

Peregrine

AssetCenter

Administering the database



© Copyright 2002 Peregrine Systems, Inc.

All Rights Reserved.

Information contained in this document is proprietary to Peregrine Systems, Incorporated, and may be used or disclosed only with written permission from Peregrine Systems, Inc. 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® and AssetCenter® are trademarks of Peregrine Systems, Inc. or its subsidiaries.

This document and the related software described in this manual are supplied under license or nondisclosure agreement and may be used or copied only in accordance with the terms of the agreement. The information in this document is subject to change without notice and does not represent a commitment on the part of Peregrine Systems, Inc. Contact Peregrine Systems, Inc., Customer Support to verify the date of the latest version of this document.

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.

This product contains software components developed by Apache Software Foundation (<http://www.apache.org>).

This edition applies to version 4.1.0 of the licensed program

AssetCenter

Peregrine Systems, Inc.
Worldwide Corporate Campus and Executive Briefing Center
3611 Valley Centre Drive San Diego, CA 92130
Tel 800.638.5231 or 858.481.5000
Fax 858.481.1751
www.peregrine.com



Table of Contents

Introduction (Administration)	17
Who is the Administration module intended for?	17
What does the Administration module do?	17
Chapter 1. AssetCenter Database Administrator - GUI	19
Introduction	19
Graphical interface	20
Presentation of the graphical interface	20
Manipulating files	22
Edit functions	24
Application options	25
Chapter 2. Standard database description files	27
Foreword	27
Definition of a database	29
Structure of the database.txt and tables.txt files	29
Description of the tables	30
Identifying AssetCenter tables	31
Table description strings	31
Description of fields	32
Identifying AssetCenter fields	32

Field types and data-entry formats	33
dtLastModif fields	36
"FullName" fields	37
Description of links	38
Identifying links	38
Type of a link	39
Cardinality of a link	40
Intermediary tables	41
Contextual links	42
Description of indexes	43
Identifying indexes	43
Uniqueness of indexed field values	43
Customizing the database	44
Customizing tables	44
Customizing fields and links	44
Customizing the indexes	45
Default values for fields and links	46
Counters in field default values	47
Errors in the description of default values	48
HTML tags recognized in help on fields and links	48
Chapter 3. Creating an AssetCenter database	51
Creating an empty AssetCenter database using your DBMS.	52
Warnings and recommendations	52
Creating a database at the DBMS level	54
Creating the DBMS login	64
Changing the DBMS used with AssetCenter	64
Destroying an AssetCenter database	65
License rights and available functions	65
Declaring a connection to an empty shell	66
Creating a database with AssetCenter Database Administrator	67
Options of the Database frame	68
Options of the Create database frame	68
Options of the Create system data frame	68
Options of the Import extra data frame	69
Selecting a license file	70
Entering the administrator password and "physically" creating the database	72
Modifying the license	72
Modifying the license using AssetCenter Database Administrator	72
Changing your DBMS	73
Creating a dump	74
Restoring a database from a dump	74

Chapter 4. Creating a database description file	75
Introduction	77
Database description parameters	77
Syntax for description templates	80
Fixed text	81
Comments	81
Including another template	81
Browsing, sorting and filtering components	82
\$if...\$else...\$elseif...\$endif conditions	85
Functions available in description templates	85
Processing the value of a property using a function defined in a template	89
Deleting the end-of-paragraph mark	91
Counting the number of browsed components	92
Defining a global variable for the template	92
Information about certain database description parameters	92
Database instance	93
Table instance	94
Field instance	96
Link instance	100
Index instance	103
Script instance	104
Instance FeatDesc	106
Instance FeatParamDesc	107
Instance FeatClassDesc	108
Instance CalcFieldDesc	109
Instance View	110
Instance Page	111
Instance PageItem	112
Instance SysEnumValue	113
Instance String	113
Instance ScriptField	114
Global variables	114
Chapter 5. Diagnostic and repairs of a database	117
Chapter 6. Updating your database structure	121
Chapter 7. Accessing the database via ODBC	123
Overview of access to the AssetCenter database	123
Installing the ODBC driver	123
When to use the ODBC driver	124
Data items accessible via the ODBC driver	125

Which ODBC connection to use	125
Example: Creating a report under Crystal Reports with the ODBC driver . . .	126
Chapter 8. Customizing the database	127
Customizing existing objects	127
Customizing tables	129
Customizing objects	131
Creating new objects	136
Methods of creation	136
Creating a table	136
Creating a field, link or index	137
Creating a detail	140
Creating action buttons	141
Creating a page	142
Adding a page to a detail	145
Saving modifications	145
Verifying your modifications	146
Important note	146
Chapter 9. Database options	147
Conventions	147
Chapter 10. Import	151
Overview of importing data	152
Importing data from a single text file	152
Importing data from several text files	152
Importing data from a full database	152
Recommendations	153
Default values for mandatory fields	153
Avoid using "Id" field as reconciliation keys	154
Backup the AssetCenter database	154
Avoid simultaneous access to the AssetCenter database	154
Constraints to be respected when importing data into a field	154
According to the value of the "UserType" property of the target field . .	155
According to the value of the "type" property of the target field	157
According to the value of other properties of the target field	157
Importing text files or an ODBC database	158
Before importing text files	158
Step 1: Select the text files or ODBC database to be imported	159
Step 2: Define the parsing of files or tables to be imported	160
Step 3: Describe the fields to be imported	164
Step 4: Map source fields to target fields in the AssetCenter database . .	166

Step 5: Map each text file or source table to a target table	167
Step 6: Map the fields to be imported to fields in the AssetCenter database	168
Step 7: Add additional calculated fields in the source file	174
Step 8: Special cases	178
Step 9: Examples of using keys	179
Step 10: Configuring the transfer	183
Step 11: Data transfer	185
Saving and executing an import script	186
To save a script	187
To modify a script	187
To create a new script when you are defining import settings	188
To execute a script	188
Executing an import script from a command prompt	189
How it works	189
Syntax	189
Chapter 11. User profiles	191
Importance and overview of database access rights	192
Defining access conditions for each user	192
Managing user connections	193
How to ensure data security and confidentiality	193
Definitions	193
Definition of a user profile	194
Definition of a user right	194
Definition of a functional right	194
Definition of an access restriction	194
Defining access conditions	195
Defining user profiles	195
Defining user rights	195
Defining functional rights.	197
Defining access restrictions	199
Defining AssetCenter users	201
Defining a database administrator	202
Defining a non-administrative user	202
Managing user connections	202
Database access types	202
How connection slots work	204
Destroying connection slots	205
Managing passwords	207
Password for the Admin login	207
Modifying the password for the Admin login	208
User password	208

Lost passwords	209
Chapter 12. AssetCenter Server	211
Overview of AssetCenter Server	211
Executing AssetCenter Server	213
Recommendations	213
Launching AssetCenter Server	213
Executing AssetCenter Server manually under Windows	215
Executing AssetCenter Server on Unix	215
Using the security-integrated Windows NT, 2000 or XP	216
If you exit AssetCenter Server	221
AssetCenter Server and the messaging system	221
Main screen of AssetCenter Server	222
General options of AssetCenter Server	223
Database reconnection delay	223
Messaging reconnection delay	223
Log file	223
Time zones	224
Configuring the modules monitored by AssetCenter Server	225
Introduction	225
Verification schedules	226
Verify history lines module (History)	228
Verify stocks module (Stock)	228
Verify alarms module (Alarms)	229
Calculate rents module (Rent)	231
Calculate stipulated loss values module (LostVal)	234
Split expense lines in cost centers module (CostCenter)	234
Verify database server time zone module (TimeZone)	236
Signal presence to database server module (UpdateToken)	236
Search for new workflow execution groups module (WorkflowFinder)	237
Execute workflow rules for execution group 'G' modules	237
Add the computers listed in the NT domain to the database module (AddCpu)	239
Add NT users to the database module (AddUser)	242
Create assets, consumptions, etc. corresponding to items received module (Delivery)	244
Send the scanner to the computers module (SendScan)	245
Get the results of the scanners module (GetFsf)	250
Update the database with the results of the scanners module (IddAc)	254
Verify null-identifier records module (History)	256
Purge the input-events table module (PurgeEventInTable)	257
Purge the outgoing-events table module (PurgeEventOutTable)	257
Update statistics for tables module (Stats)	258

