>Chapter 4 - Managing itemized lists</b>	<b>73</b>
Free itemized lists	73
System itemized lists	75
<hr/>	
<b>Chapter 5 - How the history function works</b>	<b>77</b>
<hr/>	
<b>Chapter 6 - Managing user access to AssetCenter databases</b>	<b>83</b>
Importance and overview of database access rights	84
How to ensure data security and confidentiality	85
Definitions	85
Definition of a user profile	86
Definition of a user right	86
Definition of an access restriction	86
Defining access conditions	86
Defining user profiles	86
Defining user rights	87
Defining access restrictions	89
Defining AssetCenter users	91
Managing user connections	92
Database access types	92
How connection slots work	94

Destroying connection slots	94
Managing passwords	96
Password for the Admin login	96
Modifying the password for the "Admin" login	97
User password	97
Lost passwords	97
<hr/>	
<b>Chapter 7 - Writing queries in AQL</b>	<b>99</b>
Presentation	99
Queries in AssetCenter	99
AQL	100
Recommendations for writing AQL queries	102
Presentation of AQL joins	103
Reason for and usefulness of primary key 0 records	104
Usage of NULL	106
Self	106
CurrentUser	107
System itemized-lists	108
Hierarchic tables	109
Simplified AQL notation	110
Sorts and indexes	111
The query editor	114
Overview	114
Accessing the query editor	115
Creating a query using the query editor	116
Fields used in queries	118
Writing expressions	118
Constants	119
AQL syntax	120
Conventions	120
Syntax of queries	121
FROM clause- Tables involved in a query	122
Elements of a query	123
WHERE clause	128
GROUP BY clause	129
HAVING clause	130
ORDER BY clause	130
AQL function reference	131
"Aggregate" type AQL functions	131
"String" type AQL functions	132
"Date" type AQL functions	133
"Numeric" type AQL functions	135
"Test" type AQL functions	136
Examples of queries	136
<hr/>	
<b>Chapter 8 - Editing forms</b>	<b>141</b>
Definition of a form	141
Creating forms	141
Editing forms and objects	142
Properties of object in forms	144
Design grid	147
Form page setup	147

How to easily create regular reports	148
<hr/>	
<b>Chapter 9 - Editing reports</b>	<b>151</b>
Operation and installation of the report generator	151
Detail of a report	152
How to modify a Seagate Crystal report	153
Crystal Reports statistics	153
Creating a detail report	154
<hr/>	
<b>Chapter 10 - Read-only access to the database using the AssetCenter ODBC driver</b>	<b>157</b>
Overview of access to the AssetCenter database	157
Example: Creating a report under Crystal Reports with the ODBC driver	160
<hr/>	
<b>Chapter 11 - Defining actions</b>	<b>161</b>
Definition of an action	161
Creating an action	162
Types of actions	162
General method	164
Populating the DDE tab	166
Populating the <i>Messaging</i> tab	167
Examples of actions	169
Example of "Executable" type action	170
Example of "DDE" type action	170
Example of Messaging type action	174
Example of a Script type action	174
Using variables	176
Specificities of the Sybase SQL Anywhere database engine	176
Testing an action	177
Executing an action	177
<hr/>	
<b>Chapter 12 - Managing deadlines with AssetCenter Server</b>	<b>179</b>
Overview of AssetCenter Server	179
Executing AssetCenter Server	180
Main screen of AssetCenter Server	183
General options of AssetCenter Server	184
Modules monitored by AssetCenter Server	186
Selecting the modules to be monitored	187
Verification schedules	187
History lines monitored by AssetCenter Server	189
Stock lines monitored by AssetCenter Server	189
Alarms monitored by AssetCenter Server	190
Rent calculated by AssetCenter Server	192
Loss values updated by AssetCenter Server	194
Split operations performed for expense lines by AssetCenter Server	194
Verifying database server time zone	196
Signaling presence to database server	196
Searching for new workflow execution groups	197
Executing workflow rules	197
Activating the verification immediately	199



Administrating AssetCenter Server by the Web	199
Launching AssetCenter Server as an NT service	200
Accessing the AssetCenter Server service via the Web	201
Driving the AssetCenter Server service via the Web	201
<b>Chapter 13 - Managing messages</b>	<b>205</b>
Overview of messaging	205
Prerequisites for sending messages via an external messaging system	207
Configuring AssetCenter to use messaging systems	207
Populating employee details in order to use an external messaging system	208
Creating messages	211
Common connection problems	211
<b>Chapter 14 - Workflow</b>	<b>215</b>
Definitions	215
Workflow	216
Workflow activity	217
Workflow event	217
Workflow transition	217
Workflow task	218
Workflow activity assignee	218
Definition of a workflow execution group	218
General overview	218
Main tables involved in workflow	221
Using the graphical workflow editor	222
Activities	223
Events	223
Transitions	224
Other functionality	225
How to implement workflow	225
Example of workflow used in request approval	225
Aim	226
Preliminary steps	227
Creating activities	233
Creating the start event	237
Creating transitions	238
Example of activating a workflow instance	239
The context of a workflow instance	243
Defining the context of a workflow instance	243
Object referenced by a workflow instance	244
Limiting workflow instances in progress for a given object	244
Workflow roles	245
Activities	247
Question type activities	247
User action type activity	248
Automatic action type activity	249
Test/ Script type activities	250
Start activity	251
Activity templates	251
Triggering activities	251
Tasks	252
Creating tasks	252

Automatic action or test/ script type activity	253
Displaying the list of tasks in progress	253
Performing a user task	254
Assigning user tasks	254
Administrating a workflow task	255
Events	255
System events	255
Alarm events	256
User events	256
General conditions of activation	258
Processing of events	259
Application: Implementing a synchronous workflow scheme	262
Terminal event	264
Workflow transitions	265
Workflow alarms and time limits	265
Time limit	266
Workflow alarms	266
Workflow execution groups	267
Workflow tracking	268
<hr/>	
<b>Chapter 15 - Importing data</b>	<b>269</b>
Overview of importing data	270
Recommendations	271
Constraints to be respected when importing data into a field	272
Importing text files or an ODBC database	276
Before importing text files	277
Step 1: Select the text files or ODBC database to be imported	278
Step 2: Define the parsing of files or tables to be imported	279
Step 3: Describe the fields to be imported	283
Step 4: Map source fields to target fields in the AssetCenter database	285
Step 4: Map each text file or source table to a target table	286
Step 4: Map the fields to be imported to fields in the AssetCenter database	286
Step 4: Add additional calculated fields in the source file	292
Step 4: Special cases	297
Step 4: Examples of using keys	297
Step 5: Configuring the transfer	306
Step 6: Data transfer	308
Importing an AssetCenter database	309
Saving and executing an import script	309
Executing an import script from a 32-bit command prompt	311
<hr/>	
<b>Chapter 16 - Exporting data and managing SQL views</b>	<b>315</b>
Definitions	315
Exporting data from the AssetCenter database	316
Managing SQL views in the AssetCenter database	318
Recommendations	318
Defining an export script	319
Methodology	319
Defining export queries	320
Output format of an export script	322
Actions concerning SQL views	323
Executing an export script	324

Executing an export script from AssetCenter Export	324
Executing an export script from DOS	325
<hr/>	
<b>Chapter 17 - Using scripts</b>	<b>327</b>
Definition of a script	327
Applications of scripts	329
Introduction to functions	330
Definition of a function	330
Built-in functions and programmable functions	330
Function and parameter types	332
Classifying Basic functions	334
First steps in writing scripts	334
Example scenario	335
Step 1: Create the feature "Tutorial"	335
Step 2: Open the edit window	336
Step 3: Analyze and define the algorithm	336
Step 4: Draw up the Basic script	337
Step 5: Test the BASIC script	337
Tips and warnings	338
Precautions when using programmable functions	338
Format for "Date+Time" constants in scripts	338
Format for "Duration" constants in scripts	339
Read and write access to a system itemized list	339
The "CurrentUser" virtual link	341
Commenting a Basic script	342
Triggering an error message	343
First example	343
Description of the problem	344
Step 1: Analyze and define the algorithm	344
Step 2: Draw up the BASIC script	346
Step 3: Test the BASIC script	346
Second example	346
Description of the problem	347
Step 1: Analyze and define the algorithm	347
Step 2: Draw up the BASIC script	348
Step 3: Test the BASIC script	348
<hr/>	
<b>Chapter 18 - Managing calendars</b>	<b>349</b>
Overview of calendars	349
Impact of calendars on certain areas of functionality	350
Methodology used to create a calendar	350
Description of how to create a calendar	351
Entering general information	351
Populating the <i>Timetable</i> tab	352
Populating the <i>Exceptions</i> tab	353
Checking the calendar	356
<hr/>	
<b>Chapter 19 - Managing time zones</b>	<b>359</b>
Why manage time zones?	359
Implementing time zones	360
Creating time zones	361

Managing a time zone	361
Format of Daylight time field	362
Values of the <Year> argument	362
Values of the <DaylightInfo> argument	362
Example	365
Managing time zones in AssetCenter Server	367
Consequences for various operations	369
Creating the database	369
Connecting to a database	370
Import and Export	371
 <b>Chapter 20 - Using AssetCenter as a DDE server</b>	 <b>373</b>
Definition of a DDE server	374
DDE call mechanisms	374
DDE service	374
DDE topic	374
DDE command	375
Introduction to DDE commands	375
Global commands	376
Connect(<Cnx>, <User>, <Password>)	376
Disconnect()	377
ExecuteAction(<ActionName>)	378
ListAllTables([Mask])	378
ListOpenTables([Mask])	379
OpenView(<ViewName>)	380
Commands associated with a table	381
OpenTable(<Table>)	381
CloseTable(<Table>)	382
<Table>.GetRecordCount()	382
<Table>.SetViewMode(<Mode>)	383
<Table>.SetRecordMode(<Mode>)	384
<Table>.ListAllFields([Mask])	385
<Table>.ListAllLinks([Mask])	386
<Table>.SetFilter(<Condition>)	387
<Table>.SetSelection(<Condition>)	388
<Table>.GetSelectionId()	388
Commands associated with a table and a field or a link	389
<Table>:<Objet>.AddLink()	389
<Table>:<Object>.GetValue()	390
<Table>:<Object>.Hilight()	391
<Table>:<Object>.RemoveLink()	392
<Table>:<Object>.SetFocus()	392
<Table>:<Object>.SetValue(<Value>)	393
<Table>:<Link>.SetValueWhere(<Condition>)	394
<Table>:<Object>.Show()	395
Introduction to Examples of DDE calls	396
First scenario: AssetCenter internal DDE calls	396
Answering the call	397
Entering data	397
Creating the feature	398
Second scenario: DDE calls from Excel 97	398
Description of the macro	399
Source code of macro	399

Third scenario: DDE calls from Visual Basic 5.0	400
Before starting	400
Executing the program	401
Source code of the program	402
<hr/> <b>Chapter 21 - Calculated fields</b>	<b>403</b>
Definition of a calculated field	403
Usefulness of calculated fields	404
Creating a calculated field	404
Introduction	405
Methodology used to create calculated fields	406
Using calculated fields	409
<hr/> <b>Chapter 22 - Introduction to wizards</b>	<b>411</b>
Definition of a wizard	411
Who are wizards aimed at?	412
Classification of wizards	412
Wizards that exchange information with the AssetCenter database	412
Independent wizards	413
<hr/> <b>Chapter 23 - Creating wizards</b>	<b>415</b>
Conventions used	417
Definitions	417
Structural model	419
Model of a wizard page	421
General points	422
Generic structure and syntax	423
Properties of a node	423
Sequencing wizards	429
Basic functions	430
Definition of a Root node	430
Syntax of a Root node	430
Properties of a Root node	431
Sub-nodes of a Root node	434
Definition of a Page node	434
Syntax of a Page node	435
Properties of a Page node	435
Sub-nodes of a Page node	437
Definition of a Transition node	437
Syntax of a Transition node	437
Properties of a Transition node	438
Specificities of a Transition node	439
Definition of a Finish node	440
Definition of a Start node	440
Definition of Long and String nodes	441
Definition of a Control node	441
General syntax of a Control node	442
Types of controls and associated properties	442
Common properties	443


The CheckBox control	445
The ComboBox control	445
The ListBox control	446
The Label control	449
The OptionButtons control	449
The ProgressBar control	450
The CommandButton control	451
The DBListBox control	451
The DBQueryBox control	454
The DBEdit control	456
The DBTable control	457
The DBPath control	457
The LinkEdit control	458
The TextBox control	458
Example of creating a wizard	459
Step #1: Analyze your needs	459
Step #2: Define how the wizard is organized	460
Step #3: Transcribe the structure of the wizard using the scripting language	462
Using the graphical editor	465
Overview of the interface	465
Creating a new node	467
Editing the properties of a node	468
Compiling, executing and debugging a wizard	468
Frequently asked questions	469
<hr/>	
<b>Chapter 24 - Tuning AssetCenter for use in a WAN environment</b>	<b>473</b>
<hr/>	
<b>Chapter 25 - .Ini files</b>	<b>477</b>
List of available .ini files	477
Modifying .ini files	478


# Chapter 1 - Database options

When you are connected as administrator, the **Tools/ Administration/ Database options** menu item allows you to configure certain options inherent to AssetCenter and/or the database. The options are organized by section:

Section	Name of the option	Description	Example(s) of values	Editable
Agent	Deactivate history during import	If this option is set to "Yes" history lines associated with records are not imported.	No	Yes
Wizards	Configuration script	Wizard configuration script. This option defines the wizard configuration parameters and provides some useful functions.	N/A	Yes
Authorization	String	Authorization string of the software.	This string is provided by Peregrine Systems.	No
Authorization	AssetCenter Server identifier	MAC address of the machine on which AssetCenter Server is running.	This value is defined automatically by AssetCenter Server and may not be modified by the user.	No
Authorization	Key	Authorization key of the software.	This string is provided by Peregrine Systems together with your user license.	No
Feature	Deactivate consolidation of features while modifying a feature value	If this option is set to "Yes", the value of a consolidated feature will not be recalculated when the value of at least one of the features making it up changes.	No	Yes
Access control	Slot update delay in minutes	Interval at which the client updates the login slot.	10	Yes
Access control	Slot timeout in seconds	Timeout period for the database connection. This timeout is only used if the automatic disconnection option is activated (value "Yes").	1800	Yes
Access control	Automatic disconnection	Activation of the automatic disconnection option.	Yes	Yes



Section	Name of the option	Description	Example(s) of values	Editable
Time zone	Server	Time zone of the server. The expected value is a numeric value specifying the difference with reference to GMT. Warning: An empty value means that no time zone is used. "0" defines the time zone as GMT.	4 (time zone GMT+4)	Yes
Time zone	Data	Time zone of data in the database. The expected value is a numeric value specifying the difference with reference to GMT. Warning: An empty value means that no time zone is used. "0" defines the time zone as GMT.	-2 (time zone GMT-2)	Yes
Remote control	Use the default action	If this option is set to <b>Yes</b> , AssetCenter uses the connection parameters defined in the <b>Action par défaut</b> option for controlling an asset remotely.  If this option is set to <b>Non</b> , AssetCenter uses the action you have defined in the <b>Customized action</b> option.	Yes	Yes
Remote control	Default action	This option defines the connection parameters for Remote Management. You can start this application via the <b>Remote control</b> contextual menu in the table of assets, via the <b>Repository\ Assets\ Remote control</b> menu item, or using the corresponding icon:  .	RLOADER.EXE /COM=TCP /TCPPH=[fv_tcpip] /TCPP=1234 /LogID= /title=[AssetTag]	Yes

Section	Name of the option	Description	Example(s) of values	Editable
Remote control	Customized action	If you have set the <b>Use the default action</b> to <b>No</b> , enter the name of the action here (Value of the <b>Name</b> field in the <b>Actions</b> table) that you have created in AssetCenter for remotely controlling assets using Remote Management. You can start this application via the <b>Remote control</b> contextual menu in the table of assets, via the <b>Repository\ Assets\ Remote control</b> menu item, or using the corresponding icon:  .	Remote Management : Nom de l'action créée dans la table <b>Actions</b> , (nom SQL : amAction) d'AssetCenter	Yes
Leasing	Authorize automatic LRF calculation	If this option is set to "Yes", AssetCenter will automatically evaluate the LRF for rent payments at the asset and the contract levels.  If this option is set to "No", AssetCenter will not automatically calculate the LRF. However, if $\text{Rent} = \text{LRF} * \text{Total}$ is not verified, AssetCenter will set the LRF to "0" and inform the user.	Yes	No

The edit screen displays the attributed values for each option, and the minimum and maximum values for the timespan values. You can only modify the "Current value" of the option. Values displayed in black cannot be edited. Specific controls are intended to help you edit the values of options (drop-down list for a "Yes/No" value, text control for entering a text string)

# Chapter 2 - Structure of the AssetCenter database

This section provides information required to understand the description of the database.

- Foreword
- Definition of a database
- Structure of the database.txt and tables.txt files
- Description of the tables
- Description of fields
- Description of links
- Description of indexes
- Customizing the database

Warning: You should never delete or modify a unique index, nor directly write in the database using external tools (via INSERT, DELETE, UPDATE statements or triggers). For the latter, we suggest you use the AssetCenter APIs.

---

## Foreword

To import data, access the database using external tools, write queries, etc., you need to understand the structure of the database. You will need information such as: the names of the fields, their maximum length, format, and whether or not their values must be unique.

There are several ways to obtain a description of the AssetCenter database structure:

- **Database.txt** and **Tables.txt** files: they contain the complete structure of the database. These files are located in the **Infos** sub-directory in the AssetCenter installation directory.

These files describe the default database structure. It does not include any customization you may have performed.

In order for these description files to reflect the customization of your database, use the AssetCenter Database Administrator software with a connection to your database.

- AssetCenter Database Administrator program: it allows you to freely create description files of the AssetCenter database (tables, fields, links and indexes).

It makes use of:

- ❖ A description file of the AssetCenter database (file with the **.dbb** extension) or a connection to the AssetCenter database.
- ❖ A template (file with the **.tpl** extension) that describes which information should be generated. We provide you with standard templates, and you can create your own templates. Sophisticated templates allow you to create files in "RTF" or "HTML" format.

Among the templates that are supplied with the standard version of AssetCenter, one of them, "dbdict.tpl", allows you to export the customization data (including information on features, calculated fields, configuration scripts, etc.) from your database to a standard text file. Used with a source control tool, this description file can be useful in order to track customizations made to the database.

- The AssetCenter program.

---

## Definition of a database

An AssetCenter database is a group of files containing all the information on the assets you are managing. For simple installations, these files are all located in the same directory, either on the local disk drive or on a network file server.

AssetCenter comes with a demonstration database.

AssetCenter provides the ability to create several databases and to open one at a time per user session. Several user workstations can be simultaneously connected to the same database. The program uses a transactional method to update data in the database. This method employs a sophisticated mechanism to manage updates and to ensure that they are carried out in a secure and optimized fashion.

---

## Structure of the **database.txt** and **tables.txt** files

The database structure is described in the **Database.txt** and **Tables.txt** files located in the **Infos** sub-directory of the AssetCenter installation directory.

These files were created using the AssetCenter Database Administrator software and the **Dbase.tpl** and **Tables.tpl** templates. These templates are located in the **Infos** sub-directory of the AssetCenter installation directory.

Here is the format of these files:

- Type: text.
- Separator: tab.
- Character set: ANSI (Windows).

The easiest way to view these files is to open them in a spreadsheet.

We have chosen to include as much information as possible in these files. You may choose to hide or delete the information you do not require.

The **Database.txt** file has the following structure:

- One line per field, link or index.
- The tables are sorted by their SQL name.
- For each table, the following information appears in succession:
  - ❖ Fields,
  - ❖ Links,
  - ❖ Indexes.
- Fields, links and indexes are sorted by their SQL name.
- One data item per column. The first line in the table indicates the template parameters that were used to generate the information. The second line includes a clear description of the type of information.

The structure of the **Tables.txt** file is very simple:

- One line per table.
- Tables are sorted by the SQL table name.
- One data item per column. The first line in the table indicates the template parameters that were used to generate the information. The second line includes a clear description of the type of information.

For information on the parameters used to generate the Database.txt and Tables.txt files, refer to the manual entitled "**Reference Guide: Administration and Advanced Use**", chapter "Creating, customizing and describing the AssetCenter database", section "Description of an AssetCenter database".

Note: the SQL names identify the tables, fields, links and indexes. They are the same for all the various language versions of AssetCenter.

---

## Description of the tables

This section describes the tables in the AssetCenter database:

- Identification of AssetCenter tables
- String describing the AssetCenter tables

### Identifying AssetCenter tables

Each table in the AssetCenter database is described by:

- a "SQL name". The SQL names of the AssetCenter tables are in English, and have the "am" prefix. They are the same for all the various language versions of AssetCenter.
- a "Label": this is the name of the table as it is displayed in AssetCenter.

It is used when AssetCenter has little space available to display the table name (tree view of the database in filters and queries, error messages, menu names, etc.).

- A "Description": this description is used when AssetCenter has sufficient space available to display the entire description (status bar, for example).

## Table description strings

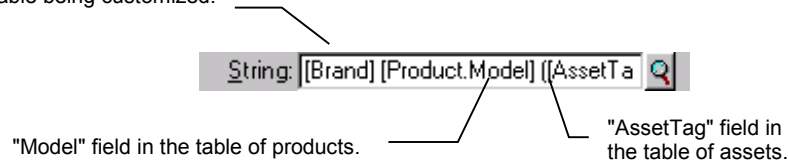
**Use AssetCenter Database Administrator *to define the description strings used for the tables in AssetCenter.***

The description string of an AssetCenter table defines the information displayed to describe records in the drop down lists allowing you to select linked records. It also defines the name that appears in the title bar of the window that displays the records.

The description string can contain the following elements:

- field SQL names, between brackets and square brackets.
- text strings without text delimiters.
- links at 1 or several levels.

The "Brand" field in the table of assets. You do not indicate the table name, because the field is part of the table being customized.



### *Example of customizing the table of assets*

In this table, the title of the detail window for an asset whose brand is "Asus", model is "AsusLX512" model and asset tag is "1" will be:

Detail of asset 'Asus AsusLX512 (1)'
--------------------------------------

When a drop-down list is displayed in list mode, it is sorted in ascending order, except if a filter is applied to the list by AssetCenter.

On the other hand, when a list is displayed in tree view, records are sorted in ascending order according to the "Full name". The "Full name" is a field used in hierarchical tables. Its SQL name is "FullName"

---

## Description of fields

This section describes the fields in the AssetCenter database:

- Identification of AssetCenter fields
- Field types and data entry formats
- dtLastModif fields
- FullName fields

### Identifying AssetCenter fields

Each field in an AssetCenter table is described by:

- A "SQL name". SQL names are in English. They are the same for all the various language versions of AssetCenter. The SQL names are prefixed according to the data type of the field:

Prefix	Data type of the field
"b"	Boolean
"d"	Date
"dt"	Date+Time
"l"	Long integer
"m"	Monetary
"p"	Percentage
"se"	System enumeration (system itemized list)
"ts"	Duration (timespan)
"mem"	Comment
None	Character string

- A "Label". This is the description used when AssetCenter has little space available to display the field name (detail screens, tree view description used when creating queries, etc.).
- A "Description", used to describe the field in the AssetCenter status bar.



## Field types and data entry formats

This section lists:

- The type of AssetCenter fields.
- The data entry formats and data types for AssetCenter fields.

To view the type of a field, its data entry format and the type of its data, you can:

- Use the **Type** field in AssetCenter Database Administrator.
- Read the **Database.txt** file. (The **Field Type** column).
- Display the context-sensitive help for that field.

### Type of a field ("Type")

Gives the field's storage format.

Value given in <i>Field Type</i> column in <i>Database.txt</i> file	Value in <i>Type field</i> in AssetCenter Database Administrator	Description
BYTE	Integer (8 bit)	Integer from -128 to +127.
SHORT	Integer (16 bit)	Integer from -32,768 to +32,767.
LONG	Integer (32 bit)	Integer from -2,147,483,647 to +2,147,483,646.
FLOAT	Floating point number	4 byte floating point number.
DOUBLE	Double-precision number	8 byte floating point number
STRING	Text	Limited length text field. All characters are accepted.
BLOB	Binary field	Variable length binary field (or BLOB Binary large object) is used to store images and forms, for example, without size restriction.
MEMO	Long text field	Variable length text field (comments, for example). Certain DBMSs have limitations concerning these fields. Oracle, for example, does not allow you to sort on this type of field.
DATE+TIME	Date and time	Date and time.
DATE	Date	"Date" format field only (no time).
TIME	Time	"Time" format field only (no date).

### Data entry format and data type

The data entry format and the data type are indicated by:

- The value of the **User type.** in AssetCenter Database Administrator: by default, this value comes from the **Type** field.
- The value displayed for the field in the **Field data display and entry type** column in the **Database.txt** file.

You can nevertheless indicate the data entry type when it is verified:

Value given in <i>Field data display and entry type</i> column in <i>database.txt</i> file	Value in <i>User type field in AssetCenter Database Administrator</i>	Description
Default	Default	The information is displayed and entered just like it is stored in the database, i.e. depending on the field's "Type".
Numerical	Number	Number.
Yes/No	Boolean (Yes/No)	Boolean.
Money	Monetary	Monetary values.
Date	Date	This field can only contain "date" type values.
Date+Time	Date+Time	A field with this data entry format contains "Date+Time" type values.
N/A	Time	This field may only contain "time" type values
Timespan	Duration	"Duration" type values. The authorized and default units are defined by the "UserTypeFormat" property.
System itemized list	System itemized list	The value for this type of field must be selected from a "System" itemized list. This is called a "system" itemized list because the values are defined by the software and cannot be customized by the user.
Custom itemized list	Itemized list	The value for this type of field is selected from an itemized list that may be customized by the user.
Percentage	Percentage	A field with this format contains percentages. This format displays values with two decimal places.
N/A	Feature value	Reserved. Do not use.
N/A	BASIC script	Reserved. Do not use.
Table or field SQL	Table or field name	SQL name of a table or a field.

Value given in <i>Field data display and entry type</i> column in <i>database.txt</i> file	Value in <i>User type</i> field in AssetCenter Database Administrator	Description
name		
N/A	Unknown	Any other type.

Additional information on the format of fields is defined if the **User type** field in AssetCenter Database Administrator is set to one of the following values:

Value of <i>Type util.</i> field in AssetCenter Database Administrator	Additional information
Time span	<p>Display format.</p> <p>Syntax:</p> <p>%U<sub>1</sub>[l][d][%U<sub>2</sub>[l][d]]...[%U<sub>n</sub>[l][d]]</p> <p>where U<sub>x</sub> is one of the following units:</p> <ul style="list-style-type: none"> <li>❖ Y: year</li> <li>❖ M: month</li> <li>❖ D: day</li> <li>❖ H: hour</li> <li>❖ N: minute</li> <li>❖ S: second</li> </ul> <p>The optional "d" parameter specifies the default unit to be applied if no unit is entered. This parameter is assigned to one unit only.</p> <p>The optional "l" parameter specifies that the duration should be displayed in "long" format (e.g. 1998 is the long format of 98).</p> <p>For example:</p> <ul style="list-style-type: none"> <li>❖ "+HM,...H: displays hours then minutes. By default, you enter hours.</li> <li>❖ "%YI%MId%DI": displays years, then months, then days.</li> </ul>
System itemized list	Values taken by the itemized list.
Itemized list	Name of the itemized list.
Object	Used internally by AssetCenter.

## dtLastModif fields

The field whose SQL name is dtLastModif exists for all tables in the AssetCenter database.

- SQL name: "dtLastModif".
- Short description: "Modified on".
- Long description: "Modification date".

This field is updated each time a record is modified in the AssetCenter database, whether this is via the user-interface or an import operation. It

indicates the date of modification or creation of the record. If you import a value in this field, that value is used instead of the actual import date.

## "FullName" fields

The "FullName" is a field in the hierarchical tables. Its SQL name is "FullName".

### Structure of the "FullName" field

For each record in a hierarchical table, the "FullName" field stores the value of a record field, preceded by a hierarchy made up of the values of the fields of the parent records to which the record is linked, up to the root.

Values are separated by the "/" character, without spaces.

This character also appears at the start and end of the hierarchy.

Examples:

- Location table: the full name of a location consists of the name of the location preceded by the name(s) of the parent location(s).

E.g.: "/USA/LA/San Mateo site/".

- Departments and employees table: the full name of an employee consists of the employee's surname, first name and ID, preceded by the name(s) of the parent record(s).

E.g.: "/Sales/Telemarketing/Colombo,Gerald,P223/".

- Table of assets: the full name of an asset consists of its asset tag preceded by the asset tags of the parent asset(s).

E.g.: "/P123/DD456/CM0125/".

**Warning:** You cannot directly write in the "FullName" fields. They are entirely managed by AssetCenter.

### Special case

If one of the values included in the value of a "FullName" field contains the "/" character, this character is replaced by "-".

Example: for the departments and employees, if a department name is "Sales/Marketing", the "FullName" field for its components will have the following format: "/A.../Sales-Marketing/B.../".

---

## Description of links

This section describes the links in the AssetCenter database:

- Identification of links
- Type of a link
- Cardinality of a link
- Intermediary tables
- Contextual links

### Identifying of links

A link is identified by:

- Its "SQL name". SQL names are the same for all the various language versions of AssetCenter.
- Its "Label".
- Its "Description".

Example of the **Location** (SQL name: Location) link from the table of assets:

- The SQL name of the source field in the source table (table of assets) is "lLocaId".
- The SQL name of the target table (table of locations) is "amLocation".
- The SQL name of the target field in the target table (table of locations) is "lLocaId".
- The "database.txt" file indicates that the link's cardinality is "1": a given asset has one location only.

This link may be understood as: "source field = target field".

---

Warning: the label and description of the link are different from the label and description of the target table because there may be several links between AssetCenter tables. E.g., between the table of assets and the table of departments and employees, the link whose label is **User** (SQL

name: User) defines the asset's user, and the link whose label is **Supervisor** (SQL name: Supervisor) defines the asset's supervisor.

## Type of a link

The following table lists the various types of existing links:

Type	Description
Normal	If the source record is deleted, the link is deleted and the references to the source record in the target records are emptied.
Own	If the source record is deleted, the link's target records are deleted.
Define	You cannot delete a source record if it is linked to target records.
Neutral	If the source record is deleted, the link is deleted. There is no information to update in the target records.
Copy	When the source record is duplicated, the links of this type are also duplicated.
Owncopy	The same meaning as "Own" and "Copy" links.

The following table lists the nature of the information stored by links:

Nature of the information	Description
Normal	The link stores something different from what is stored by the other "UserType" links.
Comment	The link stores a comment field.
Image	The link stores an image.
History	The link stores history.
Feature value	The link stores a feature value.

## Cardinality of a link

We have defined two types of links from a given table A in AssetCenter:

- "1-->1" links: A record in table A can only be linked to a single record in table B. For example, the table of employees is linked to the table of locations via a "1-->1" link: An employee can only be associated with a single location.
- "1-->N" links: A record in table A can be linked to several records in table B. For example, the table of contracts is linked to the table of assets via a "1-->N" link: A contract can concern several assets.



Important note: In theory, there are three types of logical links between tables in a database:

- 1 links: A record in table A can only be linked to a single record in table B, and vice-versa. A 1 link between two tables in AssetCenter is represented by two "1->1" links.
- n links: A record in table A can be linked to several records in table B, but a record in table B is only linked to a single record in table A. A n link between two tables in AssetCenter is represented by a "1->1" link and a "1->N" link.
- n-n links: A record in table A can be linked to several records in table B, and vice-versa. A n-n link between two tables in AssetCenter is represented by two "1->N" links.