Activating the verification immediately	258
Administering AssetCenter Server by the Web	258
Launching AssetCenter Server as a service	259
Accessing the AssetCenter Server service via the Web	260
Driving the AssetCenter Server service via the Web	261
Chapter 13. Messaging	265
Overview of messaging	265
Configuring AssetCenter to use messaging systems	266
Multiple recipients	267
SMTP protocol	267
MAPI protocol	270
VIM protocol	272
AM protocol	275
Common connection problems	276
Test to validate a connection to a messaging system	277
Test to perform if a problem is encountered	277
Troubleshooting	277
Chapter 14. Using AssetCenter as a DDE server	281
Definition of a DDE server	281
DDE call mechanisms	281
DDE service	282
DDE topic	282
DDE command	282
Introduction to DDE commands	283
Steps to follow	283
Particularities	283
Global commands	284
Connect(<Cnx>, <User>, <Password>)	284
Disconnect()	285
ExecuteAction(<ActionName>)	286
ListAllTables([Mask])	286
ListOpenTables([Mask])	287
OpenView(<ViewName>)	288
Commands associated with a table	289
OpenTable(<Table>)	289
CloseTable(<Table>)	290
<Table>.GetRecordCount()	290
<Table>.SetViewMode(<Mode>)	291
<Table>.SetRecordMode(<Mode>)	292
<Table>.ListAllFields([Mask])	293
<Table>.ListAllLinks([Mask])	294

<Table>.SetFilter(<Condition>)	295
<Table>.SetSelection(<Condition>)	295
<Table>.GetSelectionId()	296
Commands associated with a table and a field or a link	297
<Table>:<Object>.AddLink()	297
<Table>:<Object>.GetValue()	298
<Table>:<Object>.Hilight()	299
<Table>:<Object>.RemoveLink()	299
<Table>:<Object>.SetFocus()	300
<Table>:<Object>.SetValue(<Value>)	301
<Table>:<Link>.SetValueWhere(<Condition>)	302
<Table>:<Object>.Show()	303
Introduction to Examples of DDE calls	303
First scenario: AssetCenter internal DDE calls	304
Answering the call	304
Entering data	305
Creating the feature	306
Second scenario: DDE calls from Excel	306
Description of the macro	306
Source code of macro	307
Third scenario: DDE calls from Visual Basic	308
Source code of the program	308
Before starting	308
Executing the program	309
Chapter 15. Optimizing AssetCenter for use in a WAN environment	311
Options in the Edit/ Options menu	311
Lists	313
Setting the parameters for a particular list	313
Setting list parameters at the database level	314
Setting list parameters at the AssetCenter client level	315
Stripping down screens	316
Connection cache	317
Access restrictions	317
Applying the configuration of one client to other clients	317
Chapter 16. AutoCAD integration	319
Presentation of AutoCAD/AssetCenter integration	319
Installation	320
Creating an AssetCenter database compatible with AutoCAD	320
FacilityCenter users	320
Configuring Connect-It	320
Using AutoCAD integration	322

Graphic interface	322
Overview	322
Chapter 17. Using a test database	325
Copying your working database	326
Categorizing data	326
Exporting the database structure	327
Importing the new database structure	328
Migrating data with Connect-It	328

List of Figures

1.1. Database opening screen	22
2.1. Intermediary tables - utilization	41
2.2. Link between the Assets table and the Fixed assets table	41
2.3. Links between the History table and a target table	42
3.1. Database - Creation window	67
8.1. Administrator - main screen	128
10.1. Import - Selecting the type	151
11.1. User profile - detail window	192
11.2. User profiles - Access restrictions	199
13.1. Messaging system - overview	266

List of Tables

2.1. Fields - type	33
2.2. Data types - Entry format	34
2.3. Value of the User types field	35
2.4. Various types of existing links	39
2.5. Nature of the information stored by links	39
2.6. HTML tags recognized in help on fields and links	49
2.7. Reserved characters	49
3.1. Important server parameters when creating the Oracle instance	55
3.2. Important database and server parameters	57
3.3. Important server parameters	59
4.1. Database instance properties	93
4.2. Table instance properties	94
4.3. Table instance objects	95
4.4. Field instance properties	96
4.5. Field instance objects	98
4.6. Possible values for the "Type" property	99
4.7. Possible values for the "UserType" property	99
4.8. Link instance properties	100
4.9. Link instance objects	102
4.10. Possible values for the "Type" property	102
4.11. Possible values for the "UserType" property	103
4.12. Index instance properties	103

4.13. Index instance objects	104
4.14. Script instance properties	104
4.15. Script instance objects	105
4.16. Properties of the FeatDesc instance	106
4.17. Objects of the FeatDesc instance	107
4.18. Properties of the FeatParamDesc instance	107
4.19. Objects of the FeatParamDesc instance	108
4.20. Properties of the FeatClassDesc instance	109
4.21. Objects of the FeatParamDesc instance	109
4.22. Properties of the CalcFieldDesc instance	109
4.23. Objects of the CalcFieldDesc instance	110
4.24. Properties of the View instance	110
4.25. Objects of the View instance	111
4.26. Properties of the Page instance	111
4.27. Properties of the PageItem instance	112
4.28. Objects of the PageItem instance	113
4.29. Properties of the SysEnumValue instance	113
4.30. Properties of the String instance	113
4.31. Properties of the ScriptField instance	114
4.32. Global variables	114
7.1. Advantages of using the ODBC driver	124
9.1. Database options	148
12.1. AssetCenter Server program menus and icons	222
12.2. Days of monitoring	226



Introduction (Administration)

PREFACE

Who is the Administration module intended for?

The Administration module is intended for all organizations using AssetCenter.

It is generally implemented by the following persons:

- Network administrator
- Database administrator
- AssetCenter administrator

What does the Administration module do?

The Administration module enables you to perform the following tasks:

- Manage an AssetCenter database
- Manage connections to an ODBC database
- Manage AssetCenter

- Manage AssetCenter Server
- Manage user profiles

Managing these tasks enables you to:

- Create, update, maintain, customize a database
- Import data
- Create, update, delete connections to an ODBC database
- Optimize AssetCenter for use over a network
- Implement automatic monitoring procedures
- Create, update, delete user profiles

1 | AssetCenter Database Administrator - GUI

CHAPTER

Introduction

AssetCenter Database Administrator is an administration tool for the AssetCenter databases. It enables you to perform a large number of operations, for example the:

- Creation of a database.
- Customization of objects stored in the database (tables, fields, links, indexes, screens, etc.).
- Repair a damaged database.
- Update the structure of a database.
- Extract information contained in a database.

This tool is intended for the administrator and requires a certain number of precautions:

- Since this tool enables you to modify the structure of an AssetCenter database, we recommend limiting its use to users with the appropriate skills.

- AssetCenter also enables you, when you use the "Admin" login, to modify the database (configuration of objects, links, etc.). However, you should never simultaneously modify the same database using AssetCenter and AssetCenter Database Administrator.
- 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.

Graphical interface

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

Note:

At startup, the main AssetCenter Database Administrator screen appears entirely in gray as long as no file has been loaded. You can automatically load the last document used by setting the **At startup, automatically load last document used** option to **Yes** (in the **Documents** section accessible by the **Edit/ Options** menu).

Presentation of the graphical interface

AssetCenter Database Administrator's user interface is composed of three panes:

- A menu bar, containing a toolbar.
- A pane containing a list of AssetCenter database tables.
- A main pane, also called a Customization pane, which regroups the information about a table's objects.

This graphical interface respects the same ergonomic database rules as AssetCenter, such as for consultation and creation. For more information about these rules, refer to the **Using AssetCenter** guide.

Selecting a table

To select a table, click its title in the left-hand pane of the user interface.

Select the displayed object type.

For any given table, you can select the object type displayed in the main pane using the **Display** menu. The available objects are:

- Fields
- Links
- Indexes
- Details
- Pages

Using the customization pane

The Customization pane is divided into three parts:

- The first part (on top) displays the general information about the selected table.
- The second part (in the middle) lists all the objects of a given type appearing in the table. You can choose the type of object to appear using the **Display** menu.
- The third part (on the bottom) contains the information about the selected object.



Note:

Only certain information can be customized. The non-editable values appear in grayed-out fields.

Manipulating files

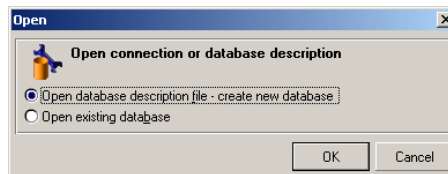
The **File** menu regroups all the functions relating to loading and saving a file.

Opening a file

Select the **File/Open** menu.

Selecting this menu displays the following screen:

Figure 1.1. Database opening screen



This screen enables you to choose one of two AssetCenter Database Administrator functions:

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

You can open an AssetCenter Database Administrator session by selecting one of these two options.

 **Note:**

In the last part of the **File** menu, AssetCenter Database Administrator lists by default the last four opened documents. You can thus quickly open one of these documents by directly clicking it in the menu. To set the number of documents kept in memory, you can use the **Maximum number of documents to store in memory** option (in the **Documents** section, available via the **Edit/Options** menu).

Open database description file - create new database

To create a new database AssetCenter Database Administrator needs a description of that database. This description comes in the form of a file that contains the structural information relating to an AssetCenter database. It plays the role of the data model during the creation of the database.


By validating this choice, you must provide AssetCenter Database Administrator the description file **gbbase.dbb** located in the **Config** sub-folder of the AssetCenter installation folder. We recommend that you make a copy of this file in order to always have in your possession a standard description file.

Note:

We also advise you to systematically generate a database description file for your databases (using the **File/Save as** menu) and to make a copy of this file: You will need it if ever you want to repair your database.

Open an existing database

You must connect to the AssetCenter database before being able to customize it. Selecting this option will display the connection that is also used for AssetCenter.

We will not list all the possibilities available to edit this connection, accessible via the  button. If you want an exhaustive list of these possibilities, refer to the **Using AssetCenter 4.1.0**.

Closing a file

The **File/Close** menu enables you to close an AssetCenter Database Administrator session. If you made any modifications, AssetCenter Database Administrator asks you if you'd like to save them before closing the session.

To signal that a modification has been made to a database - and before validating these modifications - AssetCenter Database Administrator adds an asterisk to the name of the document in the application's title bar.

Saving a file

Two modes of saving a document are available:

- The **File/Save** menu enables you to save the modifications made to the database description file or to the database.
- The **File/Save as** menu enables you to perform the following tasks:
 - 1 If a database description file is open, this menu enables you to save the structure of the database in a new database description file.
 - 2 If a database file is open, this menu enables you to save the structure of the database in a database description file.

Quit the application

This menu enables you to quit AssetCenter Database Administrator. If you made any modifications during the session, AssetCenter Database Administrator asks if you want to save them.

Edit functions

AssetCenter Database Administrator proposes all the classic edition functions:

Cut/Copy/Paste

The **Edit** menu regroups all the operations that you can perform using a selection:

- **Cut** (Ctrl+X) to cut a selection.
- **Copy** (Ctrl+C) to copy a selection.
- **Paste** (Ctrl+V) to paste a selection.

 **Note:**

These functions only work in editable field zones.

Perform a search

AssetCenter Database Administrator proposes an advanced text searching function, accessible via the **Edit/Find** menu.

As you have already seen, the database structure is composed of objects (tables, fields, links, etc.), which are characterized by the information relating to their properties. For example, the **SQL name** of a table is a property of that table, just as the **Type** of a field is one of the properties of that field.

The global structure of a database is thus composed of hierarchical objects, each one possessing one or more properties.

The search function proposed by AssetCenter Database Administrator is performed on the integrality of the database structure. When you perform a search, the software looks through all the database objects and their properties. The list of results is then created in the software's memory, and you can browse this list by using the **Edit/Next** or **Edit/Previous** menus (or their keyboard shortcuts: F3 and Shift+F3, respectively).

 **Note:**

You can specify the direction of the search by using the **Up** or **Down** options in the dialog box.

Application options

The **Edit/Options** menu enables you to define your preferences of use for AssetCenter Database Administrator. Each option is described in the **Description** zone of the dialog box.

2 | Standard database description files

CHAPTER

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

 **Note:**

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 perform functions such as importing data, accessing the database using external tools, writing queries, etc., you need to understand the structure of the database. This understanding enables you to gather necessary 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:

- The **database.txt** and **tables.txt** files: They contain the complete structure of the database. These files are located in the **Infos** sub-folder of the AssetCenter installation folder.
-

 **Note:**

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

In order for these description files to reflect the customizations made in your database, you should use AssetCenter Database Administrator with a connection to your database.

- AssetCenter Database Administrator program: It enables 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 enable you to create files in **rtf** or **html** format.
-

 **Note:**

Among the templates provided with the standard version of AssetCenter, one of them, **Dbdict.tpl**, enables you to export the customization data (including information about features, calculated fields, configuration scripts, etc.) from your database to a standard text file. Used with a source control tool, this description file can help you 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 about the assets you are managing. For simple installations, these files are all located in the same folder, either on the local disk drive or on a network file server.

AssetCenter comes with a demonstration database.

With AssetCenter, you can create several databases and open them one-at-a-time per each user session. In addition, there can be several user sessions connected to a database at the same time. The program uses a transactional method to update the data in the database, which means that AssetCenter 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, which are located in the **Infos** sub-folder of the AssetCenter installation folder.

 **Note:**

These files were created using AssetCenter Database Administrator and the **dbase.tpl** and **tables.tpl** templates. These templates are located in the **Infos** sub-folder of the AssetCenter installation folder.

Here is the format of these files:

- Type: text
- Separator: tab
- Character set: ANSI

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

We have included 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 SQL name.
- For each table, the following information appears in succession:
 - 1 Fields
 - 2 Links
 - 3 Indexes
- Fields, links and indexes are sorted by 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 generated.

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

- One line per table.
- Tables are sorted by 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 generated.

 **Note:**

Note: The SQL names identify the tables, fields, links and indexes. These names do not change from one AssetCenter language version to the next; they are always the same.

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:

- An "SQL name". The SQL names of AssetCenter tables are in English and have the "am" prefix. These names do not change from one AssetCenter language version to the next; they are always the same.
- A "Label": This is the name of the table as it appears 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 AssetCenter tables.

In an AssetCenter table, the description string describes the records appearing in drop-down lists that enable 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.

Example :

```
String: [[Brand] [Product.Model] [AssetTa
```

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 appears 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 appears 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:

- An "SQL name". SQL names are in English. SQL names do not change from one AssetCenter language version to the next; they are always the same. 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 (time span)
"mem"	Comment
None	Character string

- A "Label". AssetCenter uses labels when it 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 the **General** tab of AssetCenter Database Administrator.
- Read the **database.txt** file, located in the Infos folder of the AssetCenter installation folder. The **Field Type** column details the field types used.
- Display the context-sensitive help for that field.

Type of the field of a table ("Type")

Provides the field's storage format.

Table 2.1. Fields - type

Value given in Field Type column in database.txt file	Value in Type field in AssetCenter Database Administrator	Signification
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.

Value given in Field Type column in database.txt file	Value in Type field in AssetCenter Database Administrator	Signification
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) can stores images and forms, for example, without size restriction.
MEMO	Long text field	Variable-length text field (comments). Certain DBMSs have limitations concerning these fields. Oracle for WorkGroups, 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 the **General** tab of 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:

Table 2.2. Data types - Entry format

Value given in Field data display and entry type column in database.txt file	Value in User type field in AssetCenter Database Administrator	Signification
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.

Value given in Field data display and entry type column in database.txt file	Value in User type field in AssetCenter Database Administrator	Signification
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
Time span	Duration	Duration type values. The authorized and default units are defined by the UserTypeFormat property.
System itemized list	System itemized list	You must select the value for this type of field 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	You select the value for this type of field from an itemized list that may be customized by the user.
Percentage	Percentage	A field with this format contains percentages. This Percentage 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 name	Table or field name	SQL name of a table or a field.
N/A	Unknown	Any other type.