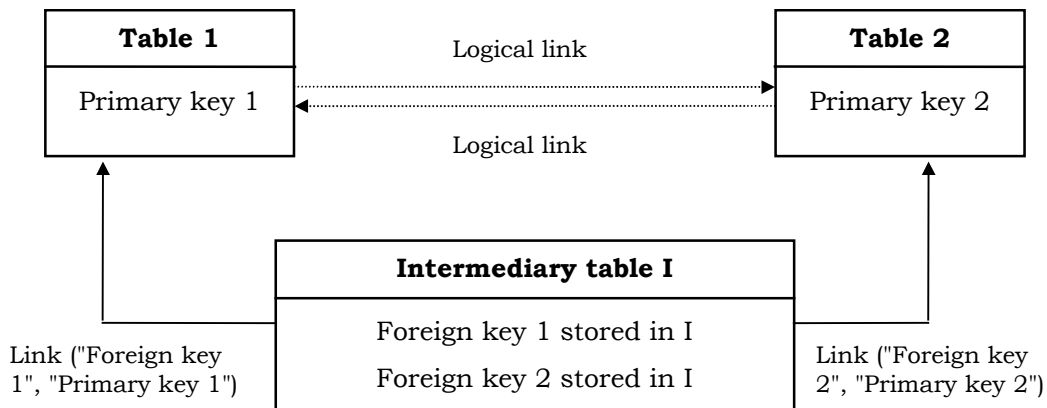
## Intermediary tables

Intermediary tables are only used in the case of n-n cardinality logical links.

They do not appear in the AssetCenter interface, which only shows the logical links between tables.

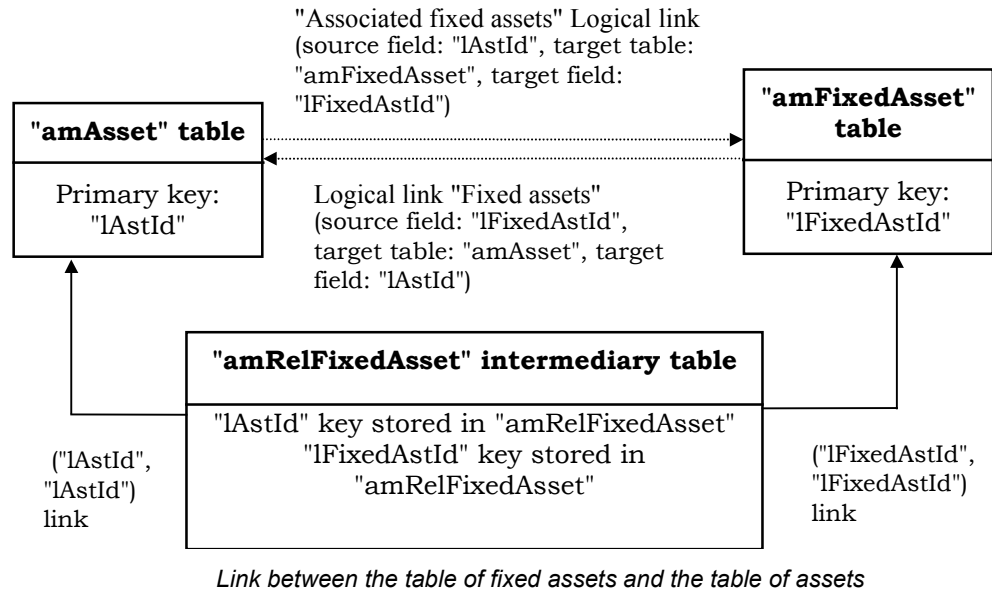
As opposed to normal tables, intermediary tables (sometimes called relation tables) do not have a primary key.

The following diagram explains how intermediary tables are used:



*Utilization of an intermediary table*

Example of the link between the table of assets and the table of fixed assets:



In this case:

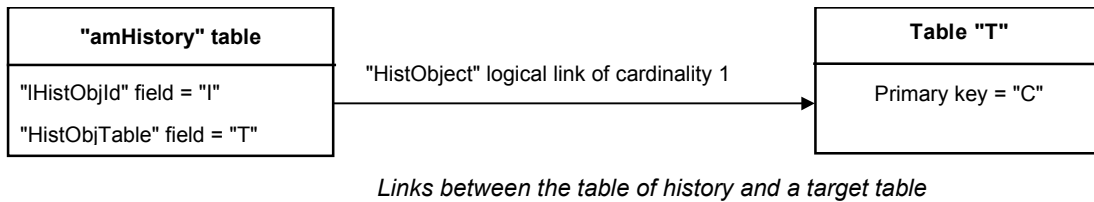
- An asset can be the subject of several fixed asset items (logical link "Associated fixed assets"):
  - ❖ Each record in the table of assets can be linked to several records in the intermediary table.
  - ❖ Each record in the intermediary table is linked to one single record in the table of fixed assets.
- A fixed asset can concern several assets (logical link "Fixed assets"):
  - ❖ Each record in the table of fixed assets can be linked to several records in the intermediary table.
  - ❖ Each record in the intermediary tab is linked to one single record in the table of assets.

Note: Although intermediary tables do not appear in the AssetCenter user interface, it is sometimes necessary to use them when writing complex queries.

## Contextual links

In some cases, the target table is not pre-defined, but is specified in the source table. This is called a "contextual link". This kind of link, which has a cardinality of 1, has no reverse link.

Example: the case of a link between the table of history and a target table:



---

## Description of indexes

This section describes the indexes in the AssetCenter database:

- Identification of indexes
- Uniqueness of field values of an index in a table

### Identifying indexes

An index is identified by:

- Its "SQL name", ending with "Id". SQL names are the same for all the various language versions of AssetCenter.
- Its "Label"/
- Its "Description".

### Uniqueness of indexed field values

The values of indexed fields may or may not be duplicated according to the nature of the index.

In AssetCenter Database Administrator, the nature of the index is shown by an icon displayed to its left:

- No icon: No constraints,

- 📌: The indexed list of fields must not occur more than once in the table.
- 🔑: The indexed list of fields must not occur more than once in the table, except for the "NULL" value which may be repeated any number of times.

For example:

In the **Software** table (SQL name: amSoftware), the "Soft\_PublisherName" index uses the following fields:

- **Name** (SQL name: Name),
- **Publisher** (SQL name: Publisher),
- **Version level** (SQL name: VersionLevel).

This index is "unique or null". This means that a record such as:

Microsoft, Word, 97
---------------------

Cannot be found more than once.

On the other hand, it is possible to have duplicates of a record for which these three fields are "null" at the same time.

---

## Customizing the database

AssetCenter allows you to customize the database in order to tailor it to the needs of your enterprise. This level of customization is restricted to an administrator.

Modifications are visible to AssetCenter users. All users view the database in the same way, as defined by the administrator. When the administrator modifies the name of a field or a link, the new name appears whenever it is used in AssetCenter, in particular in the list screens, detail screens, and when building queries.

This section provides a detailed explanation of how to customize the various objects in the AssetCenter database:

- Customizing tables
- Customizing fields and links
- Customizing the indexes
- Default values for fields and links
- Counters in field default values

- Errors in the description of default values
- HTML tags recognized in help on fields and links

## Customizing tables

You can customize a table's "Description", "Label" and "Description string" using AssetCenter Database Administrator.

## Customizing fields and links

There are several way to customize the database fields and links:

- From the **Configure object** popup menu:  
To display the **Configure object** popup menu, move to the field to configure and click the right mouse button.

Note: if you use the popup menu, changes are recorded: either when you close the database and confirm the validation message, or when you select the **Tools/ Administration/ Save database configuration** menu item.

- By using AssetCenter Database Administrator.

The **Detail** and **Scripts** tabs in AssetCenter Database Administrator, as well as the **General** tab allow you to define:

- the labels for fields and links,
- the descriptions of fields and links,
- the default values for fields and links,
- the mandatory fields and links,
- the fields and links for which you want to log modification history,

the read-only fields and links.

Note: you can also set the maximum size of a field of type "Text" via AssetCenter Database Administrator when creating the database.

You can also modify the context-sensitive help for a field or a link from the **Help** tab:

- "Description": explains the contents of the field or link.
- "Example": provides examples of how to populate the field or link.
- "Important": list of "sensitive" points: data entry precautions, related automatic mechanisms, etc.

The help system for fields is a sub-set of the HTML language.

## Customizing the indexes

You can customize the indexes in the database using AssetCenter Database Administrator.

For each index, you can define:

- The label,
- The description.

## Default values for fields and links

Default values for fields and links are built from a series of:

- constant values surrounded by "quotation marks".
- script functions returning values.
- references to other database fields.

Note: AssetCenter automatically applies default values when creating a new record. Users creating or modifying the record can modify these values.

Default values are only applied when a record is created.

For example, if the default value of field x in record A refers to field y in record B, and if field y or record B is modified, the value stored in field x is unaffected unless record A is created after the modification.

## Example

```
RetVal="DS"+AmCounter("Delivery", 2)+AmDate()
```

- AmDate() provides the date on which the record was created.
- AmCounter("Delivery", 2) provides the current value of the "Delivery" counter incremented by 1, and expressed in two digits.

## Special case of calculated fields

**Calculated string** and **Basic script** type calculated fields may only be used in the calculation of the default value of a standard field.

### Special case of links to the *Comments* table (SQL name: amComment)

It is not possible to assign default values to links to this table.

Example: **Remarks** (SQL name: Comment).

### Be careful when defining default values for fields and links

The calculation of default values for fields does not take access restrictions into account. You must therefore only reference those fields and links that may be viewed by all users.

### Counters in field default values

When you define a default value for a field, you can reference a counter.



When creating a new record containing this field, AssetCenter automatically inserts the number, incremented by one unit, for each new record.

***Counters are managed by the administrator using the Tools/ Administration/ Counters menu item.***

You must have created a counter using this menu before you can insert the counter in a formula for a field default value. Counter names cannot contain the following characters: <space>, "\$", "(" or ".")"

You may create as many counters as you want.

AssetCenter users never see the counter names.

The counter is incremented as soon as you click the  button. If you cancel the record creation without clicking the  button, the counter is not decremented.

You are free to readjust the values of the counters.

The value displayed in the counter detail is the last number used for this counter.

Note: if you enter a formula such as AmCounter(<Counter name>, [n]) in the default value of a field instead of AmCounter(<Counter name>), the value of the counter will be displayed using n digits.

## Errors in the description of default values

Here follow some of the most common mistakes made when describing default values:

### You have defined a tag that must be unique in the default value

Some fields do not accept several variables. This is the case for date fields, for example. In this case, you can only enter one variable in the default value.

### The type is not compatible with the tag

In some cases the variable may be incompatible with the field type. This is the case if you try to define AmLoginName() on a field of type date, for example.

### Field XXX is not found in table XXX

When you refer to a link, you must use the "Link.Link.Field" format where each link is referenced by name. This lets AssetCenter directly follow the links.

For example, to set a default supervisor for an asset, you may replace the cost center supervisor with: "CostCenter.Supervisor".

The default value will be used as soon as the AssetCenter user populates the asset's cost center field. Then the supervisor can be located in the database.

## HTML tags recognized in help on fields and links

The in-depth help system for AssetCenter fields uses a sub-set of the HTML language. Only a few "tags" are recognized, used mainly for formatting the Tooltip. This section does not attempt to provide an exhaustive description of the HTML language. It simply gives a brief description of each of the HTML tags managed by AssetCenter.

In the following table, the HTML tags are listed as they must be entered by the user. They do not follow the notation conventions applicable to the rest of the on-line help.



HTML Tag	Description
<FONT FACE="font name">	Defines the font used to display the text following this tag. This font remains in use until the next change of font.
<FONT COLOR=#RRGGBB>	Defines the font color used to display the text following this tag. This color remains in use until the next change of color.
<FONT SIZE=+n>	Increases the font size by "n" levels. This font size remains in use until the next change of font size.
<FONT SIZE=-n>	Decreases the font size by "n" levels. This font size remains in use until the next change of font size.
<B> </B>	These two tags surround text to be displayed in bold.
<I> </I>	These two tags surround text to be displayed in italics.
<LI>	This tag is the start of a bulleted list. It causes a carriage return and inserts a bullet at the start of the text.
<HR>	Draws a horizontal line used as a text separator.

For further information on the HTML language, refer to one of the numerous publications on the subject.

## Reserved characters

The following table lists the reserved characters in the HTML language. If you enter these characters, they will not be displayed as such on the screen. You should use the corresponding tag to display the desired character.

Reserved character	Use the following tag to display this character
"<"	"&lt;"
"&"	"&amp;"
Non breaking space	"&nbsp;"



# Chapter 3 - Creating, customizing and describing the AssetCenter database

This chapter explains how to create, customize and describe a database using AssetCenter Database Administrator.

You will find information on the following topics:

## ***Introduction***

- Using AssetCenter Database Administrator
- Launching AssetCenter Database Administrator

## ***Overview***

- Description of the File menu
- Description of the Edit menu
- Description of the Action menu

## ***Database management***

- Creating a database
- Customizing a database
- Repairing a database

## ***Creating database description files***

- Description of an AssetCenter database
- Database description parameters

- Syntax for description templates
- Information on certain database description parameters
- Values generated during the export process

In the remainder of this chapter, we will use the term "File" to describe one of the types of files managed by AssetCenter Database Administrator, i.e. the databases and database description files. A work session begins when you open a file.

---

## Using AssetCenter Database Administrator

AssetCenter Database Administrator is a tool for creating a customizing AssetCenter databases.

This tool is aimed at the administration and requires a certain number of precautions:

- Since this tool allows you to modify the structure of an AssetCenter database, we recommend limiting its use to users with the appropriate skills.
- In day to day use, AssetCenter also allows the user logged in as "Admin" to modify the database (configuration of objects, links, etc.) using the **Configure object** command from the popup menu and the **Tools/ Administration/ Save database configuration** menu item. You must not modify the database using AssetCenter and AssetCenter Database Administrator at the same time.
- AssetCenter Database Administrator requires a connection with the "Admin" login (Administrator) or a login with administrative rights to the database in question. We advise against giving more than one person this login. It can result in conflicts when updating the database or compromise the integrity of the structure of the database thus rendering it unusable.

---

## Launching AssetCenter Database Administrator

AssetCenter Database Administrator is executed from the same program group as AssetCenter. Click its icon to launch it.

When you launch AssetCenter Database Administrator, the main screen is shown grayed-out. To start using AssetCenter Database Administrator, you must first open a file using the **File/ Open** command.

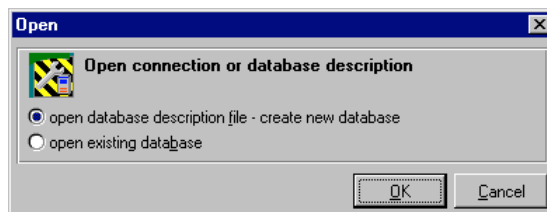
---

## Description of the File menu

This menu groups together all operations concerning loading and saving files.

### The File/ Open menu item

Selecting this menu item displays the following screen:



*Database opening screen*

This screen allows you to select one of the two functions of AssetCenter Database Administrator, i.e.:

- Create a new database or modify a database description file by selecting **Open database description file - Create new database**
- Customize an existing database by selecting **Open existing database**

Selecting one of these options opens a session under AssetCenter Database Administrator.

### The Open database description file - Create new database" option

In order to create a new database, AssetCenter Database Administrator needs a database description. This description takes the form of a file containing the structural information on an AssetCenter database. This file serves as a template which can be customized when creating a database.

When you confirm this choice you need to give AssetCenter Database Administrator a description file. Description files have the "dbb" file extension. The "gbbase.dbb" file shipped with AssetCenter in the installation folder is intended for this purpose. We recommend making a copy of this file, in order to always have a standard description file at your disposal.


Similarly, we advise systematically generating a database description file for your databases (using the **File/ Save as** menu item) and making a copy of this file. You will need this file if you ever have to repair your database.

### The Open an existing database option

You must, of course, connect to an AssetCenter database before customizing it. Selecting this option displays the same connection window as used in AssetCenter.



*Connecting to a database to customize it*

This manual does not detail the connection edition functions that are accessible via the  button). This subject is covered in more detail in the "Reference Guide: Introduction" manual.

Please note that the **Login** field cannot be modified since the connection can only be made by the administrator.

### The File/ Close menu item

This menu allows you to close an AssetCenter Database Administrator session. If any modifications have been made, AssetCenter Database Administrator lets you save them before closing the session.

In order to indicate that modifications have been made to a database and before confirming these modification by saving, AssetCenter Database Administrator adds an asterisk to the name of the document in the title bar of the application.

### The File/ Save menu item

This menu item allows you save the modifications made to the database description file or the database. If no modifications have been made, this menu item is disabled.

### The File/ Save as menu item

This menu item allows you to perform two distinct actions:

- If a database description file is open, this menu item allows you to save the structure of the database in a new database description file.
- If a database file is open, this menu item allows you to save the structure of the database in a database description file.

### The File/ Exit menu item

This menu item allows you to exit AssetCenter Database Administrator. If any modifications have been made during the session, AssetCenter Database Administrator will offer to let you save them.

### Last documents opened

Beneath the final separator in the **File** menu, AssetCenter Database Administrator lists the four last documents opened. In this way you can rapidly access these documents by simply selecting them.

---

## Description of the Edit menu in AssetCenter Database Administrator

This menu groups together all the classic edit functions, i.e.:

- "Cut" (Shortcut Ctrl+X) to cut the selection.
- "Copy" (Shortcut Ctrl+C) to copy the selection.
- "Paste" (Shortcut Ctrl+V) to paste the selection.

---

These function only work in field edit zones.

---

---

## Description of the Action menu in AssetCenter Database Administrator

This menu groups together several actions:

- **Create database.** This action is only available when the open document is a database description file.
- **Verify database.** This action is only available when the open document is a database.
- **Edit authorization string.** This action allows you to edit the authorization string that enables database access rights as per the license contract you acquired.
- **Templates.** This action allows you to launch a template (file extension **.tpl**) which creates a description of the AssetCenter database.
- **Generate compatibility views.** This action is only available when the open document is a database.
- **Update wizard script.** This action inserts a wizard configuration file, "gbbase.wiz", in the database. In particular, this file contains a two images and library of functions available for wizards. The library of functions is inserted in the **Options of the application** table (SQL name: amOption), in the "Wizard" section, "Config" entry); the two images, "Config.bmp" and "Config16.bmp" are inserted in the **Images** table (SQL name: "amImage").

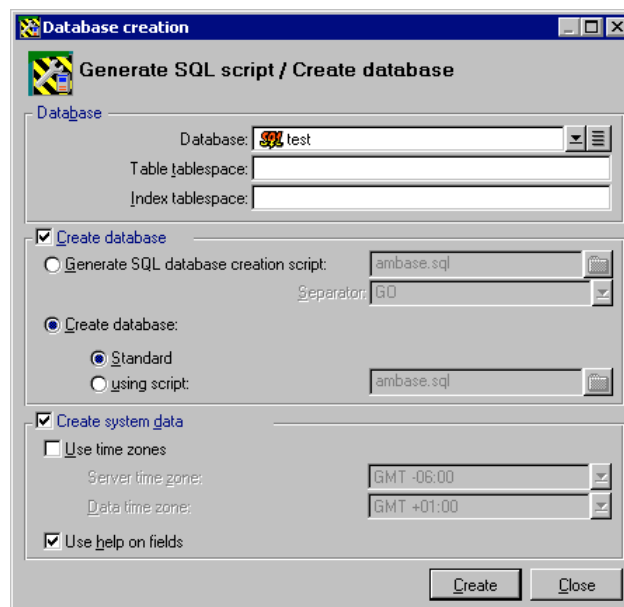


## The Action/ Create database menu item

This menu item allows you to create a database using the open database description file.

In order to be able to use this menu item, you must first create an empty shell using your DBMS. For further information, please refer to the manual entitled "Reference Guide: Administration and Advanced Use", chapter "Creating, customizing and describing the AssetCenter database", section "Creating a database".

The following window is displayed.




*Standard database creation options*

## Options in the Database frame


These options allow you to select an existing AssetCenter connection and define tablespaces for tables and indexes. Tablespaces must be created beforehand with your DBMS.

## Options in the Create database frame

The options in this frame are only available if the **Create database** option is checked. In this case, you can:

- ❖ Directly create the database, if the **Create database** and **Standard** option buttons are selected.
- ❖ Generate a SQL script to create the database later on if the **Generate SQL database creation script** option button is selected. Click  to name the script and define a SQL separator (classic separators include "/" for Oracle, and "GO", for other databases) from the drop down list.

The drop-down list is editable. You can freely define any other separator as long as they are valid (for example ";"). Nothing stops you from defining a separator such as "<MySeparator>", but the database creation script will no longer work.

- ❖ Create the database using a SQL database creation script (which you can generate using the **Generate SQL database creation script** option in the same frame) if the **Create database** and **using script** options are selected. Click  to give the name of the SQL script.

### Options in the Create system data frame

If you check the **Create system data** check box, AssetCenter Database Administrator creates the following system data:

- ❖ Database description file stored in the system table (**System tables** table (SQL name: SysBlob))
- ❖ Password (empty by default) of the "Admin" login encrypted and stored in the "sysblob" table
- ❖ Creates the "Admin" user
- ❖ Creates records with null identifiers (used to simulate outer joins) in each table
- ❖ Creates itemized lists and counters

The **Create system data** option also allows you to repair a database whose "sysblob" table is destroyed or missing.

If you check the **Use time zones** check box, AssetCenter Database Administrator creates the following system data:

- ❖ Information on time zones stored in the **Options of the application** table (SQL name: amOption).

When the **Use time zones** option is selected, it allows you to define time zones for the server and data in GMT.



*System data creation options*

If you check the **Use help on fields** box, AssetCenter Database Administrator prepares your database to accommodate the information concerning the extended help on fields.

### Entering the authorization string

To activate the database access rights corresponding to your AssetCenter license, you must provide AssetCenter Database Administrator with an authorization string. This string is stored definitively in the database; it does not need to be declared at the level of client machines. This key can be obtained from Peregrine Systems after providing certain pieces of information. The pieces of information needed by Peregrine Systems together with the field used to enter the authorization string are located in the window that will be displayed just after launching the creation of the database.

Before calling Peregrine Systems:


- ✧ Define the machine to be used to run the instance of AssetCenter Server that will regularly verify the authorization string. This is machine on which the **Signal presence to database server** module is activated (**Options/ Configure** menu item, **Modules** tab). If possible, this instance of AssetCenter Server should not be moved to another computer, since this will require changing the authorization string.
- ✧ Determine the Mac address of the network card of this computer.

To find the Mac address of the network card of a given machine, you can execute AssetCenter Server on this computer, select **Help/ About** and click **More...**. You do not need to connect to a database to do this.

The authorization string determines:

- ❖ The number of authorized users,
- ❖ The DBMS that may be used,
- ❖ The enabled functions.

Here are the steps you must follow:

- ↳ Launch the creation of the database by pressing the  button.
- ↳ When the window for entering the authorization string is displayed, call Peregrine Systems.

If you cannot reach Peregrine Systems, press the  button.

You can carry out the following steps later. At that time, follow these steps in order to return to the window for editing the authorization string, after the database has been created:

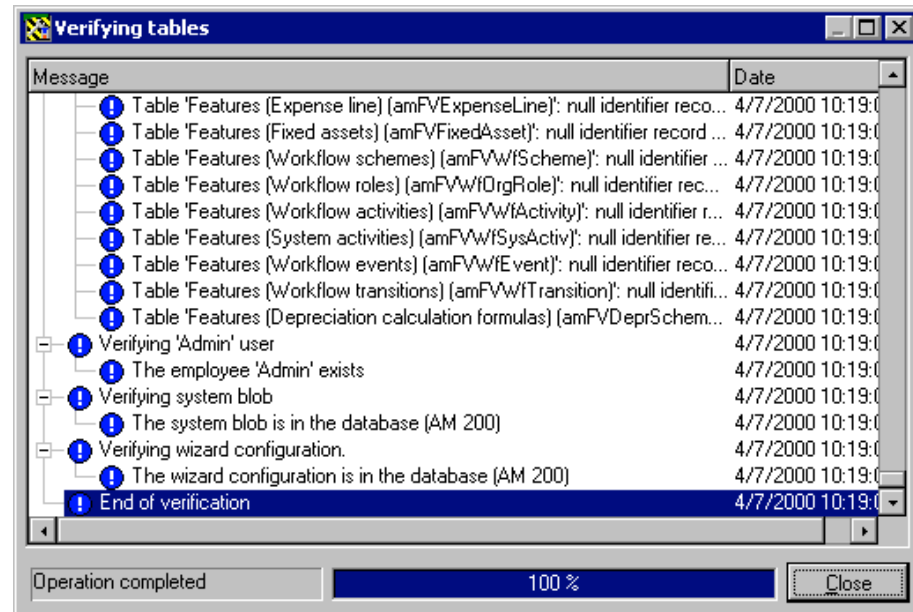
1. Start AssetCenter Database Administrator.
2. Open the database using the **File/ Open** menu item.
3. Select **Open existing database**.
4. Connect to the database you created.
5. Select the **Action/ Edit authorization string** menu item.

- ↳ Tell the person at Peregrine Systems the value of the **MAC address for AssetCenter Server** field (this value is displayed automatically), and the value of the **Record name (company)** field (you must enter this value).
- ↳ Enter the authorization string provided by the person at Peregrine Systems in the **Authorization string** field.
- ↳ Run AssetCenter Server and connect to the database you want to validate with an authorization string. This confirms the authorization and allows users to connect to the database.


If you change the network card of the computer on which AssetCenter Server is executed, you will need to call Peregrine Systems for a replacement authorization string.

## The Action/ Verify database menu item

This function allows you to test the integrity of an existing AssetCenter database. You need to be connected to the database in question in order for this command to be available.






*Verifying an existing database*

When you select this menu item  AssetCenter Database Administrator runs a series of checks:

- Presence of all tables in an AssetCenter database.
- Presence of records with null identifiers for all tables.
- Existence of the "Admin" user (database administrator) in the table departments and employees.
- Presence of the "sysblob" table (system data integrated with the database) in the database.

AssetCenter Database Administrator indicates the result of each test using icons.

-  indicates that the test has been successfully completed.
-  indicates that the test has failed but the database is useable.
-  indicates that the test has failed. The database may not be useable.

## The Action/ Templates menu item

The menu item for managing and executing database description templates is divided into several sub-menus:

- **Select folder** allows you to specify the directory where AssetCenter Database Administrator will search for description templates. The software searches all the sub-directories in the selected directory.
- **Refresh list** restarts the search for description files from the directory specified the last time you changed directory.
- The remaining sub-menus consist of all the description templates found by AssetCenter Database Administrator in the folder. You can execute a description template by selecting its name from the list displayed in the menu.

When executing a database description template, if Ambda encounters a variable whose value is not defined in the template, a screen is displayed for you to enter the value of that variable.

## The Generate compatibility views menu item

This menu allows you to create SQL views that are compatible with older versions of AssetCenter. These compatibility views allow you to use Crystal reports based on previous versions of AssetCenter.

You cannot create compatibility view for version of AssetCenter prior to version 2.0.

When you select this menu entry, AssetCenter Database Administrator invites you to select the database for which you want to create a compatibility view.

You can only select versions prior to the current version of AssetCenter.

Once you have made your selection, **AssetCenter Database Administrator** organizes the following information:

- **Tables:** Tables that no longer exist are ignored.
- **Fields:** Fields that no longer exist are assigned an arbitrary value, which maintains the integrity of the Crystal reports. Next, you can modify this value or the Crystal report as needed
- **Features.**

---

If you hold down SHIFT when selecting the menu, the compatibility view creation window displays in detail all the operations carried out and supplemental information concerning missing tables or default values assigned to missing fields, etc.

---

The SQL compatibility view created in this way can be used directly by Crystal Reports.

---

## Creating a database

Here are the steps to follow to create an AssetCenter database:

- ↳ Create an empty shell using the DBMS.
- ↳ Launch AssetCenter Database Administrator.
- ↳ Open a database description file.
- ↳ Customize the database before creating it.
- ↳ Create the database.
- ↳ Customize the database after creating it.

---

## Customizing a database

Two distinct forms of database customization are handled by AssetCenter Database Administrator:

- Customizing before creating the database.
- Customizing after creating the database.

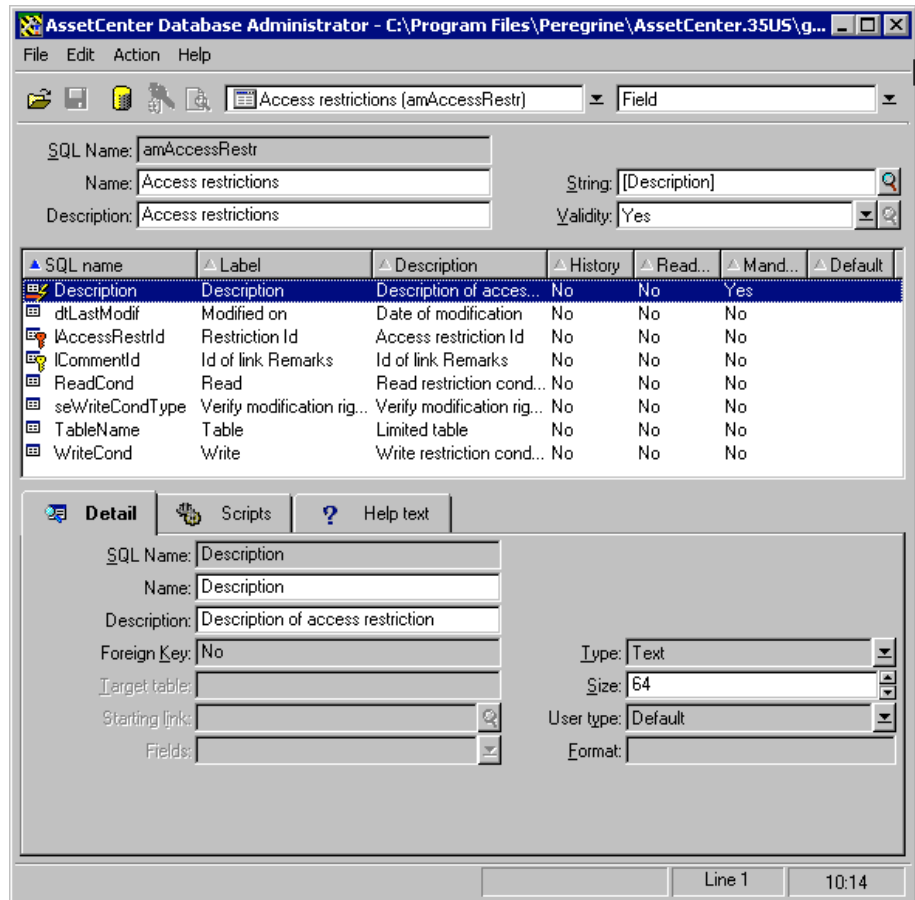
---

The difference between these two cases is that you can only modify the size of "Text" fields before creating the database.

---

For each of these two cases, two levels of customization are possible:

- Customizing a table.
- Customizing objects (fields, links and indexes) of a table.



*Database customization screen*

The customization detail screen is divided into three parts.

## First part

The first part allows you to:

- Select the table to be customized, in the "Table" field.
- Select the object to be customized (field, link or index).
- Customize that table.



SQL Name:	amAccessRestr	String:	[Description]
Name:	Access restrictions	Validity:	Yes
Description:	Access restrictions		

### *Selecting and customizing the table*

To customize the table, you can modify:

- The **Description** field.
- The **Label** field, which contains the name of the table as it is displayed under AssetCenter.
- The **String** field, which allows you to build the string which represents a record from this table under AssetCenter.
- The **Validity** field applies to all records of a table in the database. It allows you to define the conditions of validity of creation or modification of a record in the table.
  - ❖ If this field is set to **Yes**, a record in the table can always be created or modified.
  - ❖ If this field is set to **No**, a record in the table can never be created or modified.
  - ❖ If this field is set to **Script**, you can define a script conditioning the validity of creation or modification of a record in the table.

For example, for **Numerical** type features, if is possible to restrict the creation of a feature if its value is outside certain limits. The following script concerns the **Numerical** field in the "amFeature" table:

```
if [seDataType] = 1 and [fMin] > [fMax] Then
  Err.Raise(-1, "The value of the 'minimum' field must be less
  than the value of the 'maximum' field.")
  RetVal = FALSE
Else
  RetVal = TRUE
End If
```

When creation or modification of a record is invalidated by the value of the **Validity** field for the table concerned, it is good practice to display an explicit error message via the standard Basic function "Err.Raise" in order to warn the user. If you do not do this, the user will not necessarily be able to understand why it is not possible to modify or create the record.

Note: the "SQL name" field cannot be modified; it contains the table's SQL name.


## Second part

The second part of the customization screen lists all the objects of a given type.

SQL name	Label	Description	History	Read...	Mand...	Default
Description	Description	Description of acces...	No	No	Yes	
dtLastModif	Modified on	Date of modification	No	No	No	
lAccessRestId	Restriction Id	Access restriction Id	No	No	No	
lCommentId	Id of link Remarks	Id of link Remarks	No	No	No	
ReadCond	Read	Read restriction cond...	No	No	No	
seWriteCondType	Verify modification rig...	Verify modification rig...	No	No	No	
TableName	Table	Limited table	No	No	No	
WriteCond	Write	Write restriction cond...	No	No	No	

*Selecting the object type with list of objects of this type*


When you select an object in the list, AssetCenter Database Administrator displays the description of this object in the third part of the customization screen.

The red  icon identifies the primary key of the table.

The yellow  icon identifies a foreign key.

The  icon identifies a field belonging to an index.

The  icon means that the indexed list of fields must not occur more than once in the table.

The  icon means that the indexed list of fields must not occur more than once in the table, except for the "NULL" value which may be repeated any number of times.

The  represents a 1 link.

The  icon represents an n link.

## Third part

The third part of this table allows you to customize a selected object.

The screenshot shows a software interface with three tabs: 'Detail', 'Scripts', and 'Help text'. The 'Detail' tab is active. It contains several input fields and dropdown menus for customizing an object. The fields are: 'SQL Name' (containing 'Description'), 'Name' (containing 'Description'), 'Description' (containing 'Description of access restriction'), 'Foreign Key' (containing 'No'), 'Target table' (empty), 'Starting link' (empty with a magnifying glass icon), and 'Fields' (empty with a magnifying glass icon). On the right side, there are three dropdown menus: 'Type' (set to 'Text'), 'Size' (set to '64'), and 'User type' (set to 'Default'). Below these is a 'Format' field (empty).

*Customizing an object*

To customize an object, you can modify (in the "Detail" tab):

- The **Label** field.
- The **Description** field.
- The **Size** field allows you to specify the size of "Text" type fields. It can only be accessed when customizing a database before creating it or when modifying a database description file. The maximum size is 255 characters.

And in the **Scripts** tab:

Except where specified, the following attributes are taken into account both by the user interface and external tools when accessing the database.

- The **History** field: indicates whether history is kept for modifications made to the field and possible conditions (via scripts).
- The **Read only** field: indicates whether it is possible to modify the field via the user interface and under what possible conditions (via scripts).

This attribute is not taken into account when importing data using external tools. The import module does not take these into account either, as long as the fields are not defined as "Read only" at the level of mappings; on the other hand, if an import script maps a source field to a read-only target field, the read-only field can be modified all the same.

- The **Filter** field is not used in this version of AssetCenter.
- The **Formatting** field: allows you to format the value of the field automatically before storing it in the database.

For text fields:

- ❖ **Standard**: stores the value as entered.
- ❖ **Uppercase**: converts the value to uppercase before storing it.
- ❖ **Lowercase**: converts the value to lowercase before storing it.
- ❖ **Automatic**: converts the first letter to uppercase before storing the value.

For numeric fields:

- ❖ **Standard**: accepts all numbers, both positive and negative.
- ❖ **Positive**: rejects negative values. The warning message is displayed.

If you modify this attribute, existing values in the database will not be converted.

The **Mandatory** field: You can define conditions making a field mandatory.

Making a field mandatory can pose problems if it is not always visible (for example, if its being displayed is conditioned by the value of a field). Always bear this in mind when configuring a field or writing a script.

The **Default** field: You can specify the default value of a field. This value is automatically proposed by AssetCenter when creating a new record. Default values are defined using BASIC scripts.

Calculated fields can only be used in the calculation of the default value of a standard field if their type is **Calculated string** or **Basic script**.

It is not possible to attribute a default value to links to the amComment table.

In the **Help text** tab, you can customize the extended help concerning a database object. This help is available in AssetCenter via "Shift+F1" (or the **Help/ Help on this field** popup menu item) and can include up to three sections. By default, these sections are entitled "Description", "Example(s)" and "Important". The titles of these sections can be customized by modifying the labels of the links with SQL names


"Comment", "Sample" and "Warning" in the "Help on fields" table (SQL name: amHelp).

Note: The other fields in this section of the database customization screen are shown for informational purposes only and cannot be customized. Configuring the objects in this screen works in the same way as is available via the **Configure object** popup menu command.

---

## Repairing a database

To repair a database whose **System tables** table (SQL name: SysBlob) has been destroyed or is missed, use the following procedure:

- ↳ Use the **File/ Open** menu item, **Open database description file - create new database** option to open a backup of the description file.
- ↳ Select the **Actions/ Create database** menu item and specify the name of your database in the options shown in the **Database** frame. Verify that the **Create system data** and **Use time zones** check boxes are selected.
- ↳ Click .

Your database is repaired.

---

## Description of an AssetCenter database

There are several ways to obtain a description of the AssetCenter database structure:

- **Database.txt** and **Tables.txt** files: they contain the complete structure of the database. These files are located in the **Infos** sub-directory in the AssetCenter installation directory.

These files describe the default database structure. It does not include any customization you may have performed.

In order for these description files to reflect the customization of your database, use the AssetCenter Database Administrator software with a connection to your database.

- AssetCenter Database Administrator program: it allows you to freely create description files of the AssetCenter database (tables, fields, links and indexes).

It makes use of:

- ❖ A description file of the AssetCenter database (file with the **.dbb** extension) or a connection to the AssetCenter database.
- ❖ A template (file with the **.tpl** extension) that describes which information should be generated. We provide you with standard templates, and you can create your own templates. Sophisticated templates allow you to create files in "RTF" or "HTML" format.

Among the templates that are supplied with the standard version of AssetCenter, one of them, "dbdict.tpl", allows you to export the customization data (including information on features, calculated fields, configuration scripts, etc.) from your database to a standard text file. Used with a source control tool, this description file can be useful in order to track customizations made to the database.

- The AssetCenter program.

## Database description parameters

The following parameters are used to describe the database:

```
Instance DATABASE
Property P1-n
Collection TABLES as TABLE

Instance TABLE
Property P1-n
Collection FIELDS as FIELD
Collection LINKS as LINK
Collection INDEXES as INDEX
Object O1-n as <instance name>

Instance FIELD
Property P1-n
Object O1-n as <instance name>

Instance LINK
Property P1-n
Object O1-n as <instance name>
```

```

Instance INDEX
Property P1-n
Collection FIELDSINDEX as FIELD
Object O1-n as <instance name>

Instance SCRIPT
Property P1-n
Collection REFERENCEDFIELD as SCRIPTFIELD
Object O1-n as <instance name>

Global Values
Property P1-n

```

Describing the structure of an AssetCenter database is equivalent to describing the following instances:

- Database: the database itself.
- Table: database tables.
- Field: fields in the tables.
- Link: links in the tables.
- Index: indexes in the tables.
- Script: scripts for calculating the values of fields.

Each instance may be described with the following information:

- Property: a property of the instance.

Example:

```

Instance Table
Property SqlName

```

The "SqlName" property gives the SQL name of the table.

- Collection: set of items that constitute one of the components of an instance.

Example:

```

Instance Index
Collection FieldsIndex as Field

```

An index (one of the components of the "Index" instance) is in particular defined by a set of fields ("FieldsIndex" collection). Each field is an item in the "Field" instance.

- Object: designated component of an instance.

Example:

```

Instance Link
Object SrcField as Field

```

A link (one of the components of the "Link" instance) is in particular defined by a source field ("SrcField" object). This field is a component of the "Field" instance.

---

## Syntax for description templates

AssetCenter Database Administrator uses templates which define the information to extract, how to process it and present it.

These files must be in the following format:

- Type: text.
- Character set: ANSI (Windows).
- Extension: **.tpl**.

Their syntax is as follows:

- Fixed text
- Comments
- Including another template
- Browsing, sorting and filtering components
- \$if...\$else...\$elseif...\$endif conditions
- Functions available in description templates
- Processing the value of a property using a function defined in a template
- Deleting the end-of-paragraph mark
- Counting the number of browsed components
- Defining a global variable for the template

### Fixed text

Any character string that does not start with the "\$" character, and which is not part of a function, is generated as is by AssetCenter Database Administrator.

To output a "\$", the template must contain the following string: "\$\$".

Example:

The template:



```
List of tables.  
SQL NAME  
$$
```

generates the following output:

```
List of tables.  
SQL NAME  
$
```

## Comments

Lines to be ignored by AssetCenter Database Administrator, used to add comments to the template, must begin with the "\$" character and be followed by a space.

Example:

```
$ This is a comment
```

## Including another template

To include an external template in a template, use the following syntax:

```
$include "<full pathname of the template to include>"
```

Example:

```
$include "e:\templates\dbscript.tpl"
```

Example of use: allows you to definitively define a reference template containing the functions that may be used by other templates which include the reference template.

## Browsing, sorting and filtering components

### General syntax

```
$for [<name of the collection> | *] [alias <name of the alias>] [sort  
(<name of the first property> (ASC|DESC) [, <name of the next property>  
(ASC|DESC)]] [<filter condition>]  
...  
$endfor
```

## Browsing the components of a collection with "\$for...\$endfor"

To perform an iterative browse of the components of a collection, use the following syntax:

```
$for <name of the collection>
...
  $for <sub-collection>
  ...
  $endfor
$endfor
```

Example:

```
$for Tables
...
  $for Fields
  ...
  $endfor
$endfor
```

You must respect the hierarchy between collections. Examples:

- ❖ The "Fields" collection is dependent on the "Tables" collection.
- ❖ The "FieldsIndex" collection is dependent on the "Indexes" collection.

You may replace <name of the collection> with the "\*" character. This calls all the collections in the current instance.

Example:

```
$for Tables
...
  $for *
  $(SqlName)
  ...
  $endfor
$endfor
```

allows you to obtain the SQL name of all the collections in the "Table" instance, i.e.: "Fields", "Links" and "Indexes".

## Sorting the final result with "sort"

To sort the components in a collection, use the following syntax:

```
$for <collection> sort (<name of the first property> (ASC|DESC) [,
<name of the next property> (ASC|DESC)])]
...
$endfor
```

Where:

- ❖ ASC: ascending alphabetical order.
- ❖ DESC: descending alphabetical order.

Example:

```
$for Tables sort (SqlName ASC)
...
$for Fields sort (Usertype DESC, UserTypeFormat ASC, SqlName ASC)
...
$endfor
$endfor
```

## Obtaining the properties of items in a collection or an object

To obtain the value of item properties of a collection or an object, use the following syntax:

```
$for <collection>
...
$([<name or alias of the collection>].[<name of the
object>].<Property>
...
$endfor
```

<name or alias of the collection> is not necessary if the property is called in a "\$for... \$endfor" loop in the collection.

Example:

```
$for Tables
$for Fields
$(Tables.SqlName) $(SqlName)
$endfor

$for Links
$(Tables.SqlName) $(SqlName) $(Reverselink.SqlName)
$endfor

$endfor
```

## Assigning an alias with "alias"

For the moment, aliases have no particular use.

## Filtering the contents of the collection with "filter"

To filter the components of a collection, use the following syntax:

```
$for <collection> filter <filter condition>
...
$endfor
```

The filter condition is written in Basic.

Example:

```
$for tables filter $Left($SqlName, 1) = "p"
...
$endfor
```

keeps only tables whose SqlName starts with the letter "p".

## **\$if...\$else...\$elseif...\$endif conditions**

You can define the scope of a condition to include a property so that a component is selected.

Syntax:

```
$if <test condition>
...
$elseif <test condition>
...
$else <test condition>
...
$endif
```

Test conditions may be expressed using Basic formulas, functions defined in "\$script...\$endscript" format, and properties of instances.

Example:

```
$for Links
$if $(typed) = 0
$(Tables.SqlName)      $(SqlName)      $(SrcField.SqlName)
      $(DstTable.SqlName)
$else
$(Tables.SqlName)      $(SqlName)      $(SrcField.SqlName)
$endif
$endfor
```

## **Functions available in description templates**

AssetCenter Database Administrator contains a few pre-defined functions that may be used in templates.

### ValueOf(<strProperty> as String) as String

Alternative syntax for calling the value of the **Property** property.

**Property** must be in uppercase.

Example:

```
$ValueOf( "PRIMARYKEY" )
```

produces the same result as:

```
$(PrimaryKey)
```

### SetProperty(<strProperty> as String, <strValue> as String, <iValueType> as Integer) as String

Creates a global variable called **Property** and of type **ValueType** for the template.

**Property** must be in uppercase.

Examples:

```
I = SetProperty("NEWPROPERTY", "2", VarType(2))
```

creates a global variable called **NEWPROPERTY** for the template, with a numeric value of **2**, and returns a return code **I** equal to "0" if the variable was created correctly.

```
I = SetProperty("NEWPROPERTY", "Test", VarType("Test"))
```

creates a global variable called **NEWPROPERTY** for the template, with a text value of **Test** and returns a return code **I** equal to "0" if the variable was created correctly.

### Exist(<strProperty> as String) as Integer

Tests for the presence of the **Property** global variable in the template.

Example:

```
Exist( "NEWPROPERTY" )
```

returns a numeric value of **1** if the property exists, otherwise returns **0**.

### **LogError(<strErrorCode> as String, <strMessage> as String) as String**

Defines an **ErrorCode** and an error message **Message** to be returned.

Example:

```
LogError(1, "property not found!")
```

produces an ASCII error message in the defined cases.

### **SetOutput(<strFile> as String) as String**

Defines the output file for the results. Has priority over the output file defined in the command line.

Examples:

```
SetOutput("e:\exportdb\output.txt")
```

stores the results in a file called "e:\exportdb\output.txt".

```
SetOutput("")
```

displays the results on the screen.

### **CollectionCreate(<strName> as String) as Integer**

Declares a new collection of database items. The name of the collection created must be a valid database collection, such as "Fields" or "Tables". This function and the following functions are typically used to scan the components of a collection. They can thus be used as a substitute for the proprietary "\$For...\$Next" syntax

Example:

```
CollectionNext() as IntegerCollectionCreate("Fields")
```

The function returns "0" if the collection is created. Other values correspond to error codes that are displayed explicitly.

### **CollectionNext() as Integer**

Carries out an iteration on the collection previously defined using "CollectionCreate()".

Example:

```
CollectionNext()
```

The function returns "0" if the iteration is carried out successfully. All other return codes correspond to errors. The function also returns an error if the last element of a collection is reached.

### **CollectionName() as String**

Returns the name of the collection previously declared using the "CollectionCreate()" function.

Example:

```
strName=CollectionName( )
```

### **CollectionIsFirst() as Integer**

Allows you to test if the element of the collection to which the program is pointing is first of the collection.

Example:

```
CollectionIsFirst( )
```

This function returns "1" if the element is the first of the collection; else "0" in all other cases.

### **CollectionIsLast() as Integer**

Allows you to test if the element of the collection to which the program is pointing is last of the collection.

Example:

```
CollectionIsLast( )
```

This function returns "1" if the element is the last of the collection; else "0" in all other cases.

### **CollectionCurrentIndex() as Integer**

Returns the index number of the element of the collection to which the program is pointing. The collection must be declared beforehand using the "CollectionCreate()" function.

Example:

```
Number=CollectionCurrentIndex( )
```

### CollectionCount() as Integer

Returns the number of elements contained in the current collection previously declared using the "CollectionCreate()" function.

Example:

```
iCollec=CollectionCount()
```

## Processing the value of a property using a function defined in a template

### Using a function with "\$<function>"

You can define functions and process property values using those functions.

Syntax for using the function:

```
$<function>($(<property 1>,...,<property n>))
```

Examples:

```
$StrType$(Type)
```

```
$Duplicates$(Duplicates), $(NullValues))
```

### Defining functions with "\$script...\$endscript"

Functions are defined within a BASIC block delimited by two markers: "\$script" and "\$endscript":

```
$script
...
  Function
...
  End Function
...
$endscript
```

Functions have the following syntax:

```
Function <function name>({ByVal|ByRef} [<name of the input variable> as
<data entry format>]*) as <output format>
...
End Function
```



Functions may be expressed using BASIC formulas and instance properties.

Examples:

```
Function00825 ReturnYesNo(ByVal iValue as Integer) as String
if iValue = 1 then
    ReturnYesNo = "Yes"
else
    ReturnYesNo = "No"
end if
End Function
```

```
Function StrType(ByVal iValue as Integer) as String
select case iValue
    case 1: StrType = "byte"
    case 2: StrType = "short"
    case 3: StrType = "long"
    case 4: StrType = "float"
    case 5: StrType = "double"
    case 6: StrType = "string"
    case 7: StrType = "date+time"
    case 9: StrType = "blob"
    case 10: StrType = "date"
    case 12: StrType = "memo"
    case else
        Dim strError as String
        strError = "Type" + CStr(iValue) + " undefined"
        strType = LogError(1, strError)
    end select
End Function
```

## Deleting the end-of-paragraph mark

In some cases, you may need to insert information within a line, but the function generating the information must start at the beginning of a line.

In these cases, you can add the following string:

```
$nocr
```

to the end of the line before the function.

Example:

```
...
$for Indexes
$(Tables.Sqlname)    $(Sqlname)    $nocr
for FieldsIndex
    $if $(IsLast) = 1
        $(Sqlname)
    $else
```

```

    $(Sqlname)$nocr
    , $nocr
$endif
$endfor
...
```

generates the following output:

```

...
amProduct      Prod_BrandModel      Brand, Model
amProduct      Prod_CatalogRef      CatalogRef
amProduct      Prod_lCategIdBrand    lCategId, Brand, Model
```

## Counting the number of browsed components

To count the number of components that were browsed in a collection, taking a filter into account if appropriate, use the following syntax:

```

$for <collection> filter <filter condition>
$(count)
...
$endfor
```

## Defining a global variable for the template

To define a global variable, use the following syntax:

```

$<name of the variable> = <BASIC formula>
```

Examples:

```

$A = 1
```

```

$Var = "text"
```

```

$A = $(A) + 1
```

```

$Form = Left($(Var), 2)
```

---

## Information on certain database description parameters

- 

This section includes information on the following description parameters:

- Database instance
- Table instance
- Field instance
- Link instance
- Index instance
- Script instance
- Global variables

## Database instance

### Properties

Property name	Description	Connection required
LoginName	Name of the Login you use to access the database.	X
TableCount	Total number of tables in the database.	
Connected	This field may have two values: <ul style="list-style-type: none"><li>❖ 1: AssetCenter Database Administrator was executed in reference to a connection.</li><li>❖ 0: AssetCenter Database Administrator was executed in reference to a database description file.</li></ul>	
Connection	Name of the AssetCenter connection used to access the database.	X

## Table instance

### Properties

Property name	Description	Connection required
Rights.Create	This property may have two values: <ul style="list-style-type: none"><li>❖ 1: the login has create access rights for the table.</li><li>❖ 0: the login does not have create access rights for the table.</li></ul>	X
Rights.Delete	This property may have two values: <ul style="list-style-type: none"><li>❖ 1: the login has delete rights for the table.</li><li>❖ 0: the login does not have delete rights for the table.</li></ul>	X
ComputeString	Table description string.	
InternalName	Internal name. This information has no particular use.	
Label	Label.	
Desc	Description.	

Property name	Description	Connection required
SqlName	SQL name.	
FieldCount	Total number of fields in the table.	
LinkCount	Total number of links in the table.	
IndexCount	Total number of indexes in the table.	
IsFirst	Indicates whether the item is located at the start of the collection, given the filter and the sort order: ❖ 0: no ❖ 1: yes	
IsLast	Indicates whether the item is located at the end of the collection, given the filter and the sort order: ❖ 0: no ❖ 1: yes	
Count	Count of the browsed items in the collection, given the filter.	
CurrentIndex	Position of the item in the collection, given the filter and the sort order.	

## Objects

Object name	Description
MainIndex as Index	Main index.
PrimaryKey as Field	Primary key.
FeatureValueTable as Table	Table in which the feature values are stored.
Base as Database	Described database.

## Field instance

## Properties

Property name	Description	Connection required
Rights.Update	This property may have two values: ❖ 1: the login has update rights for the	X

Property name	Description	Connection required
	<p>field.</p> <ul style="list-style-type: none"> <li>❖ 0: the login does not have update rights for the field.</li> </ul>	
Rights.Write	<p>This property may have two values:</p> <ul style="list-style-type: none"> <li>❖ 1: the login has create access rights for the field.</li> <li>❖ 0: the login does not have create access rights for the field.</li> </ul>	X
Rights.Read	<p>This property may have two values:</p> <ul style="list-style-type: none"> <li>❖ 1: the login has read rights for the field.</li> <li>❖ 0: the login does not have read rights for the field.</li> </ul>	X
UserType	<p>By default, the data entry and display format is the format for the "Type" property. The "UserType" property allows you to specify the data entry and display format when it is verified.</p>	
Type	Storage format.	
UserTypeFormat	Additional information on the "UserType" parameter.	
Size	Maximum size of the values in the field, in number of characters.	
ReadOnly	<p>Defines whether the field may be modified, whatever the access rights for the person connected to the database.</p> <p>This field may have two values:</p> <ul style="list-style-type: none"> <li>❖ 1: the field can never be modified by users.</li> <li>❖ 0: the field may be modified by a user, if the user has the required access rights.</li> </ul>	
Historized	<p>This field may have two values:</p> <ul style="list-style-type: none"> <li>❖ 1: history is recorded for the field.</li> <li>❖ 0: history is not recorded for the field.</li> </ul>	
ForeignKey	<p>This field may have two values:</p> <ul style="list-style-type: none"> <li>❖ 1: the field is a foreign key.</li> <li>❖ 0: the field is not a foreign key.</li> </ul>	
PrimaryKey	<p>This field may have two values:</p> <ul style="list-style-type: none"> <li>❖ 1: the field is a primary key.</li> <li>❖ 0: the field is not a primary key.</li> </ul>	

Property name	Description	Connection required
InternalName	Internal name. This information has no particular use.	
Label	Label of the field, as it appears in detail screens, for example.	
Desc	Description.	
SqlName	SQL name.	
Comment	Comments on the use of the field.	
Sample	Examples of values that may be assigned to the field.	
Warning	Important information for the field.	

## Objects

Object name	Description
Base as Database	Described database.
Table as Table	Table containing the field.
MandatoryScript as Script	Mandatory calculation script for the field.
DefaultScript as Script	Default script for calculating the field's value.

### Possible values for the "Type" property

Stored value	Literal value	Description
1	byte	Integer from -128 to +127.
2	short	Integer from -32,768 to +32,767.
3	long	Integer from -2,147,483,647 to +2,147,483,646.
4	float	4 byte floating point number.
5	double	8 byte floating point number
6	string	Text for which all characters are accepted.
7	date+time	Date and time.
9	blob	Used for storing images and formulas, for example, with no size restrictions.
10	date	Date only (no time).
12	memo	Variable-length text field.

### Possible values for the "UserType" property

Stored value	Literal value
0	Default
1	Number
2	Yes/No
3	Money
4	Date
5	Date+Time
7	System itemized list
8	Custom itemized list
10	Percentage
11	Time span
12	Table or field SQL name

### Possible values for the "UserTypeFormat" property

This property is used when the "UserType" property is set to:

- "Custom Itemized list": indicates the name of the itemized list associated with the field.
- "System Itemized list": provides the list of itemized entries.
- "Time span": indicates the display format.
- "Table or field SQL name": the property contains the SQL name of the field which stores the SQL name of the table containing the field which specifies the described field.



## Link instance

### Properties

Property name	Description	Connection required
Rights.Update	This property may have two values: <ul style="list-style-type: none"><li>❖ 1: the login has update rights for the link.</li><li>❖ 0: the login does not have update rights for the link</li></ul>	X
Rights.Write	This property may have two values: <ul style="list-style-type: none"><li>❖ 1: the login has create access rights for the link.</li><li>❖ 0: the login does not have create access rights for the link.</li></ul>	X
Rights.Read	This property may have two values: <ul style="list-style-type: none"><li>❖ 1: the login has read rights for the link.</li><li>❖ 0: the login does not have read rights for the link.</li></ul>	X
Type	Type of the link.	
UserType	Type of information managed by the link.	
Typed	Indicates whether or not the link's target table is pre-defined. When this is not the case, the table's SQL name is stored in one of the fields of the record. <ul style="list-style-type: none"><li>❖ 1: the target table is not pre-defined.</li><li>❖ 0: the target table is pre-defined.</li></ul>	
Historized	This field may have two values: <ul style="list-style-type: none"><li>❖ 1: history is recorded for the field.</li><li>❖ 0: history is not recorded for the field.</li></ul>	
Cardinality	Cardinality of the link.	
InternalName	Internal name. This information has no particular use.	
Label	Label.	
Desc	Description	
SqlName	SQL name.	

### Objects

Object name	Description
Base as Database	Described database.
SrcField as Field	Source field.
SrcTable as Table	Source table.
DstTable as Table	Target table.
DstField as Field	Target field.
RelTable as Table	Relation table.
RelSrcField as Field	Source field of the relation table.
RelDstField as Field	Target field of the relation table.
TypeField as Field	When a link's target table is not pre-defined, this property indicates the field containing the SQL name of the target table.
ReverseLink as Link	Reverse link.

#### Possible values for the "Type" property

Stored value	Literal value
1	Normal
2	Own
4	Define
8	Neutral
16	Copy
18	Owncopy

#### Possible values for the "UserType" property

Stored value	Literal value
0	Normal
1	Comment
2	Image
3	History
4	Feature value

## Index instance

## Properties

Property name	Description
Duplicates	<p>Indicates whether or not the index can have more than one non-NULL value.</p> <ul style="list-style-type: none"> <li>❖ 1: you can create several records where the group of fields in the index can have exactly the same value.</li> <li>❖ 0: you cannot create more than one record where the group of fields in the index has a given value.</li> </ul>
NullValues	<p>This property is only significant if the "Duplicates" property is set to "No".</p> <p>It indicates whether or not the index can have more than one "NULL" value (the index has a value of "NULL" if all the fields constituting the index have a value of "NULL".)</p> <ul style="list-style-type: none"> <li>❖ 1: you can create several records having an index of NULL.</li> <li>❖ 0: you can only create one record having an index of NULL.</li> </ul>
InternalName	<p>Internal name.</p> <p>This information has no particular use.</p>
Label	Label as it appears in the detail screens.
Desc	Description.
SqlName	SQL name.

## Objects

Object name	Description
Base as Database	Described database.
Table as Table	Table containing the index.

## Script instance

### Properties

Property name	Description
CalcMode	<p>Indicates whether the value of the field is "yes", "no", or if a script calculates one of these two values. This property can have one of the following values:</p> <ul style="list-style-type: none"> <li>❖ 0: "no"</li> </ul>

Property name	Description
	<ul style="list-style-type: none"> <li>❖ 1: "yes"</li> <li>❖ 2: "script"</li> </ul>
ScriptType	<p>Type of information managed by the script.</p> <p>This property can have one of the following values:</p> <ul style="list-style-type: none"> <li>❖ 1: data entry for a field is mandatory</li> <li>❖ 2: by default, display a feature in a table</li> <li>❖ 3: available character for a feature in a table</li> <li>❖ 4: record history for a field</li> <li>❖ 5: default value of a field</li> <li>❖ 6: <b>For inheritance purposes</b> (SQL name: bForInheritance) field of a feature in a table</li> </ul>
Source	Script which calculates the value of a field as it is displayed in the interface.
RawSource	Script which calculates the value of a field as it is stored in the database.
VbReturnType	<p>Type of string calculated by the script:</p> <ul style="list-style-type: none"> <li>❖ Integer: Integer from -32,768 to +32,767</li> <li>❖ Long: Integer from -2,147,483,647 to +2,147,483,646</li> <li>❖ Double: 8 byte floating point number</li> <li>❖ String: Text for which all characters are accepted</li> <li>❖ Date: Date (not time).</li> </ul>

## Objects

Object name	Description
Table as Table	Table containing the field whose value is calculated by the script.
Field as Field	Field containing the value calculated by the script.

## Global variables

Property name	Description
Userlogin	Login you used when you connected to the database.
Time	Time at which AssetCenter Database Administrator was executed.
Date	Date on which AssetCenter Database Administrator was executed
Dbb.Fullname	Full pathname of the database description file used.
Dbb.Shortname	Name (without extension) of the database description file used.
Dbb.Path	Pathname for the database description file used.
Dbb.Name	Name (with extension) of the database description file used.
Dbb.Ext	Extension of the database description file used.
Template.Fullname	Full pathname of the database description template used.
Template.Shortname	Pathname of the database description template used (without the extension).
Template.Path	Path of the database description template used.
Template.Name	Name of the database description template used (with the extension).
Template.Ext	Extension of the database description template used.

---

## Values generated during the export process

AssetCenter Export generates information using the format defined in the "UserType" and "UserTypeFormat" properties.



# Chapter 4 - Managing itemized lists

An itemized list is a list of values proposed by AssetCenter for populating certain fields (standard fields in a detail screen, or the value of a feature), e.g. quality, position, country, brand.

This allows you to standardize the values in these fields, and facilitates data entry.

The values appear in a "drop-down list". Simply choose the correct value from the list to assign a value to the field.

AssetCenter manages two types of itemized lists:

- Free itemized lists
- System itemized lists

---

## Free itemized lists

*An AssetCenter administrator can access free itemized lists using the Tools/ Administration/ Itemized lists menu item.*




There are two types of free itemized lists:

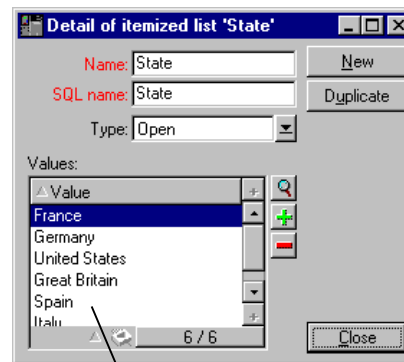
- Itemized lists that you create yourself. You can link them to features, but you cannot attach them to fields: only the software can assign an itemized list to a field.

- Lists that are linked to database fields. These lists may also be attached to features. If you delete an itemized list of this type, or modify its name, AssetCenter will recreate the itemized list (with no associated values) using its original name, as soon as the itemized list is required to create a record containing a field which should be linked to that list.

## Values of itemized lists

The list of "values" appearing in the detail of an itemized list contains the values that will be proposed when the user populates a field associated with the itemized list.