You can have additional information about field formats if you set the **User type** field in AssetCenter Database Administrator to one of the following values:

Table 2.3. Value of the User typesfield

Value of User type field in AssetCenter Database Administrator	Additional information
Time span	<p>Display format.</p> <pre>%U1[1][d][%U2[1][d]]...[%Un[1][d]]</pre> <p>Syntax:</p> <pre>Ux</pre> <p>Being one of the following units:</p> <ul style="list-style-type: none"> • Y: year • M: month • D: day • H: hour • N: minute • S: second <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>Examples</p> <ul style="list-style-type: none"> • "+HM,...H: displays hours, then minutes. By default, you enter hours. • "%Yl%Ml%Dl": displays years, then months, then days.
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 with the SQL name "dtLastModif" exists in 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** field 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 field from that record. This value is preceded by a hierarchy, composed of the field values of the parent records to which the record is linked, all the way down 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).

```
Example : "/USA/Milwaukee/The Domes site/".
```

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

```
Example : "/Commercial  
Services/Telemarketing/Colombo,Gerald,P223/".
```

- Assets table: The full name of an asset consists of its asset tag preceded by the asset tags of the parent asset(s).

```
Example : "/P123/DD456/CM0125/".
```

 **Note:**

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 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 links

A link is identified by:

- Its "SQL name". SQL names do not change from one AssetCenter language version to the next; they are always the same.
- Its "Label".
- Its "Description".

Example of the **Location** link from the Portfolio items table:

- The SQL name of the source field in the source table (Portfolio items table) is "lLocaId".

- The SQL name of the target table (Locations table) is "amLocation".
- The SQL name of the target field in the target table (Locations table) is "lLocaId".
- The "database.txt" file indicates that the link's cardinality is "1": A given item can only have one location.

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. This is because several links can exist between AssetCenter tables. For example, between the Portfolio items table and the Departments and Employees table, the link whose label is **User** (SQL name: User) defines the item's user, and the link whose label is **Supervisor** (SQL name: Supervisor) defines the item's supervisor.

Type of a link

The following table lists the various types of existing links:

Table 2.4. 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:

Table 2.5. 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 a 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 Employees table is linked to the Locations table via a "1-->1" link: An employee can only be associated with one single location.
- "1-->N" links: A record in table A can be linked to several records in table B. For example, the Contracts table is linked to the Assets table via a "1-->N" link: A contract can cover 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. An 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. An n-n link between two tables in AssetCenter is represented by two "1-->N" links.

Intermediary tables

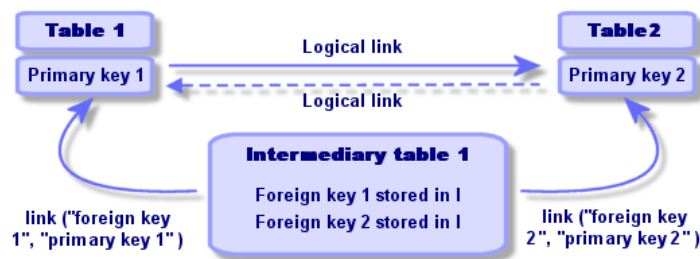
Intermediary tables are only used in the case of n-n cardinality logical links.

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.

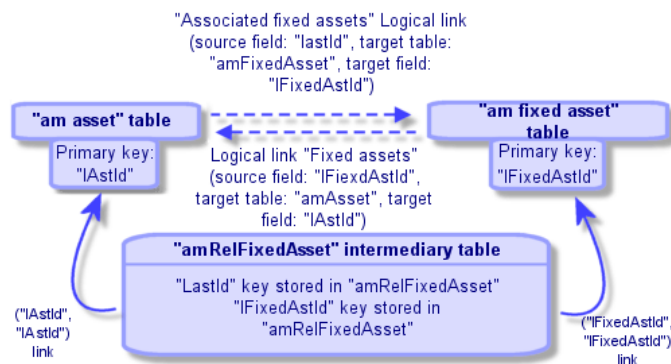
The following diagram explains how intermediary tables are used:

Figure 2.1. Intermediary tables - utilization



Example of the link between the Assets table and the Fixed assets table:

Figure 2.2. Link between the Assets table and the Fixed assets table



In this case:

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

 **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 History table and a target table:

Figure 2.3. Links between the History table 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 do not change from one AssetCenter language version to the next; they are always the same.
- 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 **Help on fields** table (amHelp), the "Help_TableNameFiel" index uses the following fields:

- **Table** (SQL name: TableName),
- **Field** (Nom SQL : FieldName),

This index is "unique or null". This means that it is not possible to find two help entries for the field in the same table.

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

Customizing the database

AssetCenter enables 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.

The following sections provide a detailed explanation of how to customize the various objects in the AssetCenter database.

Customizing tables

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

Customizing fields and links

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

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

 Note:

If you use the shortcut menu, changes are recorded: either when you close the database and confirm the validation message, or when you select the **Administration/ Save database configuration** menu.

- By using AssetCenter Database Administrator.
The **General** and **Scripts** tabs in AssetCenter Database Administrator allow you to define the:
 - Labels for fields and links.
 - Descriptions of fields and links.
 - Default values for fields and links.
 - Mandatory fields and links.
 - Fields and links for which you want to log modification history.
 - Read-only fields and links.

 Note:

You can also set the maximum size of a "text"-type field 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 text** tab:
 - "Description": explains the contents of the field or link.
 - "Example": provides examples of how to populate the field or link.
 - "Precautions": 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

- 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="BL"+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.

In the case of calculated fields

Calculated fields can only be used to calculate the default values of standard fields if their type is **Calculated string** or **Basic script**.

In the case of links going to the Comments table (SQL name: amComment)

It is not possible to assign a default value to the links going to this table.

Example

Comment (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 Administration/Counters menu.

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 can 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 are 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 date-type field, 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.

 Note:

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 online help.

Table 2.6. HTML tags recognized in help on fields and links

HTML Tag	Description
	Defines the font used to display the text following this tag. This font remains in use until the next change of font.
	Defines the font color used to display the text following this tag. This color remains in use until the next change of color.
	Increases the font size by "n" levels. This font size remains in use until the next change of font size.
	Decreases the font size by "n" levels. This font size remains in use until the next change of font size.
 	These two tags surround text to be displayed in bold.
<I> </I>	These two tags surround text to be displayed in italics.
	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 about 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.

Table 2.7. Reserved characters

Reserved character	Use the following tag to display this character
"<"	"<"
"&"	"&"
Non breaking space	" "

3 | Creating an AssetCenter database

CHAPTER

Here are the different steps in creating a database:

- 1 Creating an empty database using your DBMS.
- 2 Declaring the connection to an empty shell.

 Note:

You must create an empty shell in order to create a database. This task varies according to the DBMS used and is usually the job of an administrator. For more information, consult the documentation of your DBMS.

-
- 3 Creating the database with AssetCenter Database Administrator.
 - 4 Selecting the database license file. The license file determines your access rights, which correspond to your AssetCenter license agreement.
 - 5 Entering or modifying the administrator password. The administrator login and associated password are initially defined by the DBMS used to create the empty shell.

- 6 "Physically" creating the AssetCenter database. During this step, AssetCenter creates the full structure of the database (tables, fields, links, etc.) inside the empty shell.
- 7 Starting AssetCenter Server and connecting to the database. This step validates the license file and enables users to connect to the database.

Creating an empty AssetCenter database using your DBMS.

Warnings and recommendations

- Creating a database correctly is not an easy task and requires a high level of database administration skills.

We cannot explain all the various aspects of creating a database here; this falls within the domain of the documentation of the DBMS. On the other hand, we have included some items of information that will prove useful to skilled database administrators.

 **Warning:**

If you are not fully competent in the use of your DBMS, you must use the services of a specialized consultant. The way in which you configure the database has a direct impact on data security, the performance of AssetCenter and the scalability of your architecture.

There is no one single correct way to create a database. This depends on several factors:

- The platform on which the DBMS is installed (Windows NT, Unix, etc.).
- Your business rules.
- The database administration tools you usually use.
- If the database will be used by numerous users at the same time, you should optimize the DBMS performances (at the level of the counters

and the generation of identifiers). These operations must be performed by the database administrator, who should contact Peregrine Systems technical support to obtain the necessary instructions.

- We advise you not to install a DBMS on a server that is already host to numerous **services** (such as a domain controller, a DHCP controller, a DNS or a messaging service).