An administrator can delete, modify or add values using the ,  and  buttons located to the right of the list.



List of values that will be proposed when populating the field.

*Detail of an itemized list*

## Open itemized lists

The **Type** field (SQL name: seType) field in the detail of these itemized lists is set to **Open**.

AssetCenter users can enter other values than those proposed in the list.



If a user enters a new value, it is added to the list of values in the itemized list (the list of values is shared by all users). A message requests the user to confirm the creation.



## Closed itemized lists

The **Type** field (SQL name: seType) of the detail for these itemized lists is set to **Closed**.

AssetCenter users enter only values included in this list.

Note: After clicking  or  in the itemized lists management window, any modifications that the administrator defines for itemized lists are stored in the database. Modifications made to itemized list are only taken into account at the client level after the next connection to the database following the modification.

---

## System itemized lists

The list of values in a system itemized list is defined by AssetCenter. It cannot be customized by an administrator or a user.

These itemized lists cannot be edited from the **Tools/ Administration/ Itemized lists** menu item.

### Values in system itemized lists

The values displayed are different from the values stored in the database.

In the database, these values are stored as numbers.

Example of the **Assignment** field (SQL name: seAssignment) in the asset detail:

Value stored in the database	Value displayed
0	In used
1	In stock
2	Retired asset
3	Awaiting delivery

There are several ways to access the values in a system itemized list:


- Using the context-sensitive help for the field populated by the system itemized list.

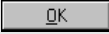
- From AssetCenter Database Administrator.
- Using the **Database.txt** file describing the database structure.

# Chapter 5 - How the history function works

Any modifications made to any of the fields or links in the database can be recorded.

In order to do this you need to specify that history is kept for the field or link. To do this:

- ↳ From the popup menu, select **Configure object**.
- ↳ Go to the **General** tab in the configuration screen.
- ↳ Set the **History** field to **Yes**.
- ↳ Click  to confirm.

Note: Any modifications concerning history are saved in the database as soon as you click  in the database customization window. You can also define whether history is kept for a field or a link using AssetCenter Database Administrator.

When history is kept for a field or a link, it is available for all users of AssetCenter.

As soon as history is kept for at least one field or link, a **History** tab appears in the record detail window for this table.

"History lines" are kept here which describe in detail any modifications which have been made to a field or a link.

History lines contain several pieces of information:

- **Modified on** (SQL name: dtLastModif): Date on which the modification was carried out
- **Author** (SQL name: Author): Person who performed the modification (login, name and first name)
- **Field** (SQL name: Field): Name of the field which has been modified (short description.)

- **Previous value** (SQL name: PreviousVal): Previous value of the modified field (except for "comment" type fields)
- **New value** (SQL name: NewVal): New value of the modified field (except for "comment" type fields). By default, this field does not appear in the list. To display it, right-click the list then select **Configure list**.

If you are importing a database from an older version of AssetCenter, the **New value** field will be empty for history lines.

- **Previous comments** (SQL name: memPreviousCmt): Previous value of "comment" type fields. "Comment" type fields are not handled in the same way as other fields, since they are stored differently in the database (max. size: 32767 characters.)

Depending on the case, AssetCenter functions differently:

## Record creation

Record creations are recorded if you have asked AssetCenter to track history of all modifications made to the identification field corresponding to the primary key of the table).

AssetCenter records:

- **Modified on:** Creation date
- **Author:** Author of the creation
- **Field:** "Creation"
- **Previous value:** "Creation"

## Modification of a field in the table or a 1 link (for example: User of an asset)

AssetCenter records:

- **Modified on:** Date on which the modification was made
- **Author:** Author of the modification
- **Field:** Name of field modified
- **Previous value:** Previous value of the modified field
- **New value:** New value of the modified field

## Addition of a n link to another table (for example: Assets covered by a contract)

AssetCenter records:

- **Modified on:** Date on which the addition was made
- **Author:** Author of the addition
- **Field:** Name of link
- **Previous value:** References of the linked record which has been added
- **New value:** New value of the modified link

### Deletion of a n link to another table

AssetCenter records:

- **Modified on:** Date on which the deletion was made
- **Author:** Author of the deletion
- **Field:** References of the linked record which has been deleted
- **Previous value:** References of the linked record which has been deleted
- **New value:** New value of the modified link (empty)

### Modification of a n link to another table

AssetCenter does not record modifications made to a link. To keep a trace, you need to delete the obsolete link, then add the new one.

### Keeping history of features associated with records

In AssetCenter, you can keep history of features, just like for any other field in the database. Feature history concerns:

- Adding features
- Removing features
- Changing the value of a feature

Several types of actions are kept in history:

#### Adding a feature

Additions of new features are recorded if the **Keep history** (SQL name: seKeepHistory) field for the feature is set to **Yes** and if the **Keep history even when creating main record** (SQL name: bCreationHistory) box is checked.

AssetCenter records:

- ❖ **Modified on** (SQL name: dtLastModif): date when the feature was added.
- ❖ **Author** (SQL name: Author): the person who added the feature.
- ❖ **Field** (SQL name: FieldName): SQL name of the feature.
- ❖ **Previous value** (SQL name: PreviousVal): "Creation".

### Deleting a feature

Deletions of features are recorded if the **Keep history** field for the feature is set to **Yes**.

AssetCenter records:

- ❖ **Modified on:** date when the feature was deleted.
- ❖ **Author:** the person who deleted the feature.
- ❖ **Field:** SQL name of the feature.
- ❖ **Previous value:** " Remove feature ("Value of feature").
- ❖ **New value:** New value of feature (empty)

### Modifying the value of a feature

Modifications of features are recorded if the **Keep history** field (SQL name: seKeepHistory) for the feature is set to **Yes**.

AssetCenter records:


- ❖ **Modified on:** date when the feature was modified.
- ❖ **Author:** the person who modified the feature.
- ❖ **Field:** SQL name of the feature.
- ❖ **Previous value:** previous value of the feature.
- ❖ **New value:** New value of the modified feature

Warning: If you delete a record, all history lines are also deleted, either straightaway or via AssetCenter Server.

### Creating, deleting or modifying history lines

It is not possible to keep history of the creation of history lines.

#### Creating history lines

To trigger the creation of history lines for a feature, you must set the **Keep history** field to "Yes". To do this, select the **Parameters** tab in the feature details and click the  button opposite the parameters line.

AssetCenter displays the screen with the details of the feature's parameters. The **Keep history** field is located in the **Constraints** tab in this screen.

When this field is set to **Yes**, AssetCenter automatically creates history lines for this feature. History lines may be viewed in the **History** tab of the table associated with this feature.

Warning: if you delete a record, all history lines are also deleted, either at the time of the deletion, or via AssetCenter Server. You cannot keep history of the creation of histories.





# Chapter 6 - Managing user access to AssetCenter databases

This chapter explains how to manage user access to AssetCenter databases.

You will find information on the following topics:

## **Introduction**

- Importance and overview of database access rights
- How to ensure data security and confidentiality
- Definitions

## **Access management**

- Defining access conditions
- Defining AssetCenter users
- Managing user connections
- Managing passwords

---

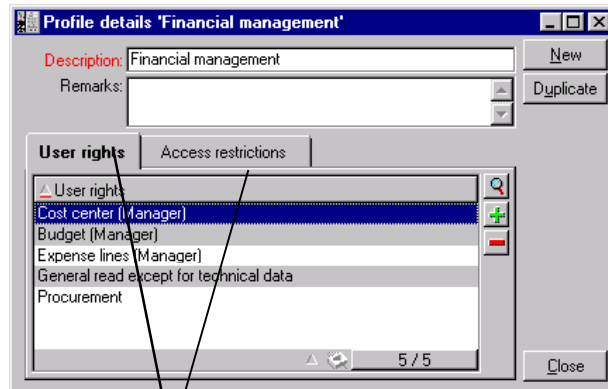
Access rules can only be managed by database administrators.

---

*Use the Tools/ Administration/ User profiles menu item to display the list of user profiles.*

*Use the Tools/ Administration/ User rights specific menu item to display the list of user rights.*

*Use the Tools/ Administration/ Access restrictions. specific menu item to display the list of access restrictions.*



A profile is made up of user rights and access restrictions.

*User profile detail*

---

## Importance and overview of database access rights

AssetCenter is a program that can be used simultaneously by several users: It uses a "shared" database.

Managing the database access rights in AssetCenter consists of two tasks:

- ✎ Defining the data which each user can access, and the access conditions.
- ✎ Managing user connections to the database according to the type of AssetCenter license acquired.

### Defining access conditions for each user

All users are not necessarily able to view or modify the same data in the database. This depends on their position and the organization of the enterprise. One employee may be authorized to create assets, another to access stocks, another to access work order slips, etc.

In order to allow an employee to access AssetCenter:

- The employee must be included in the table of departments and employees.
- An administrator must assign a "user profile" to the employee or declare the employee as an administrator.

A user profile consists of user rights and access restrictions.

A user right or an access restriction can be used in several user profiles. One user profile may be attached to several users. A given user can only have one profile, however.

## Managing user connections

The AssetCenter license you acquired limits the number of database connections. The license may be by the "number of simultaneous users" or by the "number of declared users".

AssetCenter assigns a connection slot to each connected user.

AssetCenter automatically manages the database connection slots. Nevertheless, an AssetCenter administrator can also manage them.

---

# How to ensure data security and confidentiality

To ensure data security and to make sure that your information is not wrongfully viewed, modified or destroyed, you can control three levels of access security:

- Define user access to the network.
- Define AssetCenter user profiles.
- Make regular backups of the database.

---

## Definitions

This section defines the concepts related to access management.

- Definition of a user profile
- Definition of a user right
- Definition of an access restriction

## Definition of a user profile

A user profile is a set of user rights for given tables and fields and access restrictions which apply to records within these tables.

Profiles are attributed to users of AssetCenter.

For example, you can define:

- An "Accounting" profile, with access restricted to cost centers, budgets and expense lines only.

## Definition of a user right

A user right is one of the component parts of a user profile. It concerns tables and fields in AssetCenter and not only certain records. As an administrator, you can assign read/write rights (as with an operating system) to different database users which apply to the different tables in AssetCenter.

## Definition of an access restriction

An access restriction is part of a user profile. It corresponds to a record filter on a table. For example, you can make sure technicians only have access to the assets in their own department. Access restrictions concern both reading and writing (append or modification) records.

---

# Defining access conditions

This section describes how to define access conditions:

- Defining user profiles
- Defining user rights
- Defining access restrictions

Once defined, these access conditions are associated with AssetCenter users.

## Defining user profiles

*Use the Tools/ Administration/ User profiles menu item to define user profiles.*


Each "User profile" brings together:

- User rights for reading, writing, creating or deleting information in database tables or fields.
- Access restrictions, defining read and write conditions for records in a given table. For example, a technician can only view equipment located at the site where they work, or where a group of users works.

A user profile can be considered as concerning a given function in a company and its specific prerogatives.

## Defining user rights

User rights can be managed via:

- the **Tools/ Administration/ User rights** menu item,
- or by clicking the  button to the right of the "User rights" list in a user profile detail (**Tools/ Administration/ User profiles** menu item).

User rights describe the access to database tables and fields.

We suggest you create a user right per table that describes the rights concerning the table's direct and linked fields. You can set several access levels for a single database table:

For example:

- Basic accounting,
- Advanced accounting,
- Status access for employees,
- Maintenance,
- Etc.

You can then combine these user rights to create user profiles.

- accountant,
- maintenance technician,
- intern,
- etc.

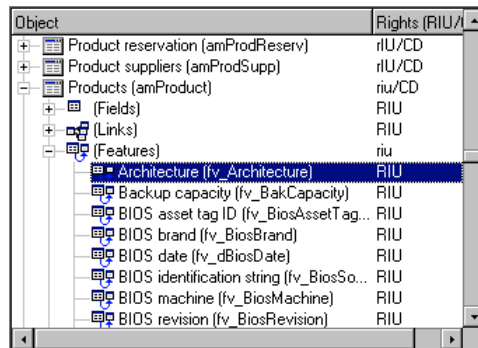
## Editing user rights

The database structure is presented as a tree of tables. For each table, the tree contains a list of fields and features specific to the table, and a

list of fields and tables from linked tables. Assign rights specific to tables, rights specific to fields, and rights specific to features.

The user right detail screen is organized as follows:

- Rights are displayed in a column.




*User rights*

- The "Display only tables with rights" filter allows you to view only tables for which user rights have been defined.
- When you select a node (Tables, Fields, Links, Features, etc.), AssetCenter automatically selects all the branches in the tree; this authorizes you to edit user rights for the entire node. For a parent node:
  - ❖ A lowercase "r" indicates that certain items in this node have read access rights.
  - ❖ An uppercase "R" indicates that all the items in this node have read access rights.
  - ❖ A lowercase "i" indicates that certain items in this node have create access rights.
  - ❖ An uppercase "I" indicates that all the items in this node have create access rights.
  - ❖ A lowercase "u" indicates that certain items in this node have update rights.
  - ❖ An uppercase "U" indicates that all the items in this node have update rights.
- You can perform multiple selections in this list. Therefore you can edit user rights for several items at once; Use the SHIFT and CTRL keys in to select multiple items in the tree.

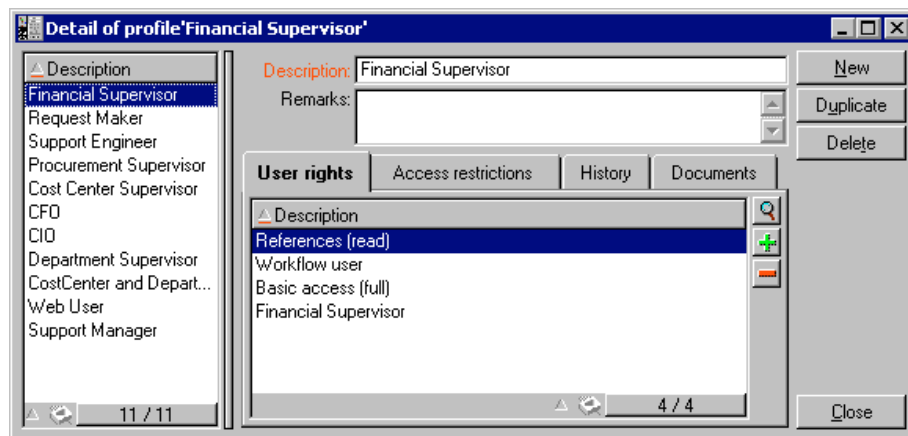
Note: In order for a user to be able to see a "conditional" tab, they must have at least the right to view the fields which conditions whether this tab is shown or not. For example, if a user does not have the right to view the **Nature of payments** field (SQL name: sePayType) in the **General** tab of a contract detail, they will not be able to see the **Rents** and **Loans** tabs in the contract detail, since these tabs are displayed or hidden depending on the value of the **Nature of payments** field.

## Defining access restrictions

Access restrictions can be managed via:

- the **Tools/ Administration/ Access restrictions** menu item,
- or by clicking the  button to the right of the "access restrictions" list in a user profile detail (**Tools/ Administration/ User profiles** menu item).

An access restriction describes which records the user can see in the different tables of the database.



*User Profile: Access restrictions tab*

You can limit status access and or modification rights by using criteria similar to queries. These criteria may be used for example to cover:

- Categories, brands, or asset models,
- Departments or locations,
- Insurance contracts.

You can then combine access restrictions with user right to create user profiles:

- Maintenance engineer at St. Louis,
- Purchasing supervisor,
- And so on.

## Editing access restrictions

Select the table concerned by the access restrictions and then define the access restrictions, for reading or for writing. These conditions are defined using the AssetCenter query editor.

### Read condition

Allows the user to view records which satisfy the criteria defined in the query editor, excluding all other records.

### Write condition

Allows the user to modify the field in an already existing record.

The screenshot shows a Windows-style dialog box titled "Select access restrictions for 'Marketing Bakersfield' user profile". On the left is a tree view with "Access restric..." expanded, showing "Bakersfield" and "Printers (read)". The main area has a "Description" field with "Bakersfield", a "Table" dropdown set to "Assets", and a "Conditions" section. Under "Conditions", there are "Read" and "Write" labels, each followed by a text box containing the query "Location.FullName LIKE '/Bakersfield Site/%'". A "Remarks" text box is at the bottom. On the right are "New", "Cancel", and "Select" buttons. A status bar at the bottom left indicates "2 / 2".

*Access restriction details*

Warning: You must create a corresponding user right. This is not done systematically by AssetCenter.



## Be careful when defining default values for fields and links

The calculation of default values for fields does not take access restrictions into account. You must therefore only reference those fields and links that may be viewed by all users.

---

# Defining AssetCenter users

To define a new user:

- Create the user in the table of departments and employees.
- Move to the **Profile** tab in the employee detail.
- Enter the employee's **Login** name and associated password. The **Login** is the name with which the user opens the database. This can be any character string.

Note: if you do not specify a password, the password is automatically the same as the **Login**. Users can then modify their passwords, after opening the database using the **Login** name.

- Populate the **Login type** (SQL name: seLoginClass) field.

The remainder of the procedure depends on whether or not you want to assign administrative rights to this employee.

## Defining a database administrator

Simply check the **Administration rights** (SQL name: bAdminRight) box in the **Profile** tab of the employee's detail. Then the employee will have all rights for all AssetCenter database tables.

Note: There is a default administrator in the table of departments and employees: this is the "Admin" **Login** record. For the first AssetCenter installation, this is the only name available for accessing the AssetCenter database for all administrative operations.

For security reasons, this "Admin" login record cannot be deleted. In effect, this is the only solution if for some reason you cannot connect under a **Login** name that also has administrative rights.



## Defining a non-administrative user

In this case, select a user profile in the **Profile** (SQL name: Profile) field in the **Profile** tab in the employee detail.

You can also assign specific rights to a user for the Procurement module.

## Rights specific to the Procurement module

To create an estimate or an order from a purchase request, simply click

 or  in the request detail window.

By default, Procurement Management does not allow you to create an order or an estimate from an unapproved purchase request. (Purchase requests are not approved until the **Approval status** field (SQL name: seApprStatus) in the **Approval** tab in the request details is set to "Approved").

The **Rights specific to the Procurement module** frame allows you to circumvent these rules.

---

# Managing user connections

This section explains how to manage user connections to the database.

- Three types of database access
- Principle of connection slots
- Destruction of connection slots

## Database access types

AssetCenter defines several types of access to the database.

***Define login types via the Profile tab in the employee detail.***

Login types govern connections made to the database via the user interfaces of the Windows versions of AssetCenter and AssetCenter Web, or via the AssetCenter APIs.

Note: The login types do not take into account connections to the database made via AssetCenter Export, AssetCenter Server, AssetCenter Database Administrator.

## Floating access

In this case, the license defines a maximum number of concurrent connections to the database; you cannot exceed this number.

You can define as many **Login** names as you want, but users may not be able to connect if the maximum number of concurrency users has been reached.

It is also possible for several connections to the database to use the same **Login** but each connection decrements the remaining number of free connections.

## Named access

In this case, the license defines the maximum number of **Login** names to the database that you can define.

Contrary to the previously defined mode, all declared users can connect to the database whenever they want.

However, you cannot enter more **Login** names than authorized by the license: AssetCenter displays an error message.

It is also not possible for several connections to the database to use the same **Login** name.

Note: The access type of the "Admin" **Login** is **Named**. This user is not deducted from the number of authorized users.

## Casual access

This login type is intended for users who rarely access the database. Casual users have standard logins and passwords, but their rights are limited.

For example, a casual user can:

- View the assets they use.
- Create purchase requests.
- Follow the evolution of purchase request that concern them.

The rights of a casual user are limited:

- By their assigned user profiles.
- By a set of access restrictions defined at the level of AssetCenter.

These two elements are linked by an AND clause.

The number of casual users is not counted off.

Several concurrent connections to the database can use the same "Login".

## How connection slots work

When a user connects to the database via AssetCenter or AssetCenter Web, AssetCenter assigns a connection slot.

For as long as the user is connected to the database, the connection slot is regularly updated by AssetCenter. The frequency of updating is defined by the **Slot update delay in minutes** option in the table displayed by the **Tools/ Administration/ Database options** menu item. By default, it is set to 5 minutes.

As soon as a user disconnects from the database, the corresponding connection slot is destroyed.

## Detail of a connection slot


*An AssetCenter administrator can view connection slots via the Tools/ Administration/ Connection slots menu item.*

A connection slot is defined by:

- The name of the application which is using it (in general AssetCenter, etc).
- The user's **Login**.
- The user's **Login type** (SQL name: seLoginClass).


## Destroying connection slots

Connection slots can be destroyed:

- Manually via the  button in the connection slot management window.
- Automatically in the case of accidental termination of the application.
- Automatically if the connection is left idle.

## Destroying connection slots manually

The administrator using the "Admin" login can force the disconnection of a user. To do this:

- Display the connection slot screen via the **Tools/ Administration/ Connection slots** menu item.
- Select the appropriate connection slot.
- Click .

Note: This is only possible under the "Admin" login and not under other administration logins.

## Detecting accidental termination of the application

In some cases, the application which uses a connection slot may be terminated accidentally (brutal shut-down of the operating system, network problem, etc.). Therefore users are not logged off from the database in the normal fashion (e.g. via the **File/ Disconnect from database** menu item).

The connection slot still exists, but it is not actually used.

As a consequence, if access to the database is based on the principle of floating users, the number of effective concurrent users is decremented by 1.

AssetCenter solves this problem by searching at regular intervals for unused connection slots, and using them for a new connection. Here is how AssetCenter determines if a connection slot is obsolete:

- AssetCenter searches for the date the slot was last modified.
- If the time at which the slot was last modified is twice as old as the slot update frequency, AssetCenter considers that the slot is inactive and therefore can be reused.

An administrator using the "Admin" login can also fix this problem by deleting obsolete connection slots. Because connection slots are refreshed regularly during the database connection, it is easy to delete obsolete records from the table of connection slots:

- ↳ Use a query filter to display all records in the table of connection slots whose **Modified on** field indicates a date earlier than a given date.
- ↳ Select those slots and delete them.

## Disconnecting inactive users

AssetCenter allows you to disconnect users automatically according to a given timeout value.

This is defined via the **Tools/ Administration/ Database options** menu item.

- The **Automatic disconnection** option allows you specify if you want to activate the automatic disconnection procedure.
- If you decide to activate the automatic disconnection procedure define a value in the **Slot timeout in seconds** option.

---

## Managing passwords

This section explains how to manage passwords with AssetCenter.

You will find information on the following topics:

- Password for the Admin login
- Modifying the password for the Admin login
- User password
- Lost passwords

### Password for the Admin login

The record of the "Admin" login is very important:

- For the first installation of AssetCenter, this is the only login name available for accessing the AssetCenter database for all administrative operations.
- You can subsequently assign administrative rights to other records in the table of departments and employees. However, the "Admin" login record cannot be deleted; it is your only solution if you can no longer connect under another **Login** name that also has administrative rights.

The password associated with the "Admin" login is therefore critical, because it gives all rights to the AssetCenter database.

Warning: Do not forget the passwords for the "Admin" login, or you will never be able to administer the database if the other records in the table of departments and employees with administrative rights are damaged.

Warning: Anyone who knows the administrator password can perform any operation on the AssetCenter database and can access all the

database information, with no restrictions. Make sure you do not tell this password to any unauthorized employees.

---

## Modifying the password for the "Admin" login

You can modify the password for the "Admin" login record by opening the database under this login name and using the **Tools/ Change password** menu item.

## User password

### Modifications performed by the user

Every AssetCenter user can modify their password by opening the database under their Login name and then selecting the **Tools/ Change password** menu item.

### Modifications performed by an administrator

An administrator may modify user passwords in the employee's **Profile** tab. This overwrites the old password.

## Lost passwords

### User password

AssetCenter does not display passwords in a readable format. If an AssetCenter user loses or forgets their password, an administrator must create a new one in the employee's **Profile** tab. This erases the old password.

### "Admin" login password

If you have forgotten your administrator password despite the warnings above, there is no way to recover it.





# Chapter 7 - Writing queries in AQL

This chapter explains how to write queries in AQL:

- Presentation
- Recommendations for writing AQL queries
- Sorts and indexes
- The query editor
- AQL syntax
- AQL function references
- Examples of queries

---

## Presentation

This section lists the places where you might have to use queries and exposes the AQL language:

- Queries in AssetCenter
- AQL

### Queries in AssetCenter

Queries allow you to combine criteria concerning information in a table or in linked tables.

Within the AssetCenter suite, you can use queries:

- To create filters on record lists. In this case, the queries are, in general, simple and based on the "Where" clause.

- To define views.
- To define export conditions in the export module.
- To create Crystal Reports.
- To create wizards.
- When you use the AssetCenter APIs and/or AssetCenter WebKit.
- If AssetCenter is used as a DDE server.

AQL ("Advanced Query Language") is AssetCenter's internal querying language intended to simply access to the AssetCenter database.

AssetCenter includes an editor which allows you to draw up queries in AQL:

- Either indirectly by using the graphic interface,
- Or by writing the query directly in AQL.

Warning: In order to be illustrative of the capabilities of AQL, the examples given later use all of the AQL syntax. The SELECT, WHERE, and FROM clauses, in particular, are explicitly shown. Certain functions, such as query filters (where the user only defines the WHERE clause in the AQL query) or the expression builder greatly simplify the creation of queries for the user (certain clauses are not visible). These examples cannot be used for these functions.

## AQL

AQL ("Advanced Query Language") is the querying language used by AssetCenter to access the AssetCenter database. It is comparable to SQL. Queries written in AQL are automatically translated into the corresponding SQL language of the database engine used.

Note: It is useful to be well acquainted with SQL and to have good knowledge of databases before using AQL directly.

## Usefulness of AQL

AQL is better suited to querying the AssetCenter database than SQL for the following reasons:

## Independence from database

The various database engines supported by AssetCenter all use differing versions of SQL which are incompatible with each other. AQL is independent of the database engine used.

As a consequence, if you migrate to another database engine, your queries written in AQL will work the same.

For example, AQL uses the same set of functions, regardless of the database engine used.

Thus, the "Substring" function in AQL is equivalent to "Substr" under Oracle SQL and "Substring" under Microsoft SQL Server SQL.

## Generation of optimized SQL code

AQL generates SQL code optimized according to the database engine used.

This makes a big difference in the utilization of sorts and indexes. For example, when searching assets (sorted by brand, using the indexes) you would write the following in AQL:

```
SELECT FIRST_ROWS Brand FROM amAsset ORDER BY Brand
```

The SQL code generated will differ according to the DBMS used and will be optimized for this. Thus, under Oracle 7.3, the SQL code will be:

```
SELECT /*+ FIRST_ROWS INDEX_ASC(a1 Ast_Brand) */ a1.Brand FROM  
amAsset a1 WHERE a1.Brand >= CHR(0)
```

Under SYBASE 11, the SQL code will be:

```
SELECT a1.Brand FROM amAsset a1(Index Ast_Brand)
```

Under Microsoft SQL Server 6.5 the code will be:

```
SELECT a1.Brand FROM amAsset a1(Index=Ast_Brand)
```

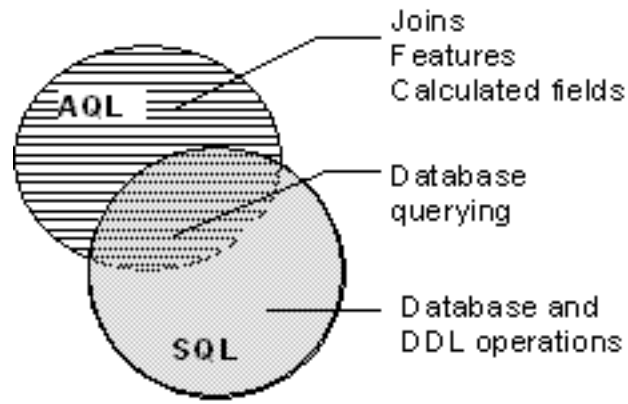
## Simplified access to the AssetCenter database

AQL simplifies the management of links and joins. This greatly facilitates access to the database when writing queries as compared to using SQL directly.

In addition, AQL simplifies access to features, allowing you to use them as direct fields in their related tables.

AQL also facilitates the utilization of calculated fields.

## How AQL compares to SQL



*Comparison of AQL and SQL*

The above diagram illustrates how AQL compares to SQL:

- AQL is compatible with SQL database querying statements ("SELECT").
- AQL does not have an equivalent for SQL statements which write to the database ("INSERT", "UPDATE", "DELETE") or DDL statements ("Data Definition Language").

Warning: You should never write to the AssetCenter database directly using SQL statements.

- AQL comprises extensions which simplify the handling of joins, features and calculated fields.

---

## Recommendations for writing AQL queries

We recommend reading this section before starting to write queries in AQL.

This section deals with:

- Notation specific to AQL.
- The specificities of AQL and the AssetCenter database which have an effect on the optimal design of queries.

The sections entitled "AQL syntax" and "AQL function reference" complement this section.

You will find the following sections:

- Presentation of AQL joins
- Reason for and usefulness of primary key 0 records
- Usage of NULL
- Self
- CurrentUser
- System itemized-lists
- Hierarchic tables
- Simplified AQL notation

Important note: Queries written in AQL use the SQL names ("SQLName") of fields, links and table in the database. Refer to the **Database.txt** file which describes the structure of the database and which includes an exhaustive list of these names.

## Presentation of AQL joins

### Definition

A join consolidates multiple data tables in a single query.

### AQL joins

AssetCenter's database description, besides defining the tables and fields, defines the links between tables. This allows you to automate the generation of joins at AQL level.

AQL links are expressed as:

```
Link[.Link[.Field]]
```

With the handling of joins simplified in this way, AQL simplifies the creation of the majority of queries used for the AssetCenter database.

### Example

The following query, written in AQL, returns for each Asset Tag, the name of its user and its supervisor:

```
SELECT AssetTag, User.Name, Supervisor.Name FROM amAsset
```

Here is the same query written in Oracle SQL:

```
SELECT B1.AssetTag, U2.Name, R3.Name FROM amAsset B1, amEmplDept U2,  
amEmplDept R3 WHERE B1.lUserId = U2.lEmplDeptId AND B1.lSupervId =  
R3.lEmplDeptId
```

The two joins between the table of assets and the table of departments and employees are automatically handled under AQL. Using the AssetCenter's graphic query editor, you just have to click the fields of a selected table or linked table in a tree-structured list in order to create the corresponding AQL code.

## Reason for and usefulness of primary key 0 records

### Primary key "0" records

The AssetCenter data module has certain specificities:

- The primary and foreign keys of each table are numeric (32-bit integer).
- A foreign key which does not point to a record is set to "0" (and not "NULL").
- Each table has an empty record whose primary key is set to "0".

### Usefulness

With these primary key "0" records, the results of a query using a non outer join between two given tables, A and B, can include records from table A which are not linked to any real record in table B (link not populated). These are records in table A which are linked to the primary key "0" record in table B.

Example:

The following query, written in AQL, returns for each Asset Tag, the name of its user and its supervisor:

```
SELECT AssetTag, User.Name, Supervisor.Name FROM amAsset
```

An asset which is not assigned to a user and/or supervisor appears in the results of the query. At database level, such an asset is linked to the primary key "0" record in the table of departments and employees.

## Reason for these specificities

This section explains why primary key "0" records are used, whereas a query using an external SQL join can select records in table A which are not linked to a record in table B.

Primary key "0" records allow you to compensate for the fact that certain RDBMs do not handle multiple outer joins: Using primary key "0" records, SQL queries generated from AQL queries do not use outer joins.

Example:

The AQL query below searches for each asset, its asset tag and name of location of its user. The results will include assets which do not have a user and assets whose users do not have a location.

```
SELECT AssetTag, user.location.name FROM amAsset
```

If the generated SQL code used the outer joins of the DBMS, the SQL code generated for Sybase SQL Server would be:

```
SELECT a.AssetId, l.name FROM amAsset a, amEmplDept e, amLocation l
WHERE a.lUserId *= e.lEmplDeptId AND e.lLocaId *= l.lLocaId
```

This code is not supported by Sybase SQL Server since it uses several outer joins one after another.

However, since there is a primary key "0" record in the table of departments and employees and locations, it is not necessary to use outer joins. AssetCenter thus generates SQL code without outer joins:

```
SELECT l.name FROM amAsset a, amEmplDept e, amLocation l WHERE
a.lUserId = e.lEmplDeptId AND e.lLocaId = l.lLocaId
```

This query gives the expected results, since the "User" and "Location" links still point to a record in the table of departments and employees or locations (they point to the primary key "0" record if the link is not populated).

## Consequences

- It is important to take these records into account in the queries that you write, especially when using aggregate functions.

Example:

```
SELECT count(AssetTag) FROM amAsset "
```

If you execute the above query which counts the number of assets in the table of assets, the primary key "0" record is taken into account in the results. You therefore need to decrease the results by 1 in order to obtain the real number of assets in the database.

- It is rarely necessary to have to generate outer joins at DBMS level.

Note: If you really want to manage outer joins at DBMS level, use the "=" and "<=" AQL operators.

## Usage of NULL

AssetCenter uses the NULL value of the DBMS in two cases only:

- For an empty "text" type field.
- For a non populated "date" or "date+time" type field.

AQL allows you to use several equivalent syntaxes, as shown below. It converts them to the equivalent valid SQL code of the database engine.

For empty "Text" type fields, you can use any of the following syntaxes, since the NULL value will be stored in the database:

```
WHERE <text field> = NULL
WHERE <text field> IS NULL
WHERE <text field > = ''
```

For non-populated "date" or "date+time" type fields, you can use any of the following syntaxes, since the NULL value will be stored in the database:

```
WHERE <date or date+time field> = NULL
WHERE <date or date+time field> IS NULL
WHERE <date or date+time field> = []
```

Note: When a "Numeric" type field is not populated, its value is "0". Similarly, the absence of a link is described as "Link = 0" or "foreign key = 0".

Example: "Location=0" or "lLocaId=0".

## Self

"Self" is an expression which is equivalent to the description string on the table to which it is applied.



Using "Self" allows you to simplify queries and take any customization of the AssetCenter database into account.

Example:

If the description string of the table of departments and employees is:

```
Name ", "FirstName" ( "Phone" ) "
```

The AQL query:

```
SELECT self FROM amEmplDept
```

Is equivalent to:

```
SELECT Name + ", " + FirstName + "( " + Phone + " )" FROM amEmplDept
```

## CurrentUser

"CurrentUser" allows you to write queries which depend on the person connected to the database.

"CurrentUser" can be used as an expression, for example in a query, or as a link. You have to enter this expression manually as it is not offered by the query editor.

### Used as "expression"

Example: We are looking for all the assets used by the employee connected to the database.

```
SELECT lAstId FROM amAsset WHERE User = CurrentUser
```

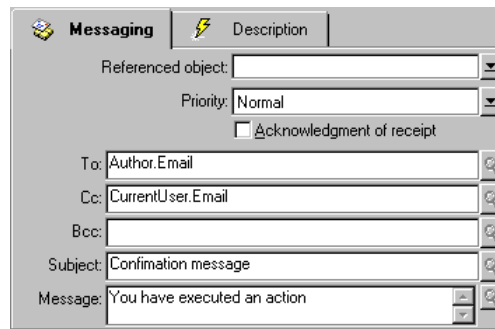
### Used as "link"

"CurrentUser" can be considered as a link found in every table which points to the record corresponding to the current user in the table of departments and employees.

- With the form "CurrentUser", this function points to the record corresponding to the current user.
- With the form "CurrentUser.Field", this function returns the value of the field for the current user.

Example: When an action is triggered by the connected user, it is possible to contextually trigger another messaging action, which

automatically sends a warning message to the connected user. You just need to populate the detail of the action as follows:



*"CurrentUser" function used as variable in an action*

## System itemized-lists

If an AQL query uses a system itemized list, you must use the values which are stored in the database and not those which displayed on screen.

Example:

The following query selects those contracts whose **Type** field (SQL name: seType) is set to **Master lease**:

```
SELECT Self FROM amContract WHERE seType = 1
```

The **Type** field (SQL name: seType) is a system itemized list. The values stored in the database are:

- 0 for **Other**
- 1 for **Master lease**
- 2 or **Lease schedule**
- 3 for **Insurance**
- 4 for **Maintenance**

Note: The values of system itemized list can be found via AssetCenter Database Administrator, or the **Database.txt** file which describes the structure of the database.

## Hierarchic tables

All the hierarchic tables contain:

- A "FullName" field.
- A "sLvl" field.

### "FullName" fields

For each record in a hierarchic table, the "FullName" field stores the value of a field of the record, preceded by a tree structure constituted by the values of the fields of parent records until root level.

Values are separated by the "/" character without spaces. This character also appears at the start and at the end of the tree-structure.

Examples:

- For the table of assets, the "FullName" field stores the Asset Tag of the asset preceded by the Asset Tag of its parent asset, that in turn preceded by the Asset Tag of its parent asset, and so on..

```
FullName = '/PC118/DD054/CR012/'
```

- In the table of locations, the "FullName" field stores the name of the location preceded by the names of parent locations.

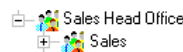
```
FullName = '/Paris/Site Tolbiac/Tour A/Floor 5/'
```

### "sLvl" fields

For each record in a hierarchic table, the "sLvl" indicates its level in the tree-structure.

Root level is level 0.

Example in the table of departments and employees:



The following query selects the "Sales Head Office" record and its sub-components:

```
SELECT Self FROM amEmplDept WHERE (FullName LIKE '/Sales Head Office/Sales/%') AND (sLvl >= 1)
```

The following query selects the "Sales Head Office" record but not its sub-components:

```
SELECT Self FROM amEmplDept WHERE (FullName LIKE '/Sales Head Office/Sales/%') AND (sLvl = 1)
```

The following query selects the sub-components of the "Sales Head Office" record but not the "Sales Head Office" record itself:

```
SELECT Self FROM amEmplDept WHERE (FullName LIKE '/Sales Head Office/Sales/%') AND (sLvl > 1)
```

## Simplified AQL notation

This section list the notation which can be used to simplify the writing of AQL queries:

### Foreign keys

In clauses other than the SELECT and ORDER BY clauses, the SQL name of a link which is not followed by a period is equivalent to the SQL name of the associated foreign key.

Example: the clause:

```
WHERE location = 0
```

Is equivalent to:

```
WHERE lLocaId = 0
```

Where "location" is the SQL name of the "Location" link from the table of departments and employees to the table of locations, and "lLocaId" is the SQL name of the associated foreign key in the table of assets.

### Description string

In SELECT and ORDER By clauses, the SQL name of a link which is not followed by a period is equivalent to the join <SQL name of link>.self, which is in turn equivalent to <SQL name of link>.<Description string>.

Example:

If the description string of the table of departments and employees is:

```
Name", "FirstName" ("Phone")"
```

Then the AQL query:

```
SELECT user FROM amAsset
```

Is equivalent to the query:

```
SELECT user.self FROM amAsset
```

Which is itself equivalent to:

```
SELECT User.Name + "," + User.FirstName + "(" + User.Phone + ")"  
FROM amAsset
```

## Features

AQL gives you direct access to the features of a table, as if they were direct fields in the table. To search feature values for a given table, you just need to write the SQL name of the feature and prefix this by "fv\_".

Example: The following query searches the values of the feature with SQL name "XXX" for the table of assets:

```
SELECT fv_XXX FROM amAsset
```

## Calculated fields

AQL facilitates the use of calculated fields associated with a table.

You just need to write the SQL name of the calculated field and prefix this by "cf\_".

---

# Sorts and indexes

AQL allows two strategies for queries which use sorts (ORDER BY clause):

- A mode whereby AssetCenter forces the indexes indicated in the query, when used, and displays the results as and when they are retrieved.
- A mode whereby AssetCenter does not force the indexes indicated in the query. In this case, the DBMS determines how the data is sorted.

Note: The choice between these two different methods is not available under SQL Anywhere. The database engine automatically selects the most suitable method.

## Example

In the case of the following query:

```
SELECT lAstId, Brand FROM amAsset ORDER BY Brand
```

- Access without "forcing the indexes": the database engine scans the table in full without using the "Brand" index indicated in the query. It searches all data items satisfying the query, sorts them according to the "Brand" and sends them to the user. The results will only be displayed after a certain period of time.
- In the other case: The database engine uses the "Brand" index, and displays the results as and when they are found. The first data items are thus shown rapidly, but the overall processing time is longer.

## How to force the indexes

The way in which you force the utilization of the indexes depends on the way in which you create the query.

### Via the Configure list menu

You can configure the data-access type for each list in AssetCenter. This is the case for both main lists and list contained in tabs. To do this:

- ❖ Go to the list you want to configure.
- ❖ Right-click to display the popup menu.
- ❖ Select **Configure list**.
- ❖ In the **Columns and sort** tab, check the **Force indexes** box in order to use the indexes indicated in the query and display the results as and when they are generated; Uncheck this box in order to select the other type of access.

### Using AQL

If you write a query directly in AQL, you can force the use of indexes via the "FIRST\_ROWS" clause.

Example:

```
SELECT FIRST_ROWS AssetTag FROM amAsset ORDER BY AssetTag
```

## Sort order

The sort order depends on:

- The database engine
- The use of indexes

## Under Oracle

### With indexes forced

- ❖ NULL records do not appear.
- ❖ The sort is performed according to the value of the ASCII codes thus differentiating between upper and lower case characters (binary sort).

### Without indexes forced

- ❖ NULL records appear.
- ❖ Oracle is not case sensitive.

### Example

<b>Starting list</b>	A B C D a b NULL NULL
<b>List with indexes forced</b>	A B C D a b
<b>List without indexes forced</b>	NULL NULL A a B b C D

## Under Microsoft SQL Server or Sybase SQL Server

The sort order depends on a parameter set when the database is created. These engines can be configured in order to be case sensitive or to take accented characters into account, etc.

## Under Sybase SQL Anywhere

Under Sybase SQL Anywhere, the indexes cannot be forced via an AQL query.

The database engine determines the optimal method used to access the data and to sort it.

## Precautions

With complex queries it is often difficult to determine whether it is more "advantageous" to force the indexes or not. In practice, we recommend performing tests before making a final decision.

In particular, we recommend testing with and without the indexes forced in the case of a list which is filtered, be it explicitly (via a simple filter, query) or implicitly (via access restrictions).

---

## The query editor

AssetCenter includes a query editor. This tool allows you to design queries and preview their results.

This section details the functioning of the query editor:

- Principle
- Accessing the query editor
- Creating a query using the query editor
- Fields used in a query
- Writing an expression
- Constants

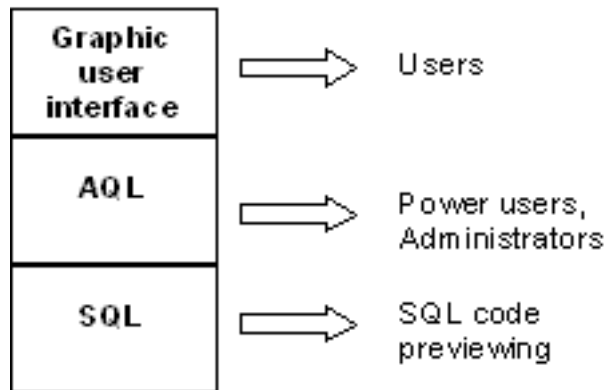
### Overview

The query editor makes it possible to design queries:

- Either by using the graphic interface (assisted query design),
- Or by writing the query directly in AQL.



Whether you use the graphic method or prefer to write directly in AQL (both approaches are frequently combined), you get to see a real-time transcription of your query in SQL. However, you cannot write your queries directly in SQL.



*Intended uses of the query editor*

Using the query editor, a power user or an administrator can create, modify or delete AQL queries. These queries can be used in the appropriate context by their author or other users.

## Accessing the query editor

You can access the query editor in different ways:

- Via the **Tools/ Queries** menu item. Using this menu, you can create queries for your own use and which can also be used freely by other users. The queries can be executed:
  - ❖ Either directly via the window displayed by the **Tools/ Queries** menu item.
  - ❖ Or by when using a "query filter" when displaying the main table of the query.
- Via the numerous functions of AssetCenter which call on queries: Access restrictions, query filters, list configuration, etc.
- Via external programs: AssetCenter Export, etc.

The version of the query editor which is shown is simplified to a certain degree according to the context.

Example: Let's suppose that you have a query such as the following:

```
SELECT [FIRST_ROWS] <field>[, <field>...] FROM <table> [WHERE <clause>]
[ORDER BY <clause>]
```

In the simplified versions of the query editor (simple filters, query filters, etc.), you only have to define the WHERE clause of the query. The other components of the query (starting table, fields, etc.) are implicit. For example, in the case of a query filter, the table is that on which the filter is applied, the fields and the sort conditions are the columns and the sort conditions which are defined via the **Configure list** popup menu item. The same is true for the query editor accessed via the **Tools/ Queries** menu item.

Thus, the following query given in full:

```
SELECT self FROM amAsset WHERE Brand='Compaq'
```

is written as follows when used in a query filter (only the WHERE clause is explicitly given) used on the table of assets:

```
Brand= 'Compaq'
```

On the other hand, the **Configure list** command allows you to access a more comprehensive version of the query editor:

- The **Columns and sort** tab defines the fields to be displayed in columns and the sort conditions (these sort conditions correspond to the ORDER BY clause).
- The **Force indexes** box replaces the FIRST\_ROWS clause in the SQL code.
- The **Filter (WHERE clause)** tab defines the "WHERE" clause.
- The table is implicit.

## Creating a query using the query editor

To create a query using the query editor, select the **Tools/ Queries** menu item. The window which is displayed has two tabs, **Filter (WHERE clause)** and **Preview**:


- The **Filter (WHERE clause)** tab is a graphical interface which determines the conditions of your query. It defines the elements of the WHERE clause.
- The **Preview** tab displays the transcription of your query in SQL code and allows you to test it.

## Step 1: Populate the fields at the top of the query detail.

You must specify the starting table of your query.

If you want the query you are creating to be able to be accessible to other users, uncheck the **Not shared** option (SQL name: bPrivate).

Note: The administrator has access to all queries stored in the database, even those queries that are **Not shared**.



Once you have filled in the basic information on the query, click  to be able to access the detail tabs of the query.

## Step 2: Define the filter conditions in the Filter (WHERE clause) tab.




The AssetCenter query editor allows you to define conditions which combine fields, calculation expressions, constants and operators.


You can define one or more filter conditions.

To define a filter condition:

- From a start table, select a field, constant or expression (**Field 1**), that you will compare with a field, constant or expression (**Field 2**).
- Confirm the filter condition by transferring it to the lower part of the screen using the  button.
- Confirm the query by clicking  in the query detail.

To define several filter conditions linked by the AND and OR logical operators:


- Create a first filter condition as indicated above.
- Define the other filter conditions and confirm them using the  or  buttons.
- Confirm the query by clicking  in the query detail.

Note: In order to make a modification to the selected conditions, click the  button to delete the contents of the window, or else modify the AQL code directly.

Note: In place of the graphic tool, you can enter your query directly in AQL in the zone at the bottom of the **Filter (WHERE clause)** tab.

### Step 3: Preview the execution of the query

To test the query and see its transcription in SQL language:

- Go to the **Preview** tab in the query detail.
- Click the  icon: AssetCenter shows a preview of the results of the query, in the form of a list of records. The number of records satisfying the query is shown at the bottom right of the window.


Note: The SQL code contained in the **Preview** tab cannot be modified directly.

### Fields used in queries

When defining filter conditions in queries, you can call on:


- A field in the table concerned by the query.
- A linked field.
- Features associated with the table.

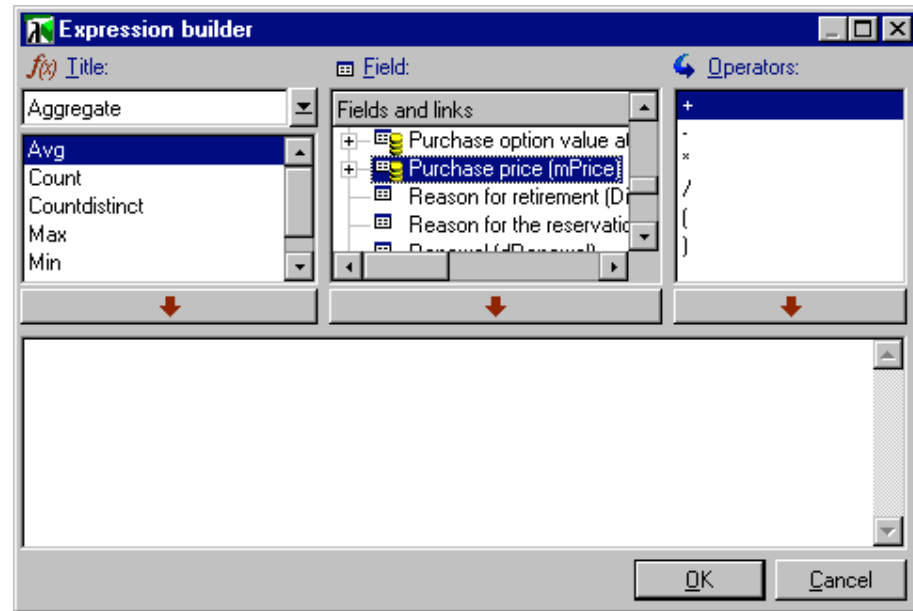
### Writing expressions

Expressions  allow you to perform calculation operations in your query. For example, you can use the "Count" function to count the number of resultant records of a query.


To write an expression, you can:

- Either enter it directly in the corresponding field,
- Or use the AssetCenter expression builder.

To use the expression builder, click the  button adjacent to the edit zone of the **Filter (WHERE clause)** tab of the query detail.




The expression builder comprises three columns:


- The "Function" column list existing AQL functions. Clicking  applies a filter on the list of AQL functions according to their type: "Aggregate", "String", "Date", "Numeric", "Test".
- The "Field" column list the different fields which can be used in a query.
- The "Operators" column lists the operators which can be used in the expression.

To insert a "Function", "Field" or "Operator" in the expression:

- Select the function, field or operator.
- Click .


Once the expression is defined, click  to transfer it over to the **Filter (WHERE clause)** tab in the query detail.


## Constants

Constants  are fixed values, which you assign to selection criteria. For example, if you are searching all assets with brand "3Com", you assign

the constant value "3Com" to the **Brand** field (SQL name: Brand) in the table of assets.

To select constant:

- Click the  icon.
- A selection window is displayed which shows the values present in the database for the field specified as search condition.

Note: Even in the case of "Itemized list" type fields, the window displayed by clicking the  icon only shows those values used in the database.

---

## AQL syntax

This section details the syntax of AQL:

- Conventions
- Syntax of queries
- FROM clause- Tables involved in a query
- Elements of a query
- WHERE clause
- GROUP BY clause
- HAVING clause
- ORDER BY clause

Important note: Using AQL requires familiarity with SQL language. However, a detailed description of the syntax of SQL is beyond the scope of this manual. For further information, please consult the appropriate reference documentation.

### Conventions

[ ]	These brackets surround optional items. Do not type in these brackets.
< >	These brackets surround logical items. Do not type in these brackets.
	The vertical bar indicates that choices are mutually exclusive.
...	The ellipsis indicates that the preceding text may be repeated once or several times.
FROM	Terms in uppercase letters indicate literal expressions.

## Syntax of queries

### Simple queries

```
SELECT [DISTINCT] [FIRST_ROWS] <selection list>
[FROM <clause>]
[WHERE <clause>]
[GROUP BY <clause>]
[HAVING <clause>]
[ORDER BY <clause>]
```

### Sub queries

AQL supports the use of sub-queries in the place of fields.

Note: In sub-queries, the SELECT statement only authorizes a single expression.

```
( SELECT [DISTINCT] <expression>
[FROM <clause> ]
[WHERE <clause> ]
[GROUP BY <clause>]
[HAVING <clause>]
)
```

Attention: Sub-queries must be contained with parentheses.

Example of utilization:

```
SELECT Self FROM amAsset WHERE dPrice >= (SELECT Max(dPrice)/2 FROM
amAsset)
```

## UNION type queries

UNION allows you to group together the results of several queries:

```
SELECT <selection list>
    [FROM <clause>]
    [WHERE <clause>]
    [GROUP BY <clause>]
    [HAVING <clause>]
[ UNION | UNION ALL | INTERSECTS | MINUS
SELECT <selection list>
    [FROM <clause>]
    [WHERE <clause>]
    [GROUP BY <clause>]
    [HAVING <clause>]...]
[ORDER BY <clause>]
```

## FROM clause- Tables involved in a query

The FROM clause indicates the table or tables concerned by a SELECT statement.

## Syntax

AQL authorizes the utilization of aliases for table names.

The FROM clause takes the following form:

```
FROM <table> [<alias>][, <table> [<alias>] ... ]
```

Examples:

```
FROM amAsset
FROM amAsset a, amLocation l
```

## Starting table of a query

The first table indicated in the FROM clause of a query is the starting table of the query.



If a query uses a field whose table is not specified, AQL considers that it belongs to the starting table of the query. The AQL "FROM" clause differs from the clause with the same name in SQL.

For example, in the following sub-query, AQL searches the "AssetTag" field in the "amAsset" table:

```
SELECT AssetTag FROM amAsset, amLocation
```

## Examples of utilization

The following queries are equivalent:

```
SELECT AssetTag FROM amAsset
SELECT AssetTag FROM amAsset a
SELECT a.AssetTag FROM amAsset a
SELECT amAsset.AssetTag FROM amAsset a
SELECT amAsset.AssetTag
SELECT amAsset:AssetTag
```

## Number of tables in a query

The number of tables authorized in a query depends on the DBMS used.

Example:

- Oracle: You can use as many tables as you like.
- Microsoft SQL Server or Sybase SQL Server: You are limited to 16 tables in a query.

Important note: When counting the number of tables in a query, do not forget to take into account those table which are not explicitly mentioned, in particular if the query uses links.

Also look out for the "fv\_" notation (search of feature values) which generates an additional join at DBMS level. Similarly, the "cf\_" notion (calculated fields) can generate additional joins.

## Elements of a query

### Fields and links

Queries involve fields and links in the AssetCenter database.

You can indicate the name of a field:

- In reference to the starting table of the query. In this case, it is not necessary to specify the name of this table:

**[Link. ...[Link.]]<field>**

Examples from the table of assets:

Brand User.Name User.Location.Name
--

- As an absolute reference. In this case, you need to indicate the name of the table to which the field belongs:
  - ❖ Either by declaring the table in the FROM clause and using its name (or a possible alias):

**<table.[link...]field>**

**<alias.[link...]field>**

- ❖ Or by not declaring the table in the FROM clause and instead using ":":

**<table.[link...]field>**

**<table[\_alias]:[link[\_alias]...]field>**

These last two notations are particularly useful if you cannot use the FROM clause.

For example, when writing a query in AssetCenter you only have access to the WHERE clause. The starting table of the query is implicit (table on which a filter is applied, **Table** field (SQL name: TableName) in query detail, etc.). However, you may need to use other tables in the query. In this case, the ":" notation allows you accomplish this.

## Constants

The following syntax is valid for the constants which may be involved in queries.

### Numeric constants

The period is used as decimal separator.

Examples:

12

52.23

### Text type constants

They are contained within single quotes.

Examples:

'Computer'

'Monitor'

### Date or time type constants

Date or time type constants are contained between # characters. Their format must respect the following rules:

- ❖ Years are expressed on 4 figures.
- ❖ Dates are expressed as Year-Month-Day.
- ❖ Time is expressed as Hours-Minutes-Seconds.
- ❖ The 24 hour clock is used (and not the 12 hour clock with A.M. or P.M.).
- ❖ The separator used in dates is the "/" or "-" character.
- ❖ The separator used for time is the ":" character.
- ❖ Months, days, hours, minutes, and seconds; these are expressed as 2 figures.
- ❖ When date and time are expressed together, the data always precedes the time and they are separated by a space.

Examples:

#yyyy-mm-dd hh:mm:ss#

#yyyy-mm-dd#

#hh:mm:ss#

#1999-01-01 01:00:03#

## Expressions

Expressions are formed using:

- Constants,
- Fields,
- Functions,
- Sub-queries.

You can combine these elements with operators and parentheses in order to build complex expressions.

Comparison expressions take the form:

**<expression> <comparison operator> <expression>**

Logical expressions take the form:

`<comparison operator> <AND | OR> <comparision expression>`

You can use parentheses to combine several logical expressions.

## Operators

### Logical operators

Logical operators are applied to link two expressions:

Operator	Meaning
AND	Logical "AND"
OR	Logical "OR"

In order to optimize a query, it is sometime wise to avoid logical operators if a comparison operator can be used instead. The following example illustrates how to optimize a query filter used to select assets whose **Assignment** field (SQL name: seAssignment) is set to **Awaiting delivery** or **Return for maintenance**. The values of these two elements of a system itemized list are "3" and "4" respectively. It is therefore possible to write:

```
(seAssignment=3) OR (seAssignment=4)
```

The last value of the system itemized list being "4", it is preferable to write the query as follows:

```
seAssignment>=3
```

### Comparison operators

Comparison operators are used to compare two expressions.

Operator	Meaning
=	Equals
<> =!	different than
>	greater than
<	less than
>=	greater or equal to
=<	less than or equal to
=*	right outer join. Because of the way in which AQL handles links, the use of this operator is limited
*=	left outer join. Because of the way in which AQL handles links, the use of this operator is limited.
LIKE NOT LIKE	<p>Works like the = operator and allows you also to use wildcard characters.</p> <p>The following wildcard characters are available:</p> <p>"%" replaces any character string.</p> <p>"_" replaces any single character.</p> <p>Depending on the database engine used (SQL Anywhere, SQL Server and Sybase support it, Oracle doesn't):</p> <p>[abc...] lets you define a list of possible characters (no space between the possible values.)</p> <p>[a-c] lets you define a range of possible values.</p> <p>DB2 does not support use of the LIKE X operator, if X includes a SQL column name. Only constants are supported for this operator. For example, the following query is not correct for DB2:</p> <pre>SELECT COL1, COL2 FROM TABLE1 WHERE COL1 LIKE COL2</pre>
IS NULL IS NOT NULL	<p>Tests whether the value of a field is "NULL" or not.</p> <p>Important note: AssetCenter only authorizes the "NULL" value for empty text fields and for "Date" or "Date+Time" fields which are not populated.</p>

Note: SQL Anywhere is not able to process "LIKE X" clauses when X contains more than 128 characters. If X is larger than 128 characters applying the query provokes an ODBC error message. This problem can, for example, occur when displaying lists in tree view since this operation uses a "LIKE" clause on a "FullName" field.

## Operators specific to sub-queries

You can compare a value to the results of a sub-query using the following operators:

- ❖ = **ANY** (sub-query)
- ❖ = **ALL** (sub-query)
- ❖ = **SOME** (sub-query)

Example:

- The following query provides the list of assets whose brand is used at the San Mateo site:

```
SELECT lAstId, Brand FROM amAsset WHERE Brand = ANY (SELECT Brand  
FROM amAsset WHERE location.fullName='/San Mateo')
```

## Selection list

Selection lists define the items to be extracted or displayed. They specify the SELECT statements in queries.

A selection list is made up of one or more expressions separated by commas:

**<expression> [, <expression>...]**

Each expression can be linked to an alias. For example:

```
SELECT MrMrs, (Name + FirstName) Identity FROM amEmplDept
```

This is particularly useful at the level of export queries to attribute a name to the exported columns.

Important note: Certain DBMSs limit the number of expressions in a given SELECT statement.

## WHERE clause

The AQL "WHERE" clause is equivalent to the "WHERE" clause in SQL. It specifies the search conditions.

**WHERE <Search conditions>**

The search conditions specify the items to be extracted from the database and can be expressed in the WHERE or HAVING clauses.

In the majority of cases, you will need to write the conditions using the following form:

```

<WHERE | HAVING> [NOT] <expression> <comparison operator> <expression>
<WHERE | HAVING> [NOT] <logical expression>
<WHERE | HAVING> [NOT] <field> [NOT] LIKE 'xxxxx'
<WHERE | HAVING> [NOT] <logical expression> <AND | OR> <logical
expression>
<WHERE | HAVING> [NOT] <field> IS [NOT] NULL

```

In some other case, you may need to write more complex queries, such as:

```

<WHERE | HAVING> [NOT] EXISTS (<sub-query>)
<WHERE | HAVING> [NOT] <expression> [NOT] IN (<list of values> | <sub-
query>)
<WHERE | HAVING> [NOT] <expression> <comparison operator> <ANY | ALL>
(<sub-query>)

```

## GROUP BY clause

The AQL "GROUP BY" clause is equivalent to the "GROUP BY" clause in SQL.

**GROUP BY <expression without aggregates>  
[, <expression without aggregates>]...**

"GROUP BY" specifies subsets of the table. The subsets are defined in the GROUP BY clause by an expression, which can be the name of a field, for example.

If aggregate functions are included in the selection list of the SELECT statement, "GROUP BY" searches the resulting value for each subset. These resultant values can be used in a HAVING clause.

When a query makes use of the GROUP BY clause, each expression of the selection list must provide a single value for each subset.

Examples of queries with and without the "GROUP BY" clause:

The following query gives the total number of brands present in the database. For each asset with an associated brand, AssetCenter returns an instance on the brand.

```
SELECT Count(Brand) FROM amAsset
```

By using the GROUP BY clause, we obtain a list of brands and the number of assets of each brand:

```
SELECT Brand, count(lAstId) FROM amAsset GROUP BY Brand
```

## HAVING clause

The AQL "HAVING" clause is equivalent to the SQL "HAVING" clause.

### **HAVING <Search conditions>**

The "HAVING" clause specifies the search conditions like the "WHERE" clause. However, these two clauses differ as follows:

- The "HAVING" clause specifies the restrictions to be applied to aggregate functions in the selection list. The restrictions affect the number of resultant lines but do not affect the calculations linked to aggregate functions.
- When the query uses an WHERE clause, the search conditions restrict the lines subject to aggregate calculation functions but do not affect the resultant lines.

Example of query where the "WHERE" clause is equivalent to the "HAVING" clause:

The following query returns the list of brands whose name starts with a letter after the letter "B" and the number of asset of each of these brands:

```
SELECT Brand, count(lAstId) FROM amAsset GROUP BY Brand HAVING Brand > 'B'
```

It is also possible to express the same query using the "WHERE" clause:

```
SELECT Brand, count(lAstId) FROM amAsset WHERE Brand > 'B' GROUP BY Brand
```

Example of query using the "HAVING" clause:

The "HAVING" clause allows you to use aggregate functions (such as "Count"); This is not the case with the "WHERE" clause. Thus, the following query searches all brands represented by more than one asset:

```
SELECT Brand, count(lAstId) FROM Asset GROUP BY Brand HAVING count(Brand) > 1
```

## ORDER BY clause

The AQL "ORDER BY" clause is equivalent to the SQL "ORDER BY" clause.

**ORDER BY <expression> [ASC | DESC] [,<expression> [ASC | DESC]...]**

Items can be sorted:

- In ascending order: ASC. This is the default sort order.



- In descending order: DESC.