 **Note:**

The examples and rough estimates of size are made on the basis of the graphical administration tools supplied with the DBMS (other third-party tools are available).

Configuring the database server

By default, select a **RAID 10** configuration. If this isn't possible, or if the server is composed of several clusters of **RAID** disks, you can distribute the databases to several disks or disk clusters.

You can do this with the DBMS by creating data files on different logical volumes and by creating file groups.

To use the DBMS correctly, the read and write load should be balanced on all hard disks.

If the hard disks do not have the same speed, it is better to store the indexes on the fastest disks.

RAID 0 is the easiest way to distribute equally inputs and outputs on hard disks.

Comparative measures of performance of MS SQL Server can be obtained on machines with several GB of memory and with several **RAIDS 10** clusters.

Systematically use **RAID 1** and **RAID 10** for the disks that contain log files (**.log**), instead of using **RAID 5**.

RAID 5 is good if the **data** is stored on more than 6 hard disks. However, the performance will be inferior to that obtained with **RAID 10**.

Creating a database at the DBMS level

 **Note:**

While Microsoft SQL Server, Sybase Adaptive Server and UDB DB2 use the term **database**, Oracle uses the term **schema**.

Steps in the creation of the database with the DBMS

The creation of the database is carried out in several steps:

- 1 Creating and reserving space for the database.
- 2 Creating a login at DBMS level.
- 3 Attributing rights to one or several users. At the DBMS level, one login is enough. Nevertheless, you can create other DBMS logins that will be used in the AssetCenter connections. In this case, you must define update rights at the level of each table of the AssetCenter database.

These steps are described below for each supported DBMS.

Information to be taken into account

The AssetCenter database is made up of about:

- 350 tables
 - 1400 indexes
-

 **Tip:**

The parameter defining the **number of open objects** must be sufficient to handle to databases that you are going to create with AssetCenter Database Administrator.

Example under Sybase Adaptive Server: The number of open objects is defined by the **number of open objects** parameter in the **sp_configure** stored procedure.

Oracle

Table 3.1. Important server parameters when creating the Oracle instance

Parameter	Recommended value
Character set	<p>For Latin code-page category languages (English, French, German, Italian, Spanish, etc.), we recommend the WE8ISO8859P15 character set.</p> <p>With certain Oracle versions, this parameter cannot be modified once the database is created.</p>
db_block_size	<p>This parameter is expressed in bytes.</p> <p>We recommend using the value 8192.</p> <p>With certain Oracle versions, this parameter cannot be modified once the database is created.</p>
db_block_buffers	<p>Size of the database memory cache.</p> <p>This size is expressed in units of db_block_size.</p> <p>Examples to obtain a 200 MB cache:</p> <ul style="list-style-type: none"> • <code>db_block_buffers = 25000</code> <code>sidb_block_size = 8192</code> • <code>db_block_buffers = 100000</code> <code>sidb_block_size = 2048</code> <p>For a medium-sized server, the memory reserved for the database cache should represent 20% to 25% of the server's total memory.</p> <p>This recommendation must be adjusted if several users simultaneously access the system (the database cache can be slightly reduced in a way that the server's memory does not swap).</p>
shared_pool_size	<p>9,000,000 to 1,5000,000 bytes on an Oracle 8.0.x according to the number of users (10 to 50).</p> <p>If the user is installed in the multi-threaded MTS, then this parameter is much higher (together with the large_pool_size on Oracle 8i).</p>
log_buffer	<p>This parameter is expressed in bytes.</p> <p>We recommend using the value 163840.</p>
processes	<p><code><Number of concurrent users> + <Number of Oracle system processes></code>.</p> <p>50 is acceptable for 10 users maximum.</p> <p>To define according to the rights granted by the Oracle license</p>
dml_locks	500

Parameter	Recommended value
open_cursors	<Number of concurrent users> * Max [30; <Maximum number of records updated in multiple selection>].
optimizer_mode	Use the RULE mode during initial import, then the FIRST_ROWS or ALL_ROWS mode as soon as the DB has been imported and statistics concerning tables and indexes calculated. (It is a function of the AssetCenter server.)
sort_area_size sort_area_retained_size	<p>On non-MTS servers, sort_area_size corresponds to the number of bytes, dedicated for sorts in memory.</p> <p>Sort_area_retained_size corresponds to the memory conserved at the end of the memory sort.</p> <p>As a basic guideline, you can set it to 65256, then increase it if necessary according to use.</p> <p>On an MTS server, the sort area is global for all multiplexed user connections, and sorts are done in the same space. Start with 65536*number of concurrent users, then increase it if necessary according to use.</p>

Verifications before creating the database with AssetCenter

- 1 Go to the Oracle server.
- 2 Using one of the Oracle utilities (Database Expander, Storage Manager, or DBA Studio, for example), make sure that there is enough free space in the tablespace to create an AssetCenter database. For example, if you install AssetCenter in a single tablespace for which the values of the **Storage** clause are set to **INITIAL 10K,NEXT 10K**: Anticipate about **150 MB** for a small database and **450 MB** for a database of 5,000 assets.
- 3 The **rollback segments** must be greatly extendable. We recommend a **rollback segment** for every 4 concurrent users. Each **rollback segment** must be extendable to about **25 MB**. The value of **initial** and **next** can vary between **256 KB** (with **5 MB** for **optimal**) and **10 MB** (with **20 MB** for **optimal**). This indication can be adapted according to your business rules and the size of the AssetCenter database.

- 4 Configure the Oracle instance so that it uses at least 30 MB of memory cache for the database.

Microsoft SQL Server

Preparing Microsoft SQL Server 7.0 or 2000

Use the Microsoft SQL Enterprise Manager utility:

Table 3.2. Important database and server parameters

Parameter	Default value	Recommended value
max server memory	OS memory	Keep some OS memory for the OS kernel and processes (at least 64 MB on Windows NT4, 128 MB with 250 concurrent users), and 128 to 256 MB with Windows 2000. The system must not swap.
auto create statistics	As set in model database at database creation.	Database setting, to be turned on for all AssetCenter Databases.
auto update statistics	As set in model database at database creation.	Database setting, to be turned on for all AssetCenter Databases.

To create the Microsoft SQL Server 7.0 database

- 1 Create a database with the name of your choice using a database file large enough for an AssetCenter database (at least **150 MB** of **DATA** for a small AssetCenter database, and **20 MB** of **LOG**).
- 2 Grant connection rights in the login detail.

 **Tip:**

In doubt, we recommend that you assign the role of **Database Owner** to the login of the user.

- 3 Display the database properties and the:
 - **Permissions** tab: Assign the necessary rights to the logins of your choice (at least **Create Table** and **Create Stored Procedures**).

- **Options** tab: Check the **Truncate Log on Checkpoint** option if you don't want to keep log files. If you want to keep them, you need to implement a procedure that will prevent you from skipping disk space.
- 4 Make sure the temporary space database, **tempdb**, is set to at least **20 MB**.

Configuring the database options

The following database options are recommended to avoid overloading the log file (**trunc. log on chkpt**) and to enable you to create the table-creation script with SQL Query Analyzer (**ANSI null default**):

- `Sp_dboption <dbname>, 'trunc. log on chkpt', true`
- `Sp_dboption <dbname>, 'ANSI null default', true`
- `Sp_dboption <dbname>, 'auto create statistics', true`
- `Sp_dboption <dbname>, 'auto update statistics', true`

Recovering the standard server configuration

The standard configuration of the server can be recovered by using the following SQL query (to be executed in SQL Query Analyzer, for example):

- `Sp_configure`
Recover the standard configuration (alloted memory, etc.).
- `Select @@version`
Recover the server version.
- `Sp_helpsort`
Recover the sort order physically used by the indexes.

Preparing client machines

Install the SQL Server client layers (SQL Server ODBC driver) on each client machine.

Sybase Adaptive Server

Table 3.3. Important server parameters

Parameter	Default value	Recommended value
memory	7,500	This setting determines the total size of the memory cache (the cache that stores data and procedures, in particular). This value is expressed in 2KB blocks. With 512 MB of RAM on the server, a 384 MB of cache is a good value (which yields a value of 192000 for this setting).
procedure cache	30	Cache memory area allocated for the storage of parsing and SQL queries. This setting is expressed in %. It should be reduced as the size of the cache memory increases. For a 384 KB cache memory, it should be defined around 3 to 10 percent.
maximum network packet size	512	8,192
default network packet size	512	512
additional netmem	0	Number of user connections * 3 * (maximum network packet size - default network packet size)
extent i/o buffers	0	(Memory; / 8 Mo +1) * 10
tempdb	2 MB	20 MB at the least; increase if necessary.