---

## AQL function reference

The following AQL functions can be used in queries and formulas:

- Aggregate type AQL functions
- String type AQL functions
- Date type AQL functions
- Numeric type AQL functions
- Test type AQL functions

Note: You can also use native SQL functions of your DBMS. In this case, the resulting code is not portable.

### "Aggregate" type AQL functions

Function	Description
Avg( <column> )	Returns the average value of a "number" type column. Returns "0" if the column does not have any records.
Count( <column> )	Counts the non-null values in a column.
Countdistinct( <column> )	Counts the distinct non-null values in a column.
Max( <column> )	Returns the maximum value in a "number", "string" or "date" type column.  If the column does not have any records, returns "0" ("number" type column), "empty string" ("string" type column), or "empty date" ("date" type column).
Min( <column> )	Returns the minimum value in a "number", "string" or "date" type column.  If the column does not have any records, returns "0" ("number" type column), "empty string" ("string" type column), or "empty date" ("date" type column).
Sum( <column> )	Returns the sum of the values of a "number" type column. Returns "0" if the column does not have any records.

These functions jointly use the "GROUP BY" and "HAVING" clauses.

## "String" type AQL functions

Function	Description
Ascii( <String> )	Returns the ASCII value of the first character of the <string>.
Char(<n>)	Returns the character with ASCII code "n".
Left( <String>, <n> )	Returns the "n" first characters of the <string>.
Lower( <String> )	Returns the <string> in lowercase.
Ltrim( <String> )	Removes the spaces at the left of the <string>.
Right( <String>, <n> )	Returns the "n" last characters of the <string>.
Rtrim( <String> )	Removes the spaces at the right of the <string>.
Substring( <String>, <n1>, <n2> )	Extracts the sub-string starting at character "n1" in the <string> and with length "n2" (the 1st character of the <string> is considered as character number 1).
Upper( <String> )	Returns the <string> in uppercase.

## "Date" type AQL functions

Function	Description
Year( <date> )	Returns the number representing the year for a "date" or "date and time type field" (e.g: 1997).
Month( <date> )	Returns the number of the month for a "date" or "date and time type field" (1=January, ..., 12=December).
Day( <date> )	Returns the number of the day in the month for a "date" or "date and time" type field (1-31).
DayOfYear( <date> )	Returns the number of the day in the year for a "date" or "date and time" type field (1-366).
WeekDay( <date> )	Returns the number of the day in the week for a "date" or "date and time" type field.  This number depends on how the server is configured. For example, the default configuration under Sybase or Microsoft SQL Server is (1=Sunday, 2=Monday, ..., 7=Saturday). The default configuration under Oracle is (1=Monday, ..., 7=Sunday).
Hour( <hour> )	Returns the number of the hour in the day for a "time" or "date and time" type field (0-23).
Minute( <hour> )	Returns the number of minutes for a "time" or "date and time" type field (0-59).
Second( <hour> )	Returns the number of seconds for a "time" or "date and time" type field (0-59).
Getdate()	Returns the server's current system date.
AddDays( <date>, <number> )	Adds a given number of days to a "date" or "date and time" type field.
AddHours( <date>, <number> )	Adds a given number of hours to a "date" or "date and time" type field.
AddMinutes( <date>, <number> )	Adds a given number of minutes to a "date" or "date and time" type field.
AddSeconds( <date>, <number> )	Adds a given number of seconds to a "date" or "date and time" type field.
DaysDiff( <date1>, <date2> )	Number of days between the dates date1 and date2 ("floating point" number with decimals)
HoursDiff( <date1>, <date2> )	Number of hours between the dates date1 and date2 ("floating point" number with decimals)
MinutesDiff( <date1>, <date2> )	Number of minutes between the dates date1 and date2 ("floating point" number with decimals)
SecondsDiff( <date1>, <date2> )	Number of seconds between the dates date1 and date2 ("floating point" number with decimals)
DbToLocalDate( <date> )	Converts a date expressed in the time zone of the

Function	Description
)	database server to a date expressed in the time zone defined at client machine level.
LocalToDbDate( <date> )	Converts a date expressed in the time zone of the client machine to a date expressed in the time zone of the database server.

Examples:

Description	AssetCenter query language
All records modified during the last week	AddDays( dtLastModif,7 )>=Getdate()
All work orders notified in the last hour	HoursDiff( Getdate(), dtNotif ) <= 1 or AddHours( dtNotif, 1 ) >= Getdate()
All work orders closed in the last half-hour	MinutesDiff( Getdate(), dtActualFixed ) <= 30 or AddMinutes( dtActualFixed, 30 ) >= Getdate()

The following query lists the work orders effectively carried out and resolved during the same day. The time zone of the client machine is taken into account:

```
SELECT Self FROM amWorkorder WHERE
DayOfYear(DbToLocalDate(dtActualFixStart)) =
DayOfYear(DbToLocalDate(dtActualFixed))
```

The following query lists all work orders which have effectively been started today:

```
SELECT Self FROM amWorkorder WHERE
DayOfYear(DbToLocalDate(dtActualFixStart)) =
DayOfYear(DbToLocalDate(GetDate()))
```

## "Numeric" type AQL functions

Function	Description
Abs( <Number> )	Returns the absolute value of a "number".
Ceil( <Number> )	Returns the smallest integer greater or equal to a "number".
Floor( <Number> )	Returns largest integer less than or equal to a "number".
Mod( <a>, <b> )	Returns the remainder of the division of "a" by "b" (a = qb + r, with q integer and $0 \leq r < q$ ).
Round( <a>, <n> )	Rounds "a" to "n" decimal places.
Trunc( <a>, <n> )	Truncates "a" to "n" decimals.

Examples of application:

Abs (2.516) = 2.

Ceil (2.516) = 3.

Floor (2.516) = 2.

Mod (6,4) = 2.

Round (31.16, 1) = 31.20.

Round (31.16, 0) = 31.00.

Round (31.16, -1) = 30.00.

Trunc (31.16, 1) = 31.1.

## "Test" type AQL functions

Function	Description
IsNull( <a>, <b> )	If "a" is "Null", replaces "a" by "b". The types of "a" and "b" must be compatible.

## Examples of queries

The following examples of queries each deal with a specific aspect of query design. You can modify or combine these examples to use them as the basis of your own queries.

These examples give the full syntax of the query. If you want to test them out as is, use AssetCenter Export. You will have to modify the syntax of these examples to use them in a query filter, for example.


Thus, the following query given in full:

```
SELECT self FROM amAsset WHERE Brand='Compaq'
```

is written as follows when used in a query filter (only the WHERE clause is explicitly given) used on the table of assets:

```
Brand= 'Compaq'
```

Further examples of queries are included in the demonstration database supplied with AssetCenter.

Note: To view the transcription of a query in the corresponding SQL code of the DBMS you are using,  display the **Preview** tab in the query detail.

### To compare a field in the main table with a value

Example: All "Compaq" brand assets.

```
SELECT Self FROM amAsset WHERE Brand = 'Compaq'
```

### To compare a link in the main table with another link

Example: All assets with the same location as their parent asset.

```
SELECT Self FROM amAsset WHERE Location = Parent.Location
```

### To compare a link in the main table with a value

Example: All departments and employees directly linked to the "Burbank Agency".

```
SELECT Self FROM amEmplDept WHERE Parent.Name = 'Burbank Agency'
```

### To compare according to a field in a table linked to the main table

Example: All assets which have the same location name as their parent.

```
SELECT Self FROM amAsset WHERE Location.Name = Parent.Location.Name
```

## Hierarchic tables

### Utilization of "FullName" field

Example: All sub-locations of the location named "Ariane Building":

```
SELECT Self FROM amLocation WHERE FullName LIKE '/Ariane Building/%'
```

### Utilization of "FullName" and "sLvl" fields

Queries on the hierarchic tables often make use of the "FullName" and "sLvl" fields.

Example: All the sub-locations of the location "Ariane Building", with a hierarchic level less than 3

The hierarchic value of the root level of a tree-structure is equal to "0".

```
SELECT Self FROM amLocation WHERE (FullName LIKE '/Ariane Building/%') AND (sLvl < 3)
```

Pay special attention to the "/" characters which appear at the start and end of full names.

### Query combining two conditions

Example: All employees with title "Account executive" and located at the "Burbank site".

```
SELECT Self FROM amEmplDEpt WHERE (Title = 'Account executive') AND (Location.Name = 'Burbank Site')
```

### Comparison of a field with numbers, dates or text

Example: All work orders carries out between January 1, 1995 and December 31, 1995.

```
SELECT Self FROM amServiceCall WHERE (dtFirstCall >= #01/01/95 00:00:00#) AND (dtFirstCall <= #12/31/95 00:00:00#)
```

### Query concerning a feature

Example: All assets whose feature with SQL name "Size" showing a value greater than or equal to 150 cm.



```
SELECT Self FROM amAsset WHERE fv_Size >= 150.00
```

### To search records according to an expression

Example: All assets whose purchase price is equal to the greatest purchase price contained in the database. Note that a sub-query is used in the main query in order to identify the maximum price.

```
SELECT Self FROM amAsset WHERE mPrice = (SELECT max(mPrice) FROM amAsset)
```

### To search a field which is not populated

Example: All employees without a telephone number. Note that an empty string is represented by two single quotes ' '.

```
SELECT Self FROM amEmplDept WHERE Phone=' '
```

### To search for the absence of a link

#### Case of a 1 link

Example: All assets which have not been assigned to a user. Note that the absence of a link is denoted by "0".

```
SELECT Self FROM amAsset WHERE User = 0
```

#### Case of n links

Example: All categories with no associated assets:

```
SELECT Self FROM amCategory WHERE 0 = (SELECT COUNT(a2.lAstId) FROM amAsset a2 WHERE a2.lCategId = lCategId)
```

This query scans the table of categories, takes each category on after the other and compares with 0 the number of assets belonging to this category.

#### Example combining a test on a 1 link and an n link

Example: All assets without parent asset or sub-asset:

```
SELECT Self FROM amAsset WHERE (0 = (SELECT COUNT(a.lAstId) FROM amAsset a WHERE a.lParentId = lAstId)) AND (Parent = 0)
```

This query performs:

- ❖ A test on a 1 link ("Parent = 0"), to select those assets without parent asset.
- ❖ A test in an n link ("0 = (SELECT COUNT(a.lAstId) FROM amAsset a WHERE a.lParentId = lAstId)", to select those assets without sub-assets. The test on the n link takes each asset, selects its identifier "lAstId", and counts all the assets whose "lParentId" identifier is equal to "lAstId".

### Another example

All assets without "Hard Drive" category sub-component:

```
SELECT self FROM amAsset p WHERE NOT ( EXISTS (SELECT lAstId FROM
amAsset WHERE (FullName LIKE (p.FullName + '%/')) AND
(Category.Name = 'Hard Drive')))
```

### Query with alias

Example: All employees having taken the 'Peregrine' training program and the 'Database' training program.

Start table: The table of departments and employees.

The query is as follows:

```
SELECT Self FROM amEmplDept WHERE (Training_1.Title = 'Peregrine') AND
(Training_2.Title = 'Database')
```

Aliases, expressed as "Training\_1" and "Training\_2" allow you to define 2 conditions concerning the 2 different records linked by the "Training" link.

If we had written:

```
SELECT Self FROM amEmplDept WHERE (Training.Title = 'Peregrine') AND
(Training.Title = 'Database')
```

We would have selected all employees having taken a training course with both titles.

If we had written:

```
SELECT Self FROM amEmplDept WHERE (Training.Title = 'Peregrine') OR
(Training.Title = 'Database')
```

We would have selected all employees having taken one training course with one of the two titles.

# Chapter 8 - Editing forms

This section explains how to design forms with AssetCenter.

You will find information on the following topics:

- Definition of a form
- Creating forms
- Editing forms and objects
- Properties of object in forms
- Design grid
- Form page setup
- How to easily create regular reports

*Use the File/ Forms menu item to display the list of forms.*

---

## Definition of a form

A form is a document model that lets you print data.

Unlike Seagate Crystal reports, forms are created directly in AssetCenter.

---

## Creating forms

*Use the File/ Forms menu item to display the list of forms.*

## Basic information

- Enter the name of the form
- Select the type of form: (list or detail)

Both types can contain text and predefined images.

Differences between the two types:

- ❖ list: allows you to print a list of records such as it is displayed in the active list (according to the columns the list contains and the filters applied).
  - ❖ detail: allows you to print fields from a record detail (example: the detail of an asset), and linked record lists (example: the components of this asset).
- Select the main table for the form

Warning: The **Table** field (SQL name: TableName) lets AssetCenter propose only those forms relevant to the given list.






---


## Editing forms and objects

To edit a form you define objects and position them on the page.

To insert a new object on the page:

- Click the object's icon at the left of the page.

	To select an object in the form in order to modify it, for example.
	To add fixed text and variables independent of the records being printed (current date, for example).
	To add an image.
	To insert a formula containing field values and fixed text strings.
	To insert a list of records. This tools allows you to position the list on the page. For detail forms, also allows you to define the linked table containing the records and list of fields to be printed.

- Position the mouse pointer on the page.
- Click the left mouse button.
- Trace a frame with the mouse: This frame defines the space reserved for the object.
- Double-click the object's position. A window describing the object's properties is displayed.
- Define the properties of the object.
- Click .

You can insert the following objects in a form:

## Fixed text

This concerns text which is independent of the records being printed. You can combine any kind of character, as well as variables:

\$D: Date printed.

\$U: Name of the AssetCenter user who is printing the form

\$C: Page number.

\$N: Total number of pages printed.

**Important note:** Do not put quotes around the text.

Example:

Document printed on \$D by \$U

## Formulas

Formulas are only available in "detail" type forms.

Formulas bring together:

- field values from the AssetCenter database.
- fixed text surrounded by quotes.

Example:

```
"Asset:" AssetTag " / " Brand
```

Formulas do not allow you to perform calculations.

## Lists

- "List" type forms: You can only include one list. This list will be replaced by the list in the active window when you select the **File/Print** menu item.
- "Detail" type forms: The number of lists is not limited. Lists display all records related to the current record. For example: All the components of an asset.

## Images

You can insert images (logos, etc.).

Note: When you draw up a "list" type form you cannot select the fields to print. AssetCenter prints the fields that appear in form of columns in the lists.

When you draw up a "Detail" form, you can select the fields to be printed one by one.

---

# Properties of object in forms

## Position and size

To modify the position and size of an object, simply drag the object or its edges.

Note: You can move and re-size several objects at once by selecting them simultaneously. If you draw a selection frame around several objects with the mouse, all objects in the frame will be selected; or select the objects one by one, while holding down the SHIFT key.

## Properties

If you double-click an object, or use the **Internal forms/ Properties** menu item, a list of the selected object's properties is displayed.

The property list has two columns: The first contains property names and the second lets you edit the property values.

To modify a property, click in the second column with the left mouse button.

Simple properties may be edited directly (text, formulae, lists, background colors, text colors, text alignment, object alignment.)

Complex properties (font for the text or formula, frame, list contents, image) display additional windows.

### Text

Enter your text directly.

### Formula

To help you in creating formulas, you can display a drop down list with a tree that displays the fields that are accessible and compatible with the type of form selected. Clicking a node in the tree replaces the current selection with the field you selected in the list. You can insert text between fields as long as you include quotes.

### Background color, text color

You may choose these colors from 16 available colors.

### Text alignment


This determines how text is aligned within the frame. A drop-down list displays the different kinds of alignment (Centered, Left aligned, Right aligned.)

## Page alignment


This determines the text's horizontal alignment in the page. A popup window lets you select the alignment:

- ❖ **Left aligned.**
- ❖ **Right aligned.**
- ❖ **Centered.**
- ❖ **Relative alignment:** In this case, the object keeps the position you defined in the form creation screen.

## Font

To select the character set of the object and its size, click  next to cell being edited.

## Frame

To add a border around objects, click the  button in the edit cell: A configuration screen for the border is displayed.

The 3D style gives an embossed effect to the frame.

For borders without the 3D style, you can select which borders will be displayed, their color and thickness.

## Image

To insert an image:

- ❖ Click the button in the edit cell.
- ❖ Select the graphic file in the "Select image" screen.

## Link in list

To select a list to be displayed in the form, use the drop down list. Click the list you want (for example, the list of assets used by an employee).

## List contents

Note: You can only configure the contents of the list when editing a detail form.

To define the contents of a list, click the button in the edit cell. A configuration window is displayed. All columns in the list are shown.

This screen lets you define:



- ❖ the title of each column,
- ❖ the formula used to define the contents of the column,
- ❖ the size of the column, (percentage used by column)
- ❖ the alignment for the title and contents of each column,
- ❖ the alignment of each title or contents of each column,
- ❖ the vertical and horizontal separators.

To remove a column from the list, use the "Delete" key.

To insert a column in the list, edit the last line of the list.

Each cell can be edited using the same principles as for the list of properties.

---

## Design grid

The design grid is made up of vertical and horizontal lines which cover the background of the screen.

The **Form/ Grid** contextual menu item allows you to:

- show and hide the design grid.
- define the spacing between lines in the grid.

Only the points of intersection between horizontal and vertical lines are shown. The spacing between lines determines the precision with which you can position objects on the page.

---

## Form page setup

The **Form/ Page setup** contextual menu allows you to define:

- The page format.
- **Portrait** or **Landscape** layout.
- Document margins.
- Header and footer.

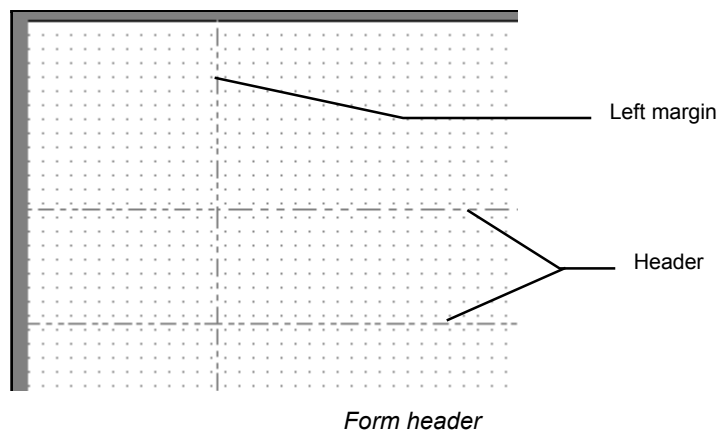
---

Note: Document margins, footer and header zones can be directly modified in the edit zone. Simply drag the margin marker and zone limits for the header and footer which are shown in dotted lines.

---

To insert text in page headers and footers:

- Select the **Form/ Page setup** menu item.
- Check the **Header** and **Footer** boxes.
- Click .
- Go to the header or footer zone in the form (header and footer zones are shown with horizontal dotted lines).
- Insert objects here or move objects from the main page zone.
- Confirm your modifications by clicking .



Note: You cannot move objects from the header and footer zones to the main page zone.

---

## How to easily create regular reports

To produce reports you need regularly, we recommend that you:

- create a "view" with the appropriate parameters.
- associate this view with a form.

The view lets you define:

- The sort criteria.
- The filter to apply and the filter values.
- The list of visible columns.

The form lets you organize the report page layout.

To print the report:

- ↳ Display the view you created beforehand (**Tools/ Views** menu item.)
- ↳ Print from the displayed view (**File/ Print** menu item.) Make sure you select the correct **Type** of report and the appropriate **Form**.



# Chapter 9 - Editing reports

This chapter explains how to print reports with AssetCenter.

You will find information on the following topics:

- Operation and installation of the report generator
- Detail of a report
- How to modify a Crystal Report
- Crystal Reports statistics
- Creating a detail report

*Use the File/ Reports menu item to display the list of reports*

---

## Operation and installation of the report generator

### Overview

AssetCenter makes use of Crystal Reports reporting software. These reports have the file extension **.rpt**.

### **You do not need to have Crystal Reports to print existing reports**

A limited version of Crystal Reports is installed with AssetCenter if you check the appropriate option during the installation.

This limited version is sufficient to preview and print existing reports with the current data from the AssetCenter database.

### You need to have Crystal Reports to modify existing reports or to create new ones

AssetCenter obviously cannot let you create these reports directly.

For this you need to install Crystal Reports version 4.5 or 5.

### Installing, configuring and inserting Crystal reports in your database

Please refer to "Installation and Upgrade Guide", chapter "Installing AssetCenter", section "Crystal Reports"


---

## Detail of a report





*Use the File/ Reports menu item to display the list of reports.*


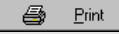
A report detail in AssetCenter is made up of the following information:

### File

You cannot edit this field directly. It indicates the name of the report file (with its extension and the relative path of its folder) that was imported using the  Import...

The following buttons let you work with reports:

-  Import...: This button in the report detail allows you to import (the first time to create the report, following times to modify the report) an external report. External reports have the **.rpt** file extension. Importing an external report updates the **File** field (SQL name: FileName) in the AssetCenter report detail.
-  Export...: This button in the report detail allows you to create an **.rpt** file from a report contained in the AssetCenter database. By default, the dialog box that opens proposes the name of the file contained in the **File** field. You can modify this. Doing this allows you to modify a report using the external reporting program.
-  Preview...: This button, accessible via the **File/ Print** menu item, lets you preview the report on-screen before printing it.
-  Print: This button, accessible via the **File/ Print** menu item, lets you print the report.

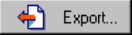
Note: When you press the  or  buttons, AssetCenter creates a temporary file from the report in the database. This file is processed by the Crystal Reports print engine. The temporary file is erased immediately afterwards. The data displayed or printed is the data currently in the database.

---

## How to modify a Seagate Crystal report

In order to modify a report contained in the AssetCenter database, you need to have Crystal Reports.

Use the following procedure:

- ↳ Display the detail of the report using the **File/ Reports** menu item.
- ↳ Click the  button to create a **.rpt** file.
- ↳ Modify the **.rpt** report using Crystal Reports and save it.
- ↳ Display the report detail again using the **File/ Reports** menu item.
- ↳ Import the **.rpt** file to update it and modify the record.

---

## Crystal Reports statistics

To display Crystal reports which are automatically updated, use the **Tools/ Crystal Reports statistics** menu item.

You can display the same reports as when using the **File/ Reports** menu item.

### Nature

Indicate the nature of the report to be displayed. The field to the right of this field allows you to select a given report. The reports which are available depend on the "Nature" you select.



**button**

- Click this button to refresh the report.
- Right-click this button to define the frequency of automatic refreshing of reports.



**button**

Modifies the zoom factor (3 levels).

---

## Creating a detail report

A "detail report" is a report which prints the detail information on one or more records selected in an list.

### Example of utilization

- ↗ Display the list of assets.
- ↗ Select an asset.
- ↗ Select the **File/ Print** menu item.
- ↗ Set the **Type** field to "Detail report (Crystal Reports)".
- ↗ Select the report.
- ↗ Print.

This procedure prints a detail report for each selected record.

### Configuring reports under Crystal Reports

To obtain a detail report, follow the procedure below (example taken under Crystal Reports Professional 5.0):

- ↗ Use the **Insert/ Formula Field** menu item to create a formula field. Its name must respect the following syntax:

<SQL name of the table for which the report is contextual>Id
--

You must respect the case of the SQL names of tables.
---

For example, to create a contextual report on the table of assets, the formula is:



amAssetId


Do not confuse the syntax of the formula field name with the SQL name of the primary key field. For example, the primary key of the table of assets is "lAstId", which is different from "amAssetId".

The "CurrentUserId" formula (respect the case) makes it possible to identify the user printing the report. When printed, this formula takes the value of the identifier number (i.e. the value of the field with SQL name: "lEmplDeptId" for the current login) of the user connected to the AssetCenter database.

If you want to see the result of the report for a given record in the contextual table, edit the formula field and enter the primary key of the table for an existing record in the AssetCenter database.

Example:

512

Edit the formula field in the window which is automatically displayed when you confirm the name of the new formula field. If the formula field exists already, click the  button to edit it.

- ↳ Use the **Report/ Edit Selection Formula/ Record** menu item to edit the selection formula. It uses the following syntax:


```
{< SQL name of the context table>.<SQL name of the field used as  
primary key>} = @<Formula Field name>}
```

The case used for the SQL names of tables and fields is unimportant.

Example:

```
{amAsset.lAstId} = {@amAssetId}
```

Using the above procedure, AssetCenter automatically identifies the report as being contextual when it is imported into the database. You will see this when you perform the following:

- ↳ Access the list of reports using the **File/ Reports** menu item.
- ↳ Create a new report.
- ↳ Import the Crystal Reports file (**.rpt** extension) by clicking the  button.

- ✎ Once this file has been added, you will notice that the **Table** field (SQL name: TableName) shows the SQL name of the context table. If this is not the case, verify the formula field and the selection formula in the Crystal Report.

# Chapter 10 - Read-only access to the database using the AssetCenter ODBC driver

This chapter explains how to use external tools to access (read-only) the AssetCenter database via the ODBC driver developed for AssetCenter.

The AssetCenter ODBC driver only supports read-only access to the database.

In order to write to the database, use the AssetCenter APIs; These are supplied separately to AssetCenter. Write-access to the database is dealt with in more detail in the relevant documentation.

This chapter includes information on the following points:

Overview of access to the AssetCenter database

Example- Creating a report under Crystal Reports with the ODBC driver

---

## Overview of access to the AssetCenter database

### Installing the ODBC driver

The AssetCenter installation program will install the ODBC driver if you:

- Specifically select this package during setup,
- Or if the driver is required by other installed packages.

The ODBC driver is called "Peregrine AssetCenter Driver". There are two versions:

- 16-bit: **Adbc16.dll** file used by a 16-bit ODBC administrator when the AssetCenter client is installed under Windows 3.1.  
**Adbc16.dll** is copied to the Windows "system" folder.
- 32-bit: **Adbc32.dll** file used by a 32-bit ODBC administrator when the AssetCenter client is installed under Windows 95 or NT.  
**Adbc32.dll** is copied to the Windows "system32" folder.

The ODBC driver is independent of the language version of AssetCenter and the DBMS used.

### When to use the ODBC driver

Using this driver is recommended when designing reports for the database using external tools such as Crystal Reports.

You are not obliged to use this driver. You can access the AssetCenter database directly if your reporting program supports the relevant DBMS directly.

Advantages of using the ODBC driver:

	Using the ODBC driver	Without the ODBC driver
Security when accessing the AssetCenter database	When using reports, an AssetCenter login and password are required to access the database. User profiles (access rights and restrictions) associated with the login are respected.	When using reports, the database access parameters required by the DBMS are requested. These are not linked with the AssetCenter user profiles.
DBMS connection parameters	You don't need to know the connection parameters required by the database in order to access the database.	You need to know the connection parameters required by the database in order to access the database.
Selecting the connection used to access the AssetCenter database.	The user of the report selects the appropriate connection.	The user directly accesses the database without using the AssetCenter connections.
Link between the DBMS engine and the report	The report is independent of the database engine used. When you change DBMS, you don't have to modify your reports.	The report depends on the database engine. When you change DBMS, you have to modify your reports.

### Data items accessible via the ODBC driver

The ODBC driver allows you to view:

- Tables
- Standard fields
- Calculated fields
- Features

All these objects are identified by their SQL name.

Links are not visible. You must reconstitute them by making the joins yourself.

### Which ODBC connection to use

A standard ODBC connection is created when the ODBC driver is installed. It is called **AssetCenter Databases**. This connection cannot be modified or removed.

You can use two types of ODBC to connection to access the AssetCenter database:

- The standard **AssetCenter Databases** connection.
- A connection you have created yourself.

### Usefulness of the standard AssetCenter Databases connection

By using this connection, you avoid having to create your own. In this way, you don't have to use the ODBC administrator. The AssetCenter connection to be used is selected when creating and using the report. The standard connection dialog box under AssetCenter is used for this.

### How to create your own ODBC connections

- ✎ Launch the ODBC administrator.
- ✎ Create a new connection by selecting the "Peregrine AssetCenter Driver".
- ✎ Proceed as usual to create the ODBC connection.

---

## Example: Creating a report under Crystal Reports with the ODBC driver

- ✎ Launch Crystal Reports.
- ✎ Open a new report.
- ✎ Indicate than the report concerns "SQL/ODBC" data.
- ✎ Select the **AssetCenter Databases** ODBC connection.
- ✎ The standard AssetCenter connection dialog box is displayed.
- ✎ Select the appropriate AssetCenter connection. Enter the login to be used to create the report and the appropriate password.
- ✎ Create the report in the usual fashion.

# Chapter 11 - Defining actions

This chapter explains how to define actions with AssetCenter.  
You will find information on the following topics:

## ***Introduction***

- Definition of an action

## ***Defining actions***

- Creating an action
- Examples of actions
- Using variables
- Specificities of the Sybase SQL Anywhere database engine

## ***Operations on actions***

- Testing an action
- Executing an action

***Use the Tools/ Actions/ Edit menu item to create actions.***

***You can execute actions via the Tools/ Actions menu item or the contextual list of "Actions" in the toolbar.***

---

## Definition of an action

An action is an operation that calls on an external program.

There are several types of actions:

- **Executable**,
- **DDE**,
- **Messaging**,
- **Script**: modifying an object in the AssetCenter database,
- **Wizard**,
- **Printing**.

Actions must first be defined before they can be executed by selection from a list.

---

## Creating an action

This section describes how to create an action:

- Types of actions
- General method
- Populating the DDE tab
- Populating the Messaging tab

### Types of actions

AssetCenter allows you to define several types of actions.

#### *Executable* actions

An **Executable** action causes a program to be executed.

It launches an **.exe**, **.com**, **.bat**, or **.pif** application. You can also refer to any type of document, as long as its extension is associated with an application in the file manager.

#### *DDE* actions

A **DDE** action sends a DDE request to a DDE server application (or DDE compliant application) capable of handling DDE requests.



DDE stands for "Dynamic Data Exchange"; it designates a method for dynamically exchanging information between programs. AssetCenter uses DDE have commands executed by another application.

Example: Through DDE, you can request Microsoft Word to open a file whose name is specified and with given contents.

## Messaging actions

AssetCenter allows you to define and launch **Messaging** actions.

**Messaging** actions allow you to send a message:

- Via AssetCenter's internal messaging system.
- Via an external VIM standard messaging system (Lotus Notes, Lotus cc:Mail, etc.).
- Via an external MAPI standard messaging system (Microsoft Exchange, Microsoft Outlook, etc.).
- Via an external SMTP standard messaging system.

Important note: You can only send messages via those messaging systems that you are able to connect to.

To issue a VIM, MAPI or SMTP standard message, AssetCenter uses:

- The **Account** (SQL name: MailLogin) and **Password** (SQL name: MailPassword) fields of the **Messaging** tab of the detail of the employee who opened the AssetCenter database (table of departments and employees) to identify the sender of the message,
- The **EMail** field (SQL name: EMail) in the **General** tab of the employee detail to identify the recipient of the message.

To send a message via AssetCenter's internal messaging system, AssetCenter uses the **Login** and **Password** fields in the **Messaging** tab of the details of both the sender and recipient.

Note: The internal messaging address of an AssetCenter user is the same as the **Login**.

Warning: The administrator must create a user with the name "Admin" and populate the **Account**, **EMail** and **Password** fields in order to use an external messaging system and make sure that AssetCenter Server functions correctly.

## Script actions

**Script** actions allow you to perform any operation on an AssetCenter database. They give advanced users extensive control over the database, allowing them to perform operations that cannot be performed with other types of actions, and in particular:

- Creating records
- Deleting records
- Duplicating records
- Modifying one or more objects in the AssetCenter database, e.g. all the records in a table, a field or a link.

The operations performed by this type of action are described by a BASIC script; this allows you to use complex functions similar to those in the AssetCenter APIs.