To create the database

- 1 Go to the Sybase Adaptive Server.
- 2 Launch Sybase Central, for example.
- 3 Reserve a **database device** large enough to create a Sybase database (at least **150 MB** for a small AssetCenter database).
- 4 Create a Sybase database with a **DATA** segment of at least **150 MB** (for a small database). After creating, select the **Truncate Log on Checkpoint** check box in the database options. If you use the **LOGs**,

reserve about **20 MB** for them, whether they are on a separate segment or added to a **DATA** segment.

- 5 Make sure the temporary space database, **tempdb**, is set to at least **20 MB**.
- 6 Configure Sybase so that it uses at least **30 MB** of memory cache. We recommend maximizing this value in relation to the memory available on the server. Performance will be greatly improved as long as the server does not swap.

Locking mode after creating database with AssetCenter

If you use the 11.9 version, we highly recommend that you take advantage of **Data Only Locking** and, in particular, **Row-Level Locking**, which reduces conflict problems. To perform this task, the database administrator must reconfigure all the AssetCenter tables after they have been created: (for each table: `alter table <table> lock datarows`).

If you notice that the performances associated with the queries on large tables (such as the Histories or Comments tables) are degenerating, you need to implement identified caches for these tables. Then, when you perform a query on these tables, data from the other tables won't be overwritten in the cache by the data from the large tables. This is sometimes an important factor in the regulation and conservation of your performances.

If you want your data access plans to be efficient, you need to perform statistics on the tables using AssetCenter Server or SQL scripts.

DB2 UDB

The information that follows should be considered after you configure the DBMS, but before you create the AssetCenter in the DBMS.

The following operations are required for AssetCenter to function correctly with DB2 UDB.

The following orders correspond to the Command Center syntax of DB2 UDB. You can adapt the following solutions if you want to:

- DOS session

- Command Center de DB2 UDB

Preparing the database on the server

We recommend that the database administrator configures the database to ensure high standards. The administrator needs to increase the **APPLHEAPSZ** and **APP_CTL_HEAP_SZ** parameters of the database using the following SQL statements (example taken from Windows NT 4):

- ```
CREATE DATABASE <Database name>
```
- ```
Update database configuration for <Database name> using
APPLHEAPSZ 2048
```
- ```
Update database configuration for <Database name> using
APP_CTL_HEAP_SZ 2048
```
- ```
Update database configuration for <Database name>; using
DBHEAP 4096
```
- ```
Update database configuration for <Database name>; using
LOGFILSIZ 500
```
- ```
Update database configuration for <Database name>; using
DFT_QUERYOPT 2
```
- ```
CATALOG TCPIP NODE <DB2_NT|SUN_NODE|AIX_NODE|Etc.> REMOTE
<Server name> SERVER 50000 REMOTE_INSTANCE DB2 SYSTEM
<Server name> OSTYPE <NT|SUN|AX|Etc.>
```
- ```
CATALOG DATABASE <Database name>; AT NODE
<DB2_NT|SUN_NODE|AIX_NODE|Etc.>
```

Preparing the application counter server

To replace the **SEQUENCE** service, which available in ORACLE but not in DB2 UDB, an application server for counters can be installed on DB2 UDB servers.

This is recommended if many users need to simultaneously access the database.

 **Note:**

Installation of this service is optional and requires your database administrator to be familiar with Java. If this service is not installed, you will need two physical connections of the AssetCenter client for each logical connection.

Preparing the clients

Start by installing DB2 client layers (DB2 Cli).

It is not necessary to create an ODBC source for the DB2 connection. However, if an ODBC source needs to be declared for another application, make sure it is a **system data source** and not a **user data source**. In the latter case, certain required optimizations are canceled by the **user data source** access type.

The following commands should be taken into account when creating your connection without using an ODBC datasource:

- CATALOG TCPIP NODE
- CATALOG DATABASE
- db2icrt

Example

Your database server is installed on a Windows NT server called **CALIFORNIUM** and the database itself is held in the **INST_1** instance (DB2 manages several concurrent instances) and is called **SAMPLE**. Here are the commands to execute before accessing this database (In the **DB2 Command Line Processor**):

```
CATALOG TCPIP NODE MyNode REMOTE CALIFORNIUM SERVER 50000  
REMOTE_INSTANCE INST_1 SYSTEM CALIFORNIUM OSTYPE NT
```

Then:

```
CATALOG DATABASE SAMPLE AS SAMPLE AT NODE MyNode
```

Then from a standard command prompt:

```
db2icrt MyNode
```

 **Tip:**

If your server is not configured in Windows NT 4, please consult your database administrator.

Runtime Sybase SQL Anywhere


The Sybase SQL Anywhere **runtime** supplied with AssetCenter is automatically offered when you install AssetCenter. If you intend to install separately a client-server version of Sybase SQL Anywhere, you do not have to install the runtime.

If you have installed the Sybase SQL Anywhere **runtime** and you want to create an empty database, use the following procedure:

- 1 Create a new folder on your hard drive (optional).
- 2 Copy the **Empty410.db** file, which can be found in the AssetCenter installation folder.

 **Tip:**

You can rename this file.

- 3 Launch AssetCenter.
- 4 Select **File/ Manage connections**.
- 5 Click **New**.
- 6 In the **Name** field, enter a name for this connection.
- 7 Select **Sybase SQL Anywhere** as your engine.
- 8 Click the  icon, which can be found on the far right of the **Data Source** field:
 - 1 Populate the **Database File** field: Indicate the path and the file name of the **Empty400.db** file that you have just copied. You can use the **Browse** button for this. Select **Custom** from among the possible values.
 - 2 Assign the value of your choice to the **Data Source Name** field.
 - 3 Click **Options**.
 - 4 Enter **Rtdsk50.exe** in the **Start Command** field.

- 5 Leave the **Database Switches** field empty and the **Autostop Database** option selected.
- 6 Click **OK** to confirm.
- 7 Click **OK** to close the ODBC configuration window.
- 9 In the **User** field, enter **itam**. In the **Password** field, enter **password**.
- 10 Click **Create**.
- 11 Click **Test** to test the connection.

Creating the DBMS login

All users of AssetCenter can share the same DBMS login. AssetCenter handles access rights via user profiles.

It is also possible to configure AssetCenter to use several DBMS logins having restricted database rights. This makes it possible to control access to the database using external tools. We do not document here the use of multiple logins.

Oracle example

Declare an Oracle login with at least **CONNECT**, **RESOURCE** and **CREATE SESSION** rights. If you have already created an AssetCenter database under Oracle, you need to create a new **schema** for the new database.

Example of Oracle SQL script to create the **schema**:

```
connect system@ORASERV;  
create user AssetCenter identified by <password> default  
tablespace <asset> temporary tablespace <temporary_data>  
profile default;  
grant connect, resource, create session to AssetCenter;
```

Changing the DBMS used with AssetCenter

If you decide to change the DBMS, you must first acquire the appropriate version of AssetCenter, then import the database using the Import module.

Destroying an AssetCenter database

For reasons of security, AssetCenter does not have a command, which lets you destroy a database. You must do this outside of AssetCenter.

Destroying an AssetCenter database is performed in several steps:

- 1 Destroying the database itself:
 - Use the administration tool of the DBMS.
 - Example for Sybase Adaptive Server: **SQL Server Manager**.
 - Example for Microsoft SQL Server 7.0: **Microsoft SQL Enterprise Manager**.
 - Example for Oracle: **Oracle User Manager**.
- 2 Delete connections defined at the level of AssetCenter on each client machine.
- 3 With Microsoft SQL Server: Delete ODBC connections on each client machine.

License rights and available functions

AssetCenter is a complete system for managing information technology with totally integrated functions, based on a shared repository:

- Portfolio
- Procurement
- Contracts
- Financials
- Cable and Circuit

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

 **Note:**

The list of available functions depends on the license rights you have acquired. These rights are stored in a license file (**license.cfg**) that Peregrine Systems provides and which you must insert into your database.