The complexity of the usable functions in **Script** action, associated with the ability to make in-depth changes to the database, make this type of action potentially dangerous to the database integrity. Therefore it should be used by advanced users only.

Different functions are used to change the value of a database object depending on the context of the action:

- If the action has no context, you must use functions derived from AssetCenter APIs such as "AmSetFieldStrValue()" or "AmSetFieldLongValue()".
- If the action has a table as a context, you can use the "Set()" function; it has the following syntax:

```
Set [<Link.Link.Field>]=<Value>
```


## "Wizard" actions

Wizards are intended to guide you step by step through complex or recurrent tasks in AssetCenter. Wizards are designed via a dedicated programming language.


Wizards are complex actions. They are documented in detail in the administration and implementation manuals of AssetCenter.

## General method

To create an action:

- Select **Tools/ Actions/ Edit**".
- Click .
- Enter a name for the action.
- In the **Type** field (SQL name: seActionType), specify the type of action you want to create. The type of action you select controls the display of one of the following tabs:
  - ❖ **Executable**.
  - ❖ **DDE**.
  - ❖ **Messaging**.
  - ❖ **Script**.
  - ❖ **Wizard**.
  - ❖ **Printing**.
- Optionally populate the **SQL name** field (SQL name: SQLName) of the action detail. The SQL name is a unique way to identify the action; it is used in particular when executing an action via a DDE command (in cases where AssetCenter is used as a DDE command server).

If you do not populate the **SQL name** field, AssetCenter does so automatically by generating a standard SQL name.

- Populate the **Context** field (SQL name: ContextTable):
  - ❖ If you select a table from the drop-down list, the action is context-sensitive: it will only be proposed if you display the list of records in that table, or the details of one of those records.
  - ❖ If the action does not depend on a table, select the **(No table)** option at the top of the drop-down list.
- Optionally attach an icon to the action:  
To do this, use the square located at the top left of the action detail screen. The image will then appear in the "Actions" context-sensitive list in the toolbar. The active icon in the list (the one displayed on the screen by default) is the icon for the last action executed using the toolbar.
- Populate the fields in the **Description** tab, and the fields in the specific tab for the type of action you want to create.
- Click .

Note: the AssetCenter administrator sees all actions, whether or not they are shared, and whether or not they were created by the administrator.

## Populating the DDE tab

*The information concerning a particular DDE action is located in the DDE tab of the action detail.*

This tab is displayed only if you assigned the value **DDE** to the **Type** field (SQL name: seActionType) field in the basic information for the action.

The DDE mechanisms are based on the "services" provided by the software. In order to execute DDE mechanisms, you must define a "topic" that indicates the context in which the "commands" should be executed.

Therefore you must indicate:

- In the **Service** field (SQL name: DDEService), the name of the DDE service provided by the executable you want to call. This is usually a unique service for an executable. Refer to the documentation of the executable for the list of services it provides.
- In the **Topic** field (SQL name: DDETopic), the context in which the action should be executed.
- In the **Command** field (SQL name: DDECommand), the commands you want the external application to execute.

For Word, the command may be a WordBasic or a Visual Basic command.

If the DDE service of the called application allows it, you can put several commands side by side.

You must follow the syntax required by the external application.

- If the service is not present, indicate in the **File** field (SQL name: ActionFile) the file used to start the application which activates the service. This is the main application that responds to DDE commands.

## Important note

Commands sent to the external application are surrounded by square brackets ("[" , "]" ). For example (using Microsoft Word):

```
[FileOpen( "c:\tmp\test.txt" ) ]
```

- When the action is contextual, you can use variables to reference the value of a field in the database. Since these variables are also surrounded by square brackets, AssetCenter cannot differentiate between commands and variables by itself. You must identify commands by prefixing the square brackets with a backslash character "\". Thus the previous example is written (in the case of a contextual action):

```
\[FileOpen("c:\tmp\test.txt")\]
```

You can combine commands and variables, as shown below (in this case the context is the table of assets):

```
\[FileOpen("c:\tmp\"+[AssetTag]+".txt")\]\[FileClose()\]\[FileExit()\]
```

- If the action is not contextual, the problem does not arise. Text surrounded by square brackets is still considered as commands to send to the external application.

## Populating the *Messaging* tab

*Information concerning a Messaging action is located in the Messaging tab of the action detail.*

This tab is displayed only if you set to the **Type** field in the basic information of the action to **Messaging**.

Warning: In order for the system to function correctly, your system's PATH variable must include the folder containing the VIM DLL (VIM.DLL for AssetCenter 16-bit and VIM32.DLL for AssetCenter 32-bit) and the MAPI DLL (MAPI.DLL for AssetCenter 16-bit or MAPI32.DLL for AssetCenter 32-bit.)

## What is the *Referenced object* field used for?

This field is used to select a link from the table selected in the **Context** field.

This field is only used for messages sent via AssetCenter's internal messaging system. It allows you to directly access the object that triggered the issuing of the message by simply clicking the Referenced object... button in the message detail. When the referenced object is directly the record that triggers the action, you do not fill in the **Referenced object** field (SQL name: RefObject).

## How to receive an acknowledgement

If you want the message sender to receive an acknowledgement via their usual messaging service, check the **Acknowledgment** box (SQL name: bAcknowledgment).

This acknowledgement will be sent to the address specified by the **EEmail** field (SQL name: EMail) in the **General** tab of the employee who opened the AssetCenter database (in the departments and employees table).

Note: You cannot receive an acknowledgement for a message sent via the AssetCenter internal messaging system, or via a MAPI or SMTP messaging system.

## How to indicate an address

Here are the different ways of indicating an address:

### Address with form <Messaging engine>:<Messaging address>

<Messaging engine> can be:

- ❖ AM: to force the utilization of AssetCenter's internal messaging system.
- ❖ MAPI: to force the utilization of a MAPI standard messaging system (Internet Mail, Microsoft Outlook, etc).
- ❖ VIM: to force the utilization of a VIM standard messaging system (Lotus Notes, etc.).
- ❖ SMTP: to force the utilization of a SMTP standard messaging system (Internet standard).

<Messaging address> has the usual form corresponding to the messaging system selected. Internal messaging addresses are the same as the "Logins".

Examples of addresses:

- ❖ AM:Admin
- ❖ MAPI:CathyBernard@taltek.com
- ❖ VIM:Cathy Bernard / TALTEK

### Address with form <AssetCenter login>

In this case, the messaging system used will be that which is indicated in the **EEmail** (SQL name: EMail) of the **General** tab of the detail of the

employee whose **Login (Profile** tab in the detail of the employee) is specified in the address.

If the **EMail** field is not populated, the message is sent via the internal messaging system.

For example:

- ❖ A message is sent to the following AssetCenter logins: "Cathy", "Gerald" and "Philip".
- ❖ The **EMail** fields show "MAPI:CathyBernard@taltek.com" for "Cathy" and "VIM:Gerald Colombo / Taltek" for "Gerald". The **EMail** field of "Philip" is empty.
- ❖ If the sender has a MAPI account, the message will be sent to "Cathy" via MAPI and to the two other recipients via AssetCenter's internal messaging system.
- ❖ If the sender has a VIM account, the message will be sent to "Gerald" via VIM and to the two other recipients via AssetCenter's internal messaging system.

### Address with contextual variables

If the action is contextual, you can use variables between brackets []. These variables call on values of fields in the AssetCenter database.

For example: to send a message to the user of an asset selected in the table of assets, you can use **[User.Email]**.

---

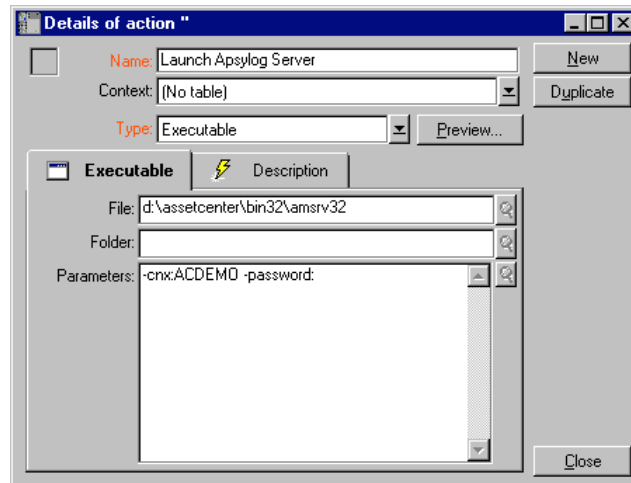
## Examples of actions

This section provides examples of AssetCenter actions:

- Example of Executable type action
- Example of DDE type action
- Example of Messaging type action
- Example of a Script type action

## Example of "Executable" type action

The following screen describes a non-contextual action which launches AssetCenter Server and connects to a database "acdemo":



*Detail of an executable type action*

## Example of "DDE" type action

There are numerous applications of **DDE** type actions:

- Inserting AssetCenter data into a Microsoft Excel worksheet.
- Inserting information related to a purchase order in an accounting software package.
- Automatic sending of a work order request.
- Etc.

This section describes a simple **DDE** action.

### Aim of the action

This action sends confirmation of a purchase request.

This action is triggered from the detail of a purchase request.

The action uses a DDE link between Microsoft Word 7 and AssetCenter. It inserts information on the request in a Word document (details of the person to contract and the request number) and prints it.



## Preparations: Creating the letter under Word

You must first create the Word document "LetterTemplate.doc" which will be printed.

This letter will be structured as follows:

I

TALTEK SERVICES  
San Mateo Site  
IT Department

<MrMrs> <FirstName> <Name>  
<Adr1>  
<Adr2>  
<Zip> <City>

San Mateo. { DATE }

Dear <MrMrs>,

Following our conversation, I am pleased to confirm that your request has been attributed the following request number: <ReqNo>.

Regards,

Procurement Manager

*"LetterTemplate.Doc"*

The "LetterTemplate.doc" document can be found in the installation folder of AssetCenter.

The corresponding Word template "Normal.dot" is attached to "LetterTemplate.doc". It contains a macro, "mymacro.bas":

```
Attribute VB_Name = "MyMacro"
Sub PrintLetterTemplate(MrMrs, FirstName, Name, Adr1, Adr2, Zip, City,
ReqNo)
'
' PrintLetterType Macro
'
Application.WindowState = wdWindowStateMinimize 'Run Winword in the
back end
Documents.Open ("LetterTempl.doc") 'Open letter pattern
Documents("LetterTempl.doc").Activate

Selection.Find.ClearFormatting 'Clear parameters for Find function
Selection.Find.Replacement.ClearFormatting 'Clear parameters for
```

## Replace function

```
With Selection.Find
.Text = "<MrMrs>"
.Replacement.Text = MrMrs
.Forward = True
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll 'Execute replacement

With Selection.Find
.Text = "<FirstName>"
.Replacement.Text = FirstName
End With
Selection.Find.Execute Replace:=wdReplaceAll

With Selection.Find
.Text = "<Name>"
.Replacement.Text = Name
End With
Selection.Find.Execute Replace:=wdReplaceAll

With Selection.Find
.Text = "<Adr1>"
.Replacement.Text = Adr1
End With
Selection.Find.Execute Replace:=wdReplaceAll

With Selection.Find
.Text = "<Adr2>"
.Replacement.Text = Adr2
End With
Selection.Find.Execute Replace:=wdReplaceAll

With Selection.Find
.Text = "<Zip>"
.Replacement.Text = Zip
End With
Selection.Find.Execute Replace:=wdReplaceAll

With Selection.Find
.Text = "<city>"
.Replacement.Text = City
End With
Selection.Find.Execute Replace:=wdReplaceAll

With Selection.Find
.Text = "<ReqNo>"
.Replacement.Text = ReqNo
End With
Selection.Find.Execute Replace:=wdReplaceAll

ActiveDocument.PrintOut 'Print document
```


```

ActiveWindow.Close (wdDoNotSaveChanges) 'Close document w/o
updating
MsgBox ("Your document is being printed.") 'Notify user
End Sub

```

## Step 1: Creating the AssetCenter action

To create the AssetCenter action:

- ↳ Open the table of actions (**Tools/ Actions/ Edit** menu item).
- ↳ Click  to create a new action.
- ↳ Populate the **Context** field (SQL name: ContextTable) in order for it to show the table of purchase requests.
- ↳ Set the **Type** field to **DDE**.
- ↳ Populate the **DDE** tab in order to define the DDE link between AssetCenter and Microsoft Word 7.


In the DDE tab:

- ↳ The **Service** field (SQL name: DDEService) is set to "Winword".
- ↳ The **Topic** field (SQL name: DDETopic) is set to "System".
- ↳ The "Service start parameters" frame shows "Winword.exe" and its path.
- ↳ The **Command** field (SQL name: DDECommand) specifies the macro to be launched and its parameters:

```

\[MyMacro.PrintLetterTemplate "[Requester.MrMrs]",
"[Requester.FirstName]", "[Requester.Name]",
"[Requester.Location.Address1]", "[Requester.Location.Address2]",
"[Requester.Location.ZIP]", "[Requester.Location.City]",
"[ReqNumber]" \]

```

Click  to confirm the creation of the action.

## Step 2: Launching the action

To launch the action:

- ↳ Open the table of purchase requests.
- ↳ Select a purchase request.
- ↳ Launch the action via the **Tools/ Actions** menu item.

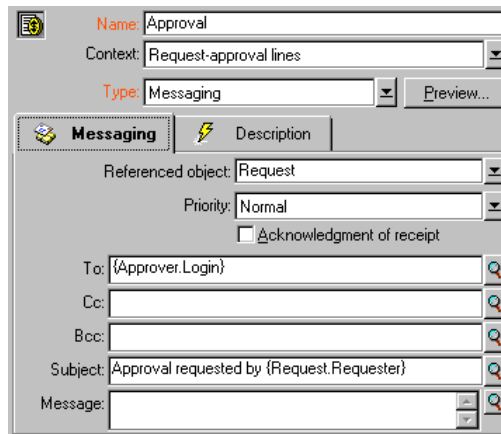
When the action is launched:

- ↳ Microsoft Word is launched and loads "LetterTemplate.doc".
- ↳ The details of the person to contact and the request number are inserted in the letter.

⇒ The letter is printed.

## Example of Messaging type action

You send a message from the detail of a purchase request approval line to indicate that it is to be validated. In order for the referenced object to be the purchase request and not the request line, configure the action detail as follows:



*Detail of a messaging type action with a referenced object*

## Example of a Script type action



Creating a **Script** type action basically involves writing the BASIC script which modifies the AssetCenter database.

The use of functions specific to these actions is authorized within these scripts. These functions are indexed in the manual entitled "Programmer's Reference", chapter "Index of functions by field of application", section "Built-in functions"

## Foreword

To prepare for creating the action, follow these steps:

- Select the **Tools/ Actions/ Edit** menu item and click the  button in the action detail screen.

- Assign a name to the action you are going to create, e.g. "Test", and set the **Type** field (SQL name: seActionType) to **Script**. Do not select a context for the action. Click .
- In the **Script** tab, click the  button to display the script builder window. The programmable function, called "Success()", used for these actions does not require an explicit return code. In the following example, we will create a new record in the table of categories based on the information contained in the table below:

Field label	SQL name of the field	Value of the field
<b>Category</b> (SQL name: Name)	Name	PC
<b>Nature</b>	seNature	Computer
<b>'Connection' tab visible by default</b>	bIsCnxClient	This box is checked

## Writing the script

Enter the following example:



```
Dim lrec As Long
Dim lres As Long
lrec=AmCreateRecord("amCategory")
lres=AmSetFieldStrValue(lrec, "Name", "Micro")
lres=AmSetFieldStrValue(lrec, "seNature", 1)
lres=AmSetFieldStrValue(lrec, "bIsCnxClient", 1)
AmInsertRecord(lrec)
```

This action creates the desired category without any user intervention.

## Demonstration of the "Set()" function

Now we will create the same category from a **Script** type action, by specifying the table of categories as the context for the action. In this case, we write the script as follows:

```
Set [Name]="Value"
Set [seNature]=1
Set [bIsCnxClient]=1
```

To execute this action, the user must open the table of categories and press . After executing the script, the user must also click  to validate the creation.

## Tip

If you want to invalidate the execution of an action within a script, simply set the value of the return code to something other than 0 (12001, for example). This value is considered as an error code. The next command interrupts the action and cancels all changes already made:


```
RetVal=12001
```

---

## Using variables

In the **Executable**, **DDE**, or **Messaging** tabs of the detail of a contextual action, you can use variables which reference the contents of fields, features or calculated fields in the database.

They use the form **[Link.Link.Field]**.

For help in entering these variables, click the magnifier  to the right of the field to be populated.

Everything not contained within braces **[]** is considered as text.

For example: **[Link.Link.Field].doc**, calls the value of the field **Field** in the table linked to the main table going through the links **Link.Link**.

Warning: In order for the variables to work, the **Context** field of the action detail must show a table in AssetCenter and you must select a record in the list of records of the table before executing the action.

---


## Specificities of the Sybase SQL Anywhere database engine

When using Sybase SQL Anywhere as AssetCenter's database engine, it is not possible to write "{d" or "m" at the start of a field in an action.


If you need a field in an action to start with "{d" or "m", we recommend preceding these strings by a space.

---

## Testing an action

To test an action when creating it, use the "Preview" button  found in the top right corner of the detail of the action to be tested.

 button

Once the context is selected, click the  button. This fills in the fields in the **Executable**, **DDE**, or **Messaging** tabs. Check that the variables have been correctly extracted from the record selected in the **Context** field (SQL name: ContextTable).






 button


This button allows you to execute the action directly from this screen.

---

## Executing an action

You may execute an action in one of several ways:

- Using the popup list  in the toolbar:
  - ❖ The  button is replaced by the icon associated with the last action used on this workstation, if such an icon exists. If an action has already been executed, click the  icon, or the icon replacing it to repeat the action.
  - ❖ The  button displays the list of available actions.
  - ❖ To insert this popup list in the toolbar, use the **Tools/ Customize toolbar** menu item: it is part of the "Tools" category.
- Using the **Tools/ Actions** menu item: Click the desired action in the sub-menu.
- From the  button in the top right corner of the detail of the action:
  - ❖ If the action is contextual, specify the **Context** (SQL name: ContextTable) by selecting a record in the action's reference table.

- ❖ Click  to execute the action.
- From the popup menu (accessible by clicking with the right button of the mouse). If at least one action is available for the open table, the **Actions** entry is shown in the popup menu.

## Multiple-selection in lists




You may select several records in a list and apply an action to them.


In this way, you can select several assets and send the same message to their users.

## "Wizard" type actions

Wizards are composed of a succession of pages. Each of these pages displays information or requires user input, such as a selection to be made or data items to be entered.

Navigating between the different pages of a wizard is simple:

- Once the page is populated appropriately, you can move to the following page (determined by a transition) by clicking the  button. This button is not available for the final page.
- You can always go back to make any corrections by clicking the  button.
- You can execute the final action of the wizard at any given moment by clicking the  button. If the wizard does not have sufficient information in order to perform its designated task, the appropriate page is displayed.

You can cancel the execution of a wizard completely (and as a consequence, its associated action) by clicking the  button.



# Chapter 12 - Managing deadlines with AssetCenter Server

This chapter explains how to manage deadlines (alarms, purchase request approvals, stock reordering, etc.) and automatic triggering of actions (automatic issuing of reminder messages, etc.).

***The administrator manages the monitoring of deadlines and the automatic triggering of actions using the AssetCenter Server program, which is independent of AssetCenter.***

In this chapter you will find information concerning how to configure and using AssetCenter Server:

- Overview of AssetCenter Server
- Executing AssetCenter Server
- Main screen of AssetCenter Server
- General options of AssetCenter Server
- Modules monitored
- Activating the verification immediately
- Administrating AssetCenter Server by the Web

---

## Overview of AssetCenter Server

AssetCenter includes a system for monitoring deadlines and automatically triggering actions: This program, called AssetCenter Server, functions independently of AssetCenter.

AssetCenter Server automatically monitors all expiration dates in the designated database:

- Alarms (end of term dates of contracts for example).

- Purchase request approvals.
- Stock line reorder levels.
- Rent calculations at the asset and the contract level.
- Lease contract loss value calculations.
- Expense line split operations associated with cost centers.
- Verification of history lines.
- Workflow deadlines.
- Search for new workflow execution groups.
- Execution of workflow rules.
- Verification of time zones.

If justified to do so by the deadlines, AssetCenter Server performs actions, such as issuing reminder messages in the AssetCenter database via the internal messaging system. If necessary, it calculates contract rent, lease contract loss-values, etc.

Important note: When you exit AssetCenter Server, all automatic monitoring functions are suspended.

It is possible to launch AssetCenter Server on several different machines. Therefore deadlines to monitor and tasks to perform can be shared by different instances of AssetCenter Server. This allows you to improve the performance of AssetCenter Server.

You must make sure that a given task is only performed by one single instance of AssetCenter Server.

You can use the same Login to connect to the database. This login must have administration rights.

## Executing AssetCenter Server

### Recommendations

AssetCenter Server needs to frequently access the database. This is more than likely to be held on a network.

- If there is a workstation with a high-speed connection to the database, execute AssetCenter Server from this workstation. Monitoring will be carried out for all users.
- If you can only access the database over a low-speed link and if your server operates using Windows, you have the possibility of running AssetCenter Server directly on the server.

### In case of modification of the structure of the database

If you modify the structure of the database via AssetCenter Database Administrator or via the "Configure object" command in the popup menu, you must disconnect AssetCenter Server from the database and then reconnect.

## Executing AssetCenter Server manually under Windows

### Launching AssetCenter Server

There are different possible methods of launching AssetCenter Server:

- ❖ By launching the AssetCenter Server program from the programs in the **Start** menu or in the AssetCenter program group.
- ❖ By launching "aamsrv32.exe" (for the 32-bit version) which can be found in the **Bin32** sub-folder of the AssetCenter installation folder.
- ❖ You can either launch the "AssetCenter Server" program automatically or as a service under Windows NT.

We recommend launching AssetCenter Server from the "Services" icon in the Windows NT control panel.


AssetCenter Server asks you which database to connect.

### Connecting AssetCenter Server to a database

Only an administrator can connect to the database via AssetCenter Server. This can be the "Admin" user or a user with administration rights.


You need to enter the relevant **Login** and password. Select the **Use this connection in service mode** check box if you want to use this connection as the default connection if **AssetCenter Server** is running in service mode (as in a service in Windows NT).

To connect AssetCenter Server to the database, use one of the following:

- ❖ the **File/ Connect** menu item,
- ❖ the  icon.

### Disconnecting AssetCenter Server from a database

To disconnect AssetCenter Server from the database, use one of the following:

- ❖ the **File/ Disconnect** menu item,
- ❖ the  icon.

### Launching AssetCenter Server from a DOS box (Windows 32-bit)

You can automate the launching of AssetCenter Server by using the following command:

```
aamsrv32 -cnx:<connection name> -login:<login> > -password:<login  
password>
```

Important note: The "login" is that of an AssetCenter administrator (either "Admin", or a user login with administration rights).

Strings between <> should have no spaces.

Example: **aamsrv32 -cnx:PeregrineDatabase -login:Gerald -password:Password**

This command can be placed in a batch file.

Important note: This command can only be executed in a DOS box when using a 32-bit version of Windows.

### If you exit AssetCenter Server

When you disconnect AssetCenter Server from the database, all monitoring functions and automatic triggering of action are suspended. When you connect to the database under AssetCenter, AssetCenter displays a warning message indicating that AssetCenter Server has not accessed the database in the last hour.

Nevertheless, if AssetCenter Server encounters a connection error after a connection has been opened it will try to reconnect to the database according to the frequency defined in the general monitoring options.

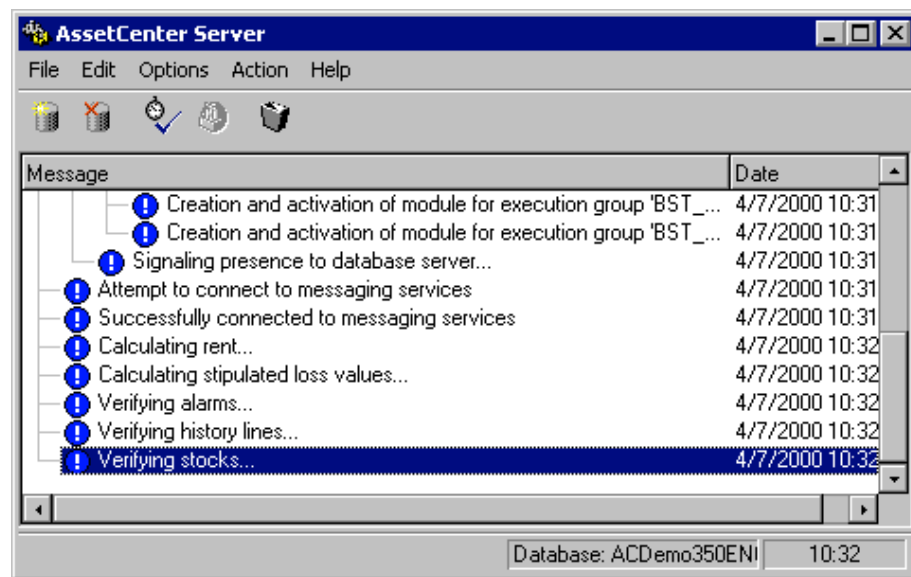
These attempts to reconnect to the database will only be made if the original connection was made without error.

### AssetCenter Server and the messaging system

The computers on which AssetCenter Server is installed must have a working messaging system installed. Each login user must be configured correctly in the database in order to send messages via the specified messaging system.






---

## Main screen of AssetCenter Server



*Main screen of AssetCenter Server*

The main window displays all events handled by AssetCenter Server. It allows you to access the following program menus and icons:

Icon	Menu	Function
	<b>File/ Connect</b>	Connect to a database.
	<b>File/ Disconnect</b>	Disconnect from a database.
	<b>Action/ Activate</b>	Selectively activate deadline monitoring agents.
	<b>Action/ Empty list</b>	Clear messages from main screen.
	<b>Options/ Configure</b>	Configure deadline monitoring options.
	<b>Action/ Connect to messaging</b>	Try to connect to external messaging systems. This button is grayed out if the connection is successful. If the connection fails, this button is available. Click to test the connection.

Whenever the list of events becomes too large, you can use the **Action/ Empty list** command to empty it.

## General options of AssetCenter Server

The general monitoring options appear in the **General** tab of the screen displayed by the **Options/ Configure** menu item.

These govern the general operation of AssetCenter Server.

### Database reconnection delay

This field is used if AssetCenter Server receives an error message when trying to connect to the database, after the initial connection has been made successfully.

In this case, AssetCenter Server considers that the connection to the database has been lost and stops monitoring deadlines. If at least one previous connection has been successful, AssetCenter Server tries to reconnect to the database according to the frequency defined in the **Database reconnection delay** field.

Monitoring function are resumed when AssetCenter Server manages to reconnect to the database.

The data entry format for this field is "Duration".

### **Messaging reconnection delay**

If a problem occurs with the external messaging system, AssetCenter Server will stop sending external messages

AssetCenter Server tries to reconnect to the external messaging system according to the frequency defined in this field.

The data entry format for this field is "Duration".

### **Log file**

#### **File**

This file stores the messages displayed in the main window of AssetCenter Server.

#### **Max. size**

This field allows you to limit the size of the file which logs the messages shown in the main screen of AssetCenter Server.

When this limit is reached, the oldest messages are deleted as and when new messages are recorded.

### **Time zones**

In the **General** tab of the configuration screen, you configure the types of time-zone test to be performed:

- **Verify time zone of database server.**
- **Verify local time compared to that of the server.**

These two tests compare the time of the database server with that of the machine on which AssetCenter Server is installed. The time difference is expressed as  $[(n * 30\text{minutes}) + m]$  where  $m$  is between -15 minutes and + 15 minutes.

### In both cases

If the time difference exceeds 5 minutes, AssetCenter Server offers to update the local time on the machine on which it is installed.

If you refuse this update (for example, if you think that it is the time of the server that requires changing), the connection is refused. You can connect again once the difference between the two times no longer exceeds 5 minutes (resulting from either a modification to the time of the database server and/or of the machine on which AssetCenter Server, is installed).

### Specific aspects of the Verify time zone of database server option

If necessary, the information on the time zone of the server in the table of options of AssetCenter is updated (if the number ( $n * 30$  minutes) does not correspond to the time zone of the server).

To do this, the machine on which AssetCenter Server is running must have the correct time and have the correct information on daylight saving changes.

### Specific aspects of the Verify local time compared to that of the server option

The time zone of the server, necessary for internal operations of AssetCenter, is recovered.

Note: Whichever option you select, the tests are performed when AssetCenter Server connects to the database, then regularly according to the frequency defined in the **Modules** tab of the configuration screen of AssetCenter Server.

---

## Modules monitored by AssetCenter Server

*Use the Modules tab in the configuration screen of AssetCenter Server to configure the module to be monitored.*

This section describes what is monitored and details the monitoring parameters:

- Selecting the modules to be monitored



- Verification schedules
- History monitoring
- Stock monitoring
- Alarms
- Rent
- Loss values
- Split operations on expense lines
- Time zones
- Presence signal
- Detection of new workflow execution groups by AssetCenter Server
- Executing workflow rules

## Selecting the modules to be monitored

The **Modules** tab in the configuration window of AssetCenter Server allows you to define:

- Those modules AssetCenter Server monitors: Check the appropriate boxes in the list of modules.
- The frequency of monitoring, specific to each module: Populate the "Verification schedules" zone.

Thus, if you want to launch AssetCenter Server on several machines, you must share the modules to be monitored between the different instances of AssetCenter Server.

Note: The **User data item** field allows you to specify additional information. It is only used for the "Rents" module.

## Verification schedules

To define the verification schedules of a module:

- Select the module from the list shown in the **Modules** tab
- Populate the sub-tab of the "Verification schedules" zone.
- If necessary, create and populate additional sub-tabs by right-clicking the labels of the sub-tabs and selecting the **Add a rule** menu item.

The sub-tabs are used to allow you to define the days and times of monitoring.

## Days of monitoring

Value of the "Days" field	Monitoring is performed
"Daily"	Every day of the year, without exception.
"Day of the week:"	The weekday selected from an itemized list in the field to the right of the "Days" field. Example: Every Monday
"Day of the year:"	A day or selection of days. These are defined via the <b>Day</b> , <b>Month</b> and <b>Year</b> check boxes. Example: July 20
"The first" "The second" "The next to last" "The last "	The days of the week defined using the <b>Day</b> check box, and for the month(s) and year(s) defined using the <b>Month</b> and <b>Year</b> check boxes. Examples: "The first" Friday of each month. "The second" Monday of the month of September. "The next to last" Wednesday of the month of November. "The last" Tuesday of every month throughout 1998.

## Monitoring times

### Periodical

You can define two frequencies for the verification of a module which depend on the time of day.

The first "During period" frequency applies to scheduled periods that you can create in two ways:

- ❖ Graphically, using the schedule bar. Click and drag on this bar to create a schedule period.
- ❖ Directly, by entering an itemized list of values in the field to right of the schedule bar. The following syntax is used:  
**<Start time of period - End time of period>**  
The time format defined in AssetCenter must be respected.  
To indicate several periods, you just need to separate them with semi-colons ";".

The second "Outside of period" frequency is applied outside of those scheduled periods you have defined.

Example:



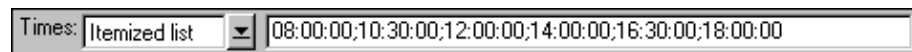
The screenshot shows a scheduling interface. On the left, a dropdown menu is set to 'Periodical'. To its right is a grid of numbers: 3, 6, 9, 12, 15, 18, 21. The number 9 is highlighted in blue. To the right of the grid is a time range field containing '8:00-19:00'. Below the grid, there are two input fields: 'During period:' followed by '15min' and 'Outside of period:' followed by '30min'. Both input fields have small up/down arrows on their right sides.

### Itemized list

You can enter those times you want the verification to be carried out on in the field to the right.

- ❖ The time format defined in AssetCenter must be respected.
- ❖ The semi-colon ";" is used to separate them.

Example:



The screenshot shows a scheduling interface. On the left, a dropdown menu is set to 'Itemized list'. To its right is a single text field containing the string '08:00:00;10:30:00;12:00:00;14:00:00;16:30:00;18:00:00'.

## Preview

You can have a preview of:

- A rule defining the monitoring schedules of a module in the **Preview** field of a rule description sub-tab.
- All the rules concerning a module in the **Preview** sub-tab of the **Modules** tab, once the module is selected.

## History lines monitored by AssetCenter Server

Sometimes when a record is destroyed in the database, the corresponding history lines are not destroyed. AssetCenter Server verifies if there are any such history lines; if it finds any it destroys them.

## Stock lines monitored by AssetCenter Server

AssetCenter Server monitors stock reorder levels.

For each stock, AssetCenter Server refers to the stock rules defined in the **Manage** tab of the stock detail.

For each stock rule concerning a product:

- AssetCenter Server calculates the quantity of items actually available from the **Assets** tab or the **Consumables** tab in the stock detail.
- When the quantity falls below the value specified in the **Reorder level** (SQL name: lReordLevel) field of the stock rule detail, AssetCenter Server automatically creates a purchase request.
  - ❖ The parameters of the purchase request can be found in the **Auto-request** tab of the detail of the stock.
  - ❖ The purchase request specifies the quantity to be reordered (**To order** field (SQL name: lQtyToOrder) in the detail of the stock rule).
- For as long as the request is not fully received, AssetCenter Server does not verify the stock rule that it has generated. Therefore, no new request is sent.
- As soon as delivery of the request is taken in full, AssetCenter Server:
  - ❖ Readjusts the stock levels.
  - ❖ Erases the contents of the **Request line** field (SQL name: ReqLine) in the stock rule detail.
  - ❖ Reactivates the stock rule.

## Alarms monitored by AssetCenter Server

### List of alarms monitored

#### At the asset level


Several key dates are monitored:

- ❖ The end date of reservation of an asset: This is shown in the **Reserv. end date** field (SQL name: dReservEnd) in the **General** tab of the asset detail.
- ❖ The warranty expiration date of an asset: Asset detail, **Maint.** tab, **Expiration** field (SQL name: dWarrEnd).
- ❖ End of term date for lease, rental, loan of an asset: This alarm can only be defined if the acquisition method of the asset (Asset detail, **Acquis.** tab, **Acq. method** field (SQL name: seAcquMethod)) is set to **Lease**, **Rental** or **Loan**. In this case the **Price and conditions** sub-tab of the **Acquis.** tab shows an **End date** field (SQL name: dEndAcqu).

- ❖ Rent end dates of an asset: Alarms can be attached to end of validity dates (**Acquis.** tab, rent descriptions sub-tabs, "Schedule" frame).

### At the consumable level

AssetCenter Server monitors the end date of reservation for consumables: This is shown in the **Reserv. end date** field (SQL name: dReservEnd) in the reservation detail of a consumable. To access the reservation detail of a consumable:

- ❖ Launch AssetCenter.
- ❖ Select **Finance/ Requests**.
- ❖ Select the purchase request reserving the consumable.
- ❖ Display the composition of this purchase request.
- ❖ Display the request line corresponding to the consumable.
- ❖ Display the **Reservations** tab of the reservation line. This tab shows the list of reservations for consumables. Click  to view the detail of a reservation of a consumable.

### At the project level

AssetCenter Server monitors the end dates of project: Project detail, **General** tab, **End** field (SQL name: dEnd).

### At the contract level

Several key dates are monitored:

- ❖ The end of term date: Contract detail, **General** tab, **End** field (SQL name: dEnd).
- ❖ If the contract **Type** (SQL name: seType) is **Lease schedule** or **Master lease**: Alarms can be attached to the notification dates for possible end of lease. These dates are shown to the right of the notification field in the sub-tabs describing the possible end of term options: **Renewal**, **Purchase**, **Return**.
- ❖ If the contract **Type** (SQL name: seType) is **Lease schedule**: Alarms can be attached to the end dates of validity for rent items as shown in the individual rent-description sub-tabs.

### At the purchase request, estimate, or purchase order level

If the acquisition method of the purchase request (Purchase request detail, **Financing** tab, **Acq. method** field (SQL name: seAcquMethod)) is set to **Lease**, **Rental** or **Loan**, it is possible to define an alarm

associated with the rental, lease or loan end dates (**To** field in **Financing** tab of purchase request detail).

The same is true for estimates and orders.

### What happens in two-level alarms when the first level action has been triggered?

In the case of 2 level alarms, the triggering of the second level depends on the first-level action.

- If the first level alarm triggers an action other than the sending of a message via AssetCenter's internal messaging system (such as sending a message via a third-party messaging system), the second level alarm will always be triggered at the defined moment.
- If the first level alarm sends a message to a group of AssetCenter users via the internal messaging system, the action defined at the second level will not be triggered if one or more of the recipients has read the message.

### Rent calculated by AssetCenter Server

AssetCenter Server monitors periodic rent payments for contracts and assets. It calculates and/recalculates the amounts involved.

The "Calculate rent" module in the **Options/ Configure** menu item of AssetCenter Server defines:

- Certain parameters concerning the generation of costs for contracts and asset level rent payments.
- The frequency of updates.

### Overview

AssetCenter Server verifies at regular intervals whether it needs to generate expense lines. If this is so, it generates them.

After checking and generating the expense lines relative to a periodic rent, AssetCenter Server stores the date of the last expense line (past or present) in the **Recalculation effective from** field (SQL name: dRecalcul).

- If the contract level rent is distributed to the assets, AssetCenter Server modifies the **Recalculation effective from** field that is found in the rent sub-tabs of the **Acquis.** tab of the assets detail.

- If the contract level rent is not distributed to asset level, AssetCenter Server modifies the **Recalculation effective from** field, that is found in the rent sub-tabs of the **Rents** tab of the contract detail.

AssetCenter Server does not recalculate every single expense line each time.

- Projected expense lines associated with a periodic rent are always recalculated.
- The **Recalculation effective from** field, proper to each rent, sets the date from which past and present expense lines associated with a periodic rent are recalculated.

The lessee may directly modify the recalculation date of the non-projected expense lines by directly modifying the **Recalculation effective from** field. This flexibility allows you to be able to recalculate erroneous expense lines in case of a change in tax rates, for example.

## Parameters

The **User data item** field in the **Modules** tab in AssetCenter Server is used to set the rent generation parameters. The syntax of this field is as follows:

```
<Duration>d
```

This duration set the number of days for which the calculation is made. For example, if you want to calculate rent over a period of 90 days, enter the following value:

```
90d
```

The maximum number of rent calculations made per transaction is specified by the **MaxRentPerTrans** entry in the **Amsrv.ini** configuration file.

## Projected rent

The "User data item" field in the **Modules** tab in **AssetCenter Server** allows you to specify the number of days for which you calculate project rent payments.

AssetCenter Server generates the projected expense lines for the specified period. In order to not generate any, you just need to set this field to '0'.

## Example

Let's consider the following configuration:

- The contract is effective from July 1, 1996 through July 1, 1999.
- The rent is payable monthly on the first day of the month.
- AssetCenter Server verifies rent payments every 2 months and generates projected rent payments for the next 12 months.

On July 1, 1997, AssetCenter Server is launched for the first time: it generates:

- Past rents from July 1, 1996 through July 1, 1997.
- The present rent on July 1, 1997.
- The projected rents from August 1, 1997 through July 1, 1998.

Following these calculations, the **Recalculation effective from** field indicates the date of the last non-projected expense line, i.e. July 1, 1997.

AssetCenter Server runs in the background: 2 months later on September 1, 1997, it generates:

- The projected rents from October 1, 1997 through September 1, 1998.
- Past or present rents for which the payment date is later than that contained in the **Recalculation effective from** field, i.e. the rents from August 1, 1997 through October 1, 1997.

## Loss values updated by AssetCenter Server

AssetCenter Server recalculates, at regular intervals, the loss values for lease contracts whose calculation method is set to **Calculate for all periods** (**Calculation** field (SQL name: seLossValCalcMode) in **Leasing** tab of lease contract detail). In this way, loss values pertaining to any loss value rules which have been modified since the last time AssetCenter Server accessed the database are updated.

## Split operations performed for expense lines by AssetCenter Server

AssetCenter Server handles split operations for expense lines.

### Overview

AssetCenter Server searches the expense lines to be split: These are the expense lines whose **Split operation status** field (SQL name: seSplitStatus) is set to **Not split**.



By default, all expense lines are to be split, regardless of their status ( **Status** field (SQL name: seStatus) of an expense line).

AssetCenter Server splits the designated expense lines. When an expense line is split:

- A debit expense line, equivalent to the split expense line is created in the parent cost center.
- Expense lines are created in the target cost centers, according to the split percentage values. By default, these are **Not split**.

### Specific example: Managing the removal of a cost center

When you decide to delete a cost center, and the cost center contains expense lines, AssetCenter will not allow you to perform the operation unless the **Authorize extended deletions** option in the **General** tab in the **Tools/ Options** menu item is checked.

In this case, AssetCenter gives you three possibilities:

- Delete all the linked records.
- Detach the linked records.
- Attach the linked records to another record.

What happens next depends on the option you choose:

#### Delete all linked records

When a cost center is deleted, AssetCenter deletes:

- ↳ The expense lines of the deleted cost center.
- ↳ The expense lines resulting from split operations on the deleted cost center.

An AssetCenter agent modifies the **Split operation status** field (SQL name: seSplitStatus) in order for it to show "Not split" at the level of the expense lines highest up in the split operation and which when split generated the expense lines belonging to the deleted cost center (after any intermediate split operations).

When AssetCenter Server finds these expense lines, which are not split but have generated split expense lines, it deletes all the expense lines resulting from their split operations. In doing this, AssetCenter Server deletes the expense lines which, when split generated the expense lines belonging to the deleted cost center.

Then AssetCenter Server performs the split operations on those expense lines, which have not yet been split, and thus recalculates, using new parameters, all the expense lines which when split generated the expense lines of the deleted cost center.

### **Detach all linked records**

In this case:

- ✎ The expense lines of the deleted cost center are no longer associated with a cost center.
- ✎ The expense lines, which when split generated the expense lines for the deleted cost center are split again.
- ✎ The expense lines, resulting from split operations on the deleted cost center, are not modified.

### **Attach linked records to another record**

In this case, you select another cost center X, which takes the place of the deleted cost center:

- ✎ The expense lines of the deleted cost center are attached to cost center X.
- ✎ The expense lines, which when split generated the expense lines for the deleted cost center, are split again; cost center X is considered as the new target cost center.
- ✎ The expense lines resulting from split operations on the deleted cost center are deleted and the expense lines of cost center X are split.

### **Verifying database server time zone**

The **General** tab in the configuration window of AssetCenter Server allows you to set the type of verification performed by AssetCenter Server.

### **Signaling presence to database server**

AssetCenter Server regularly sends a signal to the database server in order to indicate that it is functioning.

If the database server does not receive a signal from AssetCenter Server for over one hour, a message is displayed when a user connects to the database under AssetCenter.

This message indicates that AssetCenter Server has not been launched on this database for over one hour and that without this process, monitoring functions will be interrupted.

If the database server goes without receiving a signal from AssetCenter Server for over a week, it is no longer possible to connect to the database.

## Searching for new workflow execution groups

AssetCenter Server monitors the creation of new workflow execution groups.

As soon as AssetCenter Server detects a new workflow execution group G, it creates a new monitoring module "Execution of workflow rules for execution group G".

This mechanism has the following advantages:

- It allows you to define verification timetables for each workflow execution group.
- Different workflow execution groups can be monitored by different instances of AssetCenter Server.

## Executing workflow rules

Once a workflow execution group is detected AssetCenter Server executes the appropriate workflow rules.

## Monitoring of workflow execution groups

AssetCenter Server monitors the deadlines specific to workflow instances associated with the execution group.

Deadlines to be monitored by AssetCenter Server as soon as the activity is triggered are defined in the **Alarms** tab of the detail of the workflow activity.

These deadlines are defined by the time limits defined for the set tasks to be carried out.

Note: In the case of deadlines specific to Workflow, business calendars specified in the **Time limit** tab in the activity detail are taken into

account. When calculating deadlines, these time limits are converted to business hours.

---

## Processing of *Periodical* type events

According to the frequency defined in the **Parameters** tab in the detail of a **Periodical** type event, AssetCenter Server triggers the event if the activation conditions are met.

Then, the role of AssetCenter Server depends on the mode of processing of the event as indicated in the **General** tab of the event detail:

- **Log event and process by server:** As soon as the event occurs, AssetCenter Server saves it to the table with SQL name "amWfOccurEvent".  
Then, AssetCenter Server activates the transition according to the frequency of verification as defined in the configuration screen of AssetCenter Server.
- **Log event and process immediately:** As soon as the event occurs, AssetCenter Server saves it to the table with SQL name "amWfOccurEvent", and activates the transition.
- **Process event immediately without logging:** As soon as the event occurs, AssetCenter Server activates the transition.

## Activation of transitions

AssetCenter Server activates the transitions for events according to the frequency defined in the configuration screen. The following events are concerned:

- **System** events.
- **Database** and **Periodical** type events whose processing mode is set to **Log event and process by server**.

## Execution of tasks

AssetCenter Server executes tasks resulting from **Automatic action** or **Test / script** type activities, except in the possible case of tasks resulting from activities whose **Execute actions immediately** (SQL name: bExecImmediately) box is selected.

The frequency with which AssetCenter Server verifies and performs the tasks it has to carry out is indicated in the configuration screen of AssetCenter Server.

In the case of a task originating from an **Automatic action** or **Test / script** type activity whose **Execute actions immediately** box (SQL name: bExecImmediately) is checked:

- This task is executed by AssetCenter Server if it is AssetCenter Server that activates the transition creating the task. In this case, AssetCenter Server performs the task as soon as the transition it creates is activated.
- Else, the AssetCenter client machine executes the task.

---

## Activating the verification immediately

It is possible to activate a verification immediately, without having to wait for the defined period to elapse. You can do this using the **Action/Activate** menu item under AssetCenter Server.

Indicate which verifications to carry out by checking the appropriate boxes.

---

## Administering AssetCenter Server by the Web

The installation program of AssetCenter under Windows NT installs AssetCenter Server as an NT service.

Thus, you have the choice between launching:

The graphical interface of AssetCenter Server.

Or AssetCenter Server as an NT service

If you launch AssetCenter Server as an NT service, you can control its functioning via the Web.

This section deals with:

- Launching AssetCenter Server as an NT service
- Accessing the AssetCenter Server service via the Web
- Driving the AssetCenter Server service via the Web

## Launching AssetCenter Server as an NT service

During the installation of AssetCenter:

- **AssetCenter Server** is installed as an NT service; It is not started.
- Web access to the **AssetCenter Server** service is not enabled.

Note: Before installing the AssetCenter Server, we recommend creating a Windows NT user account on the machine on which this service will be installed. Next, we recommend installing the AssetCenter Server service under this account (Warning: The account must have the necessary rights to start the AssetCenter Server service and establish a connection with the database engine).

## Enabling Web access to the AssetCenter Server service

To enable Web access:

- ✚ Edit **AmSrvcf.ini** which can be found **in the** `\\AmSrv\Bin32` **subfolder of** the AssetCenter installation folder.
- ✚ In the [GLOBAL] section, modify the value of the "WebAdmin" key:
  - ❖ If WebAdmin = 1, Web access is enabled.
  - ❖ If WebAdmin = 0, Web access is disabled.
- ✚ In the [GLOBAL] section, check the value of the TCP/IP port used by the AssetCenter Server service. This value is stored in the "WebPort" key and is by default equal to 82. Modify this value if this port is already used by another program.

## Start the AssetCenter Server service

To start the AssetCenter Server NT service:

- ✚ In Control Panel, click the Services utility
- ✚ Select the AssetCenter Server service.

Then, if you want to start the service straightaway:

- ✚ Click **Start**. In the case of the AssetCenter Server service, we advise against specifying any parameters directly in the **Startup Parameters** field.

If you want to configure the AssetCenter Server service:

- ✚ Click the **Startup** button.
- ✚ Specify whether you want the **Startup Type** to be:

- ❖ **Automatic:** In this case, the service is launched when Windows NT is started.
- ❖ **Manual:** In this case, the service must be launched manually by clicking **Start** in the Windows NT services utility.
- ❖ **Disabled:** In this case, the NT service cannot be launched.

## Accessing the AssetCenter Server service via the Web

Warning: In order to be able to access the AssetCenter Server service via the Web, the service must be started.

Then, to access the AssetCenter Server service:

- ↳ Launch your usual web browser.
- ↳ Enter the address of the computer on which the AssetCenter Server service is running, followed by the TCP/IP port number used by the AssetCenter Server service on this machine. The address of the computer and the port are separated by a colon ":".

Examples of addresses:

- ❖ "http://colombo.taltek.com:82".
- ❖ "http://laguardia.taltek.com:800".

It is also possible to enter the TCP/IP address of the computer on which the AssetCenter Server service is running, followed by the value of the port.

Example: "127.0.0.1:82".

- ↳ The Home Page is opened. Click the **Connect** button in this page.
- ↳ A window authorizing access to the AssetCenter Server service is displayed. Use it to enter:
  - ❖ A "UserName": "WebAdmin".
  - ❖ The password associated with the "WebAdmin" UserName. By default, this password is empty.

## Driving the AssetCenter Server service via the Web

This section describes the commands you have at your disposal once connected to the AssetCenter Server service.

### Connect to new database

Using this command, you can:

- Connect manually to an AssetCenter database.
- Configure AssetCenter Server to connect automatically to an AssetCenter database at startup. To do this:
  - ↳ Check the **Reconnect at startup** option.
  - ↳ Enter the name of the database connection to which the AssetCenter Server service must reconnect automatically.
  - ↳ Specify the **Login** name and password.

Note: You can also configure the automatic connection of the AssetCenter Server service via the "AutoLogin" key in the "Database" section of the **AmSrvcf.ini** file.  
 AutoLogin = 0: Automatic connection disabled.  
 AutoLogin = 1: Automatic connection is enabled.

## Status of the server

This command displays the last 100 messages from the AssetCenter Server log file. These messages are similar to those shown in the main window in the GUI version of AssetCenter Server.

Click "Clear" to clear the all the messages displayed.

Note: The maximum number of messages cannot be modified.

## Configure

Using this command, you can define which modules are to be verified.

Note: You cannot modify the verification schedules concerning the modules via the Web. To do this, you must use the **Options/ Configure** menu item in AssetCenter Server's graphical interface.

## Activate

Select this command to activate certain verifications straightaway.

## WebAdmin password

Use this command to modify the "WebAdmin" password.

By default, this password is empty.



## Exit

Click this command to disconnect from the AssetCenter Server service.

Note: There is an automatic disconnection option in case of inactivity. This option is defined by the "TimeOut" key in the [SESSION] section of the **AmSrvcf.ini** file. By default, it is set to 10 minutes.



# Chapter 13 - Managing messages

AssetCenter gives you the possibility of managing two types of messages:

- Messages issued from AssetCenter and sent to the AssetCenter database via its internal messaging system.
- Messages created in AssetCenter and sent via an external messaging system.

This chapter contains information on the following:

- Overview of messaging
- Prerequisites for sending messages via an external messaging system
- Configuring AssetCenter to use messaging systems
- Creating messages
- Common connection problems

---

## Overview of messaging

### How to issue messages

Messages are generated via **Messaging** actions. These actions must first be created in order for a message to be able to be sent.

The action is triggered in different ways:

- Manually, by selecting the action in the list given by the **Tools/Actions** menu item.
- Automatically, via AssetCenter Server.
- Automatically by AssetCenter.

## How to consult messages

### Consulting messages sent to the internal messaging system

An agent test for the arrival of new messages and informs the users of the presence of new messages.

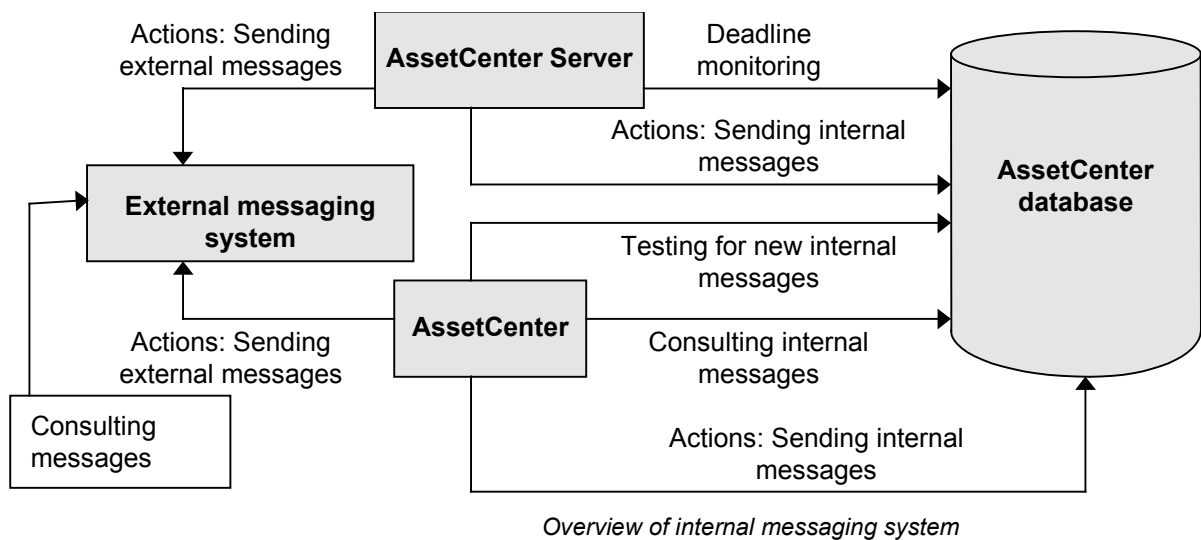
These messages can be consulted:

- ❖ Using the **Tools/ Messages** menu item.
- ❖ From the message box which informs of the presence of new messages.

### Consulting messages sent to an external messaging system

The recipient of these messages can consult them in the usual way under the external messaging system client.

## Overview



## Acknowledgement of receipt

Note: You cannot receive an acknowledgement for a message sent via the AssetCenter internal messaging system, or via a MAPI or SMTP messaging system.

To receive acknowledgment, check the **Acknowledgment** box (SQL name: bAcknowledgment) field in the detail of the **Messaging** type action.

This acknowledgement will be sent to the address specified by the **EMail** field (SQL name: EMail) in the **General** tab of the employee who opened the AssetCenter database (in the departments and employees table).

---

## Prerequisites for sending messages via an external messaging system

Supported messaging standards and the required environment for AssetCenter's internal messaging systems are described in the manual entitled "Installation and Upgrade Guide", chapter "Installing AssetCenter for the first time", section "Messaging system".

---

## Configuring AssetCenter to use messaging systems

In order for AssetCenter and AssetCenter Server to be able to send messages to an external messaging system, you need to:

### ***Populate certain fields in the employee detail screens.***

For further information, please refer to the following section:

- Populating employee details in order to use an external messaging system

### ***Configure and activate AssetCenter Server***

For further information, please refer to the following section:

- Managing deadlines with AssetCenter Server

### ***To configure the frequency of testing for new messages***

Using the **Tools/ Options** menu item, **Messaging** tab.

## **Populating employee details in order to use an external messaging system**

### **Overview**

In order for a message to be able to be sent, AssetCenter needs to know the account of the sender and the messaging address of the recipient:

- The account of the sender is identified by the **Account** (SQL name: MailLogin) and **Password** (SQL name: MailPassword) fields in the table of departments and employees (**Messaging** tab).
- The recipient is identified by the **EMail** field (SQL name: EMail) in the table of departments and employees (**General** tab).

You must populate these fields for both senders and recipients of messages.

### **Populate the *Account* field**

This field allows you to:

- Identify the messaging "account" (sometimes called "profile") of the sender of messages,
- Identify, in the case of VIM messaging systems, the recipient of acknowledgements of receipt.

Prefix the account name by "VIM:", "MAPI:" or "SMTP:" according to the standard of the messaging system used.
---

To find the exact name of the messaging account, here are some tips:

Messaging system	Where to find the account name	Sample account	Corresponding value in AssetCenter
Lotus Notes 4.x	"File/Tools/User ID", "Basics" icon, "Name" field.	Christiane Lupin/TALTEK	VIM:Christiane Lupin/TALTEK
Microsoft Exchange 97	Always use this profile: "Default Exchange Profile".	Default Exchange Profile	MAPI:Default Exchange Profile
Microsoft Outlook 97	By default: "Microsoft Outlook". To verify in Microsoft Outlook: "Tools/Options" menu item, "General" tab, name found next to the "Always use this profile" button.	Microsoft Outlook	MAPI:Microsoft Outlook
MAPI standard messaging systems	Under Windows NT, you can search an account name in the registry. To do this: <ul style="list-style-type: none"> <li>❖ Execute "regedit.exe" (or "regedt32.exe").</li> <li>❖ Go to the following key: <b>HKEY_CURRENT_USER\Software\Microsoft\WindowsNT\CurrentVersion\Windows Messaging Subsystem\Profiles</b></li> </ul> The account name can be found at the next level of the hierarchy.	Microsoft Outlook	MAPI:Microsoft Outlook
Netscape Communicator 4.05	"Edit/ Preferences" menu item, "Mail & Groups/ Mail Server" category in the list of preferences.  The first frame at the right of the screen displays three items of information: the user name, the outgoing mail server (SMTP) and the incoming mail server (POP3).  The account name is a concatenation of the following: <b>&lt;User name&gt;@&lt;incoming mail server&gt;</b>	clupin@mail.talte k.com	SMTP:clupin@mail .taltek.com

If the **Account** field is empty or badly filled in, the messaging action fails.

### Populate the *Password* field

This is the password associated with the **Account**.

The password allows you to activate the messaging system when sending messages. It is encrypted for security reasons.

### Populate the *E*Mail field (SQL name: EMail)

This field indicates the messaging address of the recipient.

#### Address with form <Messaging engine>:<Messaging address>

<Messaging engine> can be:

- ❖ AM: to force the utilization of AssetCenter's internal messaging system.
- ❖ MAPI: to force the utilization of a MAPI standard messaging system (Internet Mail, Microsoft Outlook, etc).
- ❖ VIM: to force the utilization of a VIM standard messaging system (Lotus Notes, etc.).
- ❖ SMTP: to force the utilization of a SMTP standard messaging system (Internet standard).

<Messaging address> has the usual form corresponding to the messaging system selected. Internal messaging addresses are the same as the "Logins".

Examples of addresses:

- ❖ AM:Admin
- ❖ MAPI:CathyBernard@taltek.com
- ❖ VIM:Cathy Bernard / TALTEK

### Case of the "Admin" user

A user with the name "Admin" is automatically created in your database. You must populate the user's **E**Mail (SQL name: EMail), **A**ccount (SQL name: MailLogin) and **P**assword (SQL name: MailPassword) fields. In this way, the administrator "Admin" can send and receive messages.

"Admin" receives messages in the following cases:

- When the "Admin" is clearly designated as a recipient. This is the case, for example, when AssetCenter Server addresses messages to the administrator of AssetCenter.
- When a message fails to be sent to a recipient.



"Admin" can also issue messages. This is particularly useful when AssetCenter Server is connected to the database under the "Admin" account login: AssetCenter Server can thus use the messaging account of the administrator "Admin" in order to send messages.

Warning: The appropriate messaging system clients must be installed on the machine on which AssetCenter Server is launched in order for it to be able to send messages.

---

## Creating messages

In order to create a message, you need to define and execute a **Messaging** type action.


The creation of "Messaging" type actions is described in the "**Reference Guide: Administration and Advanced Use**", chapter "Defining actions", section "Creating an action", sub-section "Populating the Messaging tab".

---

## Common connection problems

When a problem is encountered in sending a message, the administrator is sent a message to notify them of the problem.

### Test to validate a connection to a messaging system

- ↳ Launch AssetCenter Server.
- ↳ Connect to a database.
- ↳ Click .

### Test to perform if a problem is encountered

- ↳ Launch AssetCenter and connect to the database as "Admin".
- ↳ Create a new **Messaging Type** (SQL name: seActionType) action specifying a given recipient.

- ✚ Trigger the action using the **Tools/ Actions** menu item.
- ✚ Verify that the recipient has received the message and that the "router" has not sent an error message to your messaging system (unknown recipient).
- ✚ Consult any error messages.

## Troubleshooting

**"Connection to messaging system 'XXX': no messaging system specified. Verify the prefix of the messaging account in the 'Profile' tab in the employee detail."**

You need to prefix the **Account** field (SQL name: MailLogin) in the **Messaging** tab in the employee detail by:

- "MAPI:" if you use a MAPI standard messaging system (Microsoft Outlook, Microsoft Exchange, etc.)
- "VIM:" if you use a VIM standard messaging system (Lotus Notes, CCMail, etc.)
- "SMTP:" if you use an SMTP standard messaging system (Internet standard)

**"Unable to connect to messaging 'XXX'."**

The **Account** field in the **Messaging** tab of the employee detail is correctly prefixed by "MAPI:" or "VIM:", but the name of the account is incorrect. Verify that this has been correctly entered.

**"Messaging account 'VIM': a password must be given (this cannot be empty)."**

If you use a VIM messaging system, you must specify a password in the **Password** field (SQL name: MailPassword) in the **Messaging** tab of the employee detail. The password cannot be empty.

**"Messaging account 'XXX': incorrect password."**

The password specified in the **Password** field in the **Messaging** tab of the employee detail is incorrect.

**"Message not sent to 'XXX': messaging not available."**

This reveals a problem in the "win.ini" file.

In order for AssetCenter to work correctly with MAPI messaging systems, the "win.ini" file must contain the following lines in the [Mail] section:

```
MAPI=1
MAPIX=1
```

In order for AssetCenter to work correctly with VIM messaging systems, the "win.ini" file must contain the following line in the [Mail] section:

**SMI=1**

In order for AssetCenter to work correctly with SMTP messaging systems, the "win.ini" file must contain the following lines in the [Mail] section:

**SMTP=1**

**SMTPServer=<Outgoing mail server>**

The following lines are optional:

**SMTPPort=<Port number of outgoing mail server >** (25 by default)

**SMTPTimeOut=<Timeout value>** (20 seconds by default)

These four lines are not exclusive.

If one of these lines is not present or its value is set to 0, you need to verify the proper functioning of the corresponding messaging system. To do this, use a program such as Microsoft Internet Mail for MAPI and Lotus Notes for VIM. If the messaging is working and you are not in the case described below, you can modify the [Mail] section of the "win.ini" file as shown above.

---

Important note: If MAPI is set to 1 but not MAPIX, it is possible that the messaging system does not support extended MAPI. Verify whether this is the case. AssetCenter cannot function correctly if the messaging system is not compatible with extended MAPI.

---

#### **"Error opening VIM session: password required"**

With "VIM" messaging systems a password is required. Add it under your messaging system and indicate it in AssetCenter in the **Password** field (SQL name: MailPassword) in the **Messaging** tab of the employee detail.

#### **"Error opening VIM session: Incorrect password"**

The password is invalid. Modify the value of the **Password** field (SQL name: MailPassword) in the **Messaging** tab of the employee detail.