To learn more about how to install the license file into your database, refer to the "Selecting a license file" section of this chapter.



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

 **Note:**

Each user activates the functions important to them on their own client workstation using the **File/Activate modules** menu.

Declaring a connection to an empty shell

Here are the steps to follow to declare a connection to an empty shell:

- 1 Start AssetCenter.
- 2 Select **File/ Manage connections**.
- 3 Click .
- 4 In the **Connection** tab, populate the **Name** field, the DBMS, and the information that is specific to this DBMS.
- 5 Click  to create the connection.

 **Note:**

In Windows XP, you must connect with write rights in order to declare a connection.

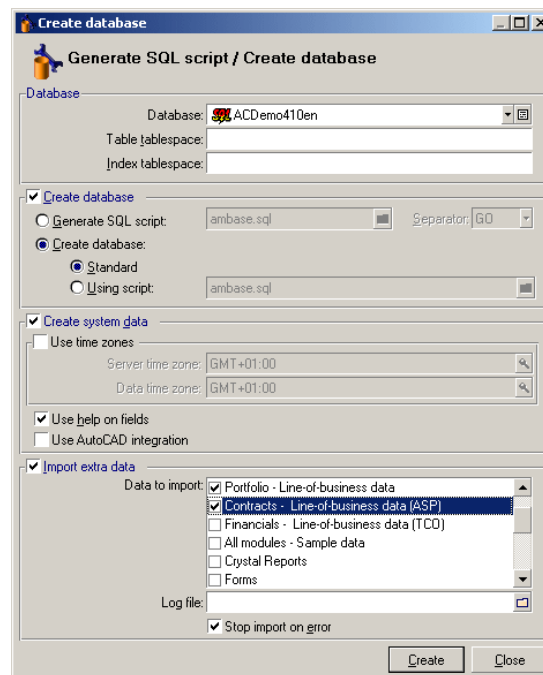
Creating a database with AssetCenter Database Administrator

AssetCenter Database Administrator enables you to create an AssetCenter database based on the open database's description file.

To create a database:

- Open the **gbbase.dbb** description file, located in the AssetCenter/Config folder.
- Open the database-creation window (**Action/ Create database**).

Figure 3.1. Database - Creation window



Options of the Database frame

These options enable you to select an existing AssetCenter connector and define the tablespaces for the tables and indexes. The tablespaces need to have already been created with the DBMS.

Options of the Create database frame

These options are only accessible if you check the **Create database** box. In this case, you can:

- Directly create the database if the **Create database** (the second one) and **Standard** options are selected.
- Generate an SQL script in order to create another database later on if the **Generate SQL script** option is selected. You must assign a name to this script (by clicking) and define an SQL separator using the drop-down list of the Separator field. (The classic separators are "/" for an Oracle database and "GO" for all other databases.)

 **Note:**

Because you can edit the drop-down list, you can add any other separator you wish provided that it is a valid one (such as ";", for example). Nothing prevents you, of course, of adding an invalid separator such as "<MySeparator>", but the database-creation script will not work.

- Create the database from an SQL database-creation script (which you can generate using the **Generate SQL script** option in the same frame) if the **Create database** and **Using script** options are selected. You must assign a name to the SQL script by clicking .

Options of the Create system data frame

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

- 1 Database description file stored in the system table (**System tables** table (SysBlob)).

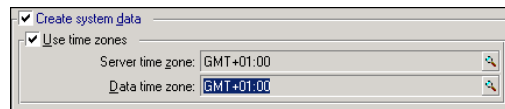
- 2 Password (empty by default) of the "Admin" login encrypted and stored in the "sysblob" table.
- 3 An "Admin" user.
- 4 Records with null identifiers (used to simulate outer joins) in each table.
- 5 Itemized lists and counters.

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

- Information about time zones stored in the **Options of the application** table (amOption).

 **Note:**

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



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.

If you think you will use the AssetCenter-AutoCAD integration functions, you must select **Use AutoCAD integration without FacilityCenter** in order to prepare your database for this use.

Options of the Import extra data frame

When you create your database, you can choose to automatically import certain **line-of business** data available with AssetCenter. To do this:

- 1 Select the **Import extra data** option.

- 2 Select the information that you want to import using the **Data to import** drop-down list. Depending on your needs, you can select all or part of the **line-of-business data**.

 **Note:**

You can also import the same information after the database is already created by using the AssetCenter import function.

- 3 You can then select the **Log file**. All the operations performed during this import, as well as any errors and warnings, will be written to this file.
- 4 In case of any problems, you can select the **Stop on error** option to stop the import of data when an any errors are encountered.

Selecting a license file


To activate the database access rights corresponding to your AssetCenter license, you must provide AssetCenter with a license file. This file is stored definitively in the database; it does not need to be declared at the level of client machines.

This license can be obtained from Peregrine Systems after providing certain information. This information needed by Peregrine Systems, together with the field used to enter the file, is located in the window that will be displayed just after launching the creation of the database.

Before calling Peregrine Systems:

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

 **Note:**

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 . You do not need to connect to a database to do this.

 **Warning:**

If AssetCenter Server runs on a computer with Windows 95, then you must install the NetBEUI protocol. This protocol allows AssetCenter Server to correctly identify the Mac address of the network card.


 **Note:**

We recommend that you run AssetCenter Server on a computer equipped with Windows NT.

The license determines:


- The number of authorized users.
- The maximum number of assets and main assets that you can create.
- The DBMS that may be used.
- The enabled functions.

Here are the steps you must follow:

- 1 Once you have declared the connection via the **File/ Manage connections** menu, click  to create the database.
 - 2 When the license-file dialog box is displayed, call Peregrine Systems.
 - 3 Tell the person at Peregrine Systems the value of the **MAC address for AssetCenter Server** field and the name of your company.
 - 4 Enter the license file provided by the person at Peregrine Systems.
-

 **Note:**

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 license file.

- 5 After the license is validated, AssetCenter displays the database-creation dialog box. If authorized to do so by your license agreement, you can declare time-zone information here. To do this, select the **Use time zones** option and populate the **Server time zone** and **Data time zone**.
- 6 To continue creating the database, click .

Entering the administrator password and "physically" creating the database

At the start of the database creation process, you must enter the administrator password. An administrator login and password are also defined when you create an empty shell using the database manager. AssetCenter allows you to change the administrator password before creating the database structure. The time taken to perform this operation varies according to the DBMS and the speed of the computer used. After the creation is complete, start AssetCenter Server and connect to the database to validate the database in order for users to be able to connect.

Modifying the license

If you change the network adapter or computer on which AssetCenter Server runs - or if your license expires - you must change the license file in your database. You do not have to be connected to the database to perform this modification.

Modifying the license using AssetCenter Database Administrator

- 1 Start AssetCenter Database Administrator.
- 2 Connect to the database whose license you want to modify.

- 3 Select the **Action/Edit license file** menu. Enter the new license file in the dialog box that appears.

Changing your DBMS

AssetCenter enables you to change the DBMS you use to run your AssetCenter database.

To change your DBMS, you must perform a **dump**, which means to export the structure and contents of your database into the old DBMS, then import it into the new one.

 **Note:**

Dump means the export of data into a text file. It does not mean the transfer of data in the strict sense of a database.

 **Warning:**

You must have identical versions of the databases. You cannot, for example, perform a dump from a AssetCenter 3.x database to a AssetCenter 4.1 database.

Follow these steps to perform the dump:

- 1 Create the dump of the old DBMS.
- 2 Create an empty shell for the new DBMS.
- 3 Declare the connection to the empty shell.
- 4 Import the dump into the empty shell.

 **Note:**

Steps 2 and 3 are not described in this section. For more information, consult this chapter, sections:

- Creating an empty AssetCenter database using your DBMS.
 - Declaring a connection to an empty shell
 - Creating a database with AssetCenter Database Administrator
-

Creating a dump

To create a dump:

- 1 Open your AssetCenter database (**Open/ Open existing database** menu in the AssetCenter Database Administrator).
 - 2 Select the **Action/Generate database dump** menu.
 - 3 Indicate the name of your dump and click **Save**.
AssetCenter generates the dump of the database.
-

 **Note:**

The dump of your database is composed of several files, which are automatically incremented: **xxx01.ar**, **xx02.ar**, **xxx03.ar**, etc. with **xxx** as the name of the dump.

Restoring a database from a dump

After the dump, the empty shell and its connection declaration have been created, you can restore the database from the dump:

- 1 Use the **Action/ Restore database from dump** menu.
- 2 In the dialog box that is displayed, select the empty shell you have created.
- 3 Select the first dump file (**xxx01.ar**) generated by AssetCenter Database Administrator.
AssetCenter imports the dump into your new database.

4 | Creating a database description file

CHAPTER

AssetCenter Database Administrator enables you to extract information from the database by controlling the nature of the extracted information and the format of extraction.

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

- The **database.txt** et **tables.txt** files: They contain the complete structure of the database. These files are located in the Infos sub-folder of the AssetCenter installation folder.

 Note:

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 AssetCenter Database Administrator with a connection to your database.

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

- An AssetCenter database description file (**.dbb** file extension) or a connection to an AssetCenter database.
 - A model (**.tpl** file extension) that describes which information to generate. We provide you with the standard models, but you can create your own if you wish. Sophisticated models enable you to create RTF or HTML formatted files.
-

 **Note:**

Among the standard models provided with AssetCenter, one of them, the **dbdict.tpl** file, enables you to export all the customization information from your database to a standard text-formatted file. This kind of information includes features, calculated fields, configuration scripts, etc. Used along with a "Source Control" tool, this description file can be very useful for keeping a trace of all customization modifications made to the database.

- The AssetCenter program.

This functionality is accessible via the **Action/Templates** menu, which is then divided into several sub-menus:

- **Select folder** allows you to specify the folder where AssetCenter Database Administrator will search for description templates. The software searches all the sub-folders in the selected folder.
 - **Refresh list** restarts the search of description files from the folder specified the last time you changed folders.
 - 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.
-

 **Note:**

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

Introduction

The internal structure of a database can be viewed as a collection of hierarchical objects: a database that contains tables, which themselves contain fields, links, indexes, etc.

Describing a database means browsing its structure and extracting from it the information that you need. The way in which AssetCenter Database Administrator extracts information (as much the contents as the form under which the contents are extracted) is described in template files. These files are little programs whose syntax is easily understandable if you have a little programming experience. This syntax is described in the following sections of this chapter.

Database description parameters

The following parameters are used to describe the database:

```
Instance DATABASE
Property P1-n
Collection TABLES as TABLE
Collection CALCFIELDS as CALCFIELDDESC
Collection FEATURES as FEATPDESC
Collection PARAMS as FEATPARAMDESC
Collection CLASSES as FEATCLASSDESC
Collection SCREENS as VIEW

Instance TABLE
Property P1-n
Collection FIELDS as FIELD
Collection LINKS as LINK
Collection INDEXES as INDEX
Collection RELEVANTSCRIPT as SCRIPT
Collection PROCESSES as BGPROC
Collection FEATURES as FEATPARAMDESC
Object O1-n as <nom de l'instance>

Instance FIELD
Property P1-n
Collection DEFVALDEPENDENCIES as DEFVALSCRIPT
```

```
Collection SYSENUMVALUES as SYSENUMVALUE
Object 01-n as <nom de l'instance>

Instance LINK
Property P1-n
Object 01-n as <nom de l'instance>

Instance INDEX
Property P1-n
Collection FIELDSINDEX as FIELD
Object 01-n as <nom de l'instance>

Instance SCRIPT
Property P1-n
Collection REFERENCEDFIELD as SCRIPTFIELD
Collection REFERENCEDSTORAGEFIELDS as STRING
Object 01-n as <nom de l'instance>

Instance FEATDESC
Collection PARAMS as FEATPARAMDESC
Object 01-n as <nom de l'instance>

Instance FEATPARAMDESC
Property P1-n
Object 01-n as <nom de l'instance>

Instance FEATCLASSDESC
Property P1-n
Object 01-n as <nom de l'instance>

Instance CALCFIELDDESC
Property P1-n
Object 01-n as <nom de l'instance>

Instance VIEW
Property P1-n
Collection PAGES as PAGE
Collection FIELDSINLISTCONFIG as PAGEITEM
Object 01-n as <nom de l'instance>

Instance PAGE
Property P1-n
Collection FIELDS as PAGEITEM
```

```

Object 01-n as <nom de l'instance>

Instance PAGEITEM
  Property P1-n
  Object 01-n as <nom de l'instance>

Instance SYSENUMVALUE
  Property P1-n
  Object 01-n as <nom de l'instance>

Instance STRING
  Property P1-n
  Object 01-n as <nom de l'instance>

Instance SCRIPTFIELD
  Property P1-n
  Object 01-n as <nom de l'instance>

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 the fields.

Each instance may be described with the following information:

- Property: a property of the instance.

For example:

```

Instance Table
  Property SqlName

```

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

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

For 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.

For 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 that define the information to extract, and how to process and present it.

These files must be in the following format:

- Type: text
- Character set: ANSI
- 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.

 **Note:**

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

For 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, and which are used to add comments to the template, must begin with the "\$" character and be followed by a space.

For example:

```
$ This is a comment line.
```

Including another template

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

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

For example:

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

Example of use: enables 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)]] [<filtering
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
```

For example:

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

You must respect the hierarchy between collections. Examples:

- 1 The "Fields" collection is dependent on the "Tables" collection.
- 2 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
```

This 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:

- 1 ASC: ascending alphabetical order.
- 2 DESC: descending alphabetical order.

For 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
objet>.]<Property>
...
$endfor
```

Note:

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

For 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.

For 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.

For 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

This is an alternative syntax for calling the value of the **Property** property.

Property must be in uppercase.

For example:

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

This produces the same result as:

```
$(PrimaryKey)
```

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

This creates a **ValueType** type global variable called **Property** for the template.

Property must be in uppercase.

Examples:

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

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

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

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

Exist(<strProperty> as String) as Integer

This tests for the presence of the **Property** global variable in the template.

For example:

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

This returns a numeric value of **1** if the property exists. Otherwise it returns **0**.

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

This defines an **ErrorCode** and an error message **Message** to be returned.

For example:

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

This produces an ASCII error message in the defined cases.

SetOutput(<strFile> as String) as String

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

Examples:

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

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

```
SetOutput("")
```

This displays the results on the screen.

CollectionCreate(<strName> as String) as Integer

This 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.

For 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

This performs an iteration on the collection previously defined using "CollectionCreate()".

For example:

```
CollectionNext()
```

The function returns "0" if the iteration is performed 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

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

For example:

```
strName=CollectionName()
```

CollectionIsFirst() as Integer

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

For example:

```
CollectionIsFirst()
```

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

CollectionIsLast() as Integer

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

For example:

```
CollectionIsLast()
```

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

CollectionCurrentIndex() as Integer

This 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.

For example:

```
Number=CollectionCurrentIndex()
```

CollectionCount() as Integer

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

For 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 such 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 <name of the function>({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.

 **Note:**

By default, the declaration of a variable's function is not verified. Thus, you could use a variable without declaring it, which would cause errors while executing your script. If you want to modify this behavior so that the variables' declarations are verified, you just need to add the following line to the Basic block of the function (right after the \$script marker):

```
Option Explicit
```

Examples:

```

Function 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.

For 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:

```
$<variable name> = <Basic formula>
```

Examples:

```
$A = 1
```

```
$Var = "text"
```

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

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

Information about certain database description parameters

This section includes information about the following description parameters:

- Database instance
- Table instance
- Field instance
- Link instance

- Index instance
- Script instance
- Instance FeatDesc
- Instance FeatParamDesc
- Instance FeatClassDesc
- Instance CalcFieldDesc
- Instance View
- Instance Page
- Instance PageItem
- Instance SysEnumValue
- Instance String
- Instance ScriptField
- Global variables

Database instance

Properties

Table 4.1. Database instance properties

Property name	Description	Connection required
LoginName	Name of the Login that you use to access the database.	Yes
LoginId	ID of the Login that you use to access the database.	Yes
TableCount	Total number of tables in the database.	No
Connected	This field may have two values: <ul style="list-style-type: none"> • 1: AssetCenter Database Administrator was executed in reference to a connection. • 0: AssetCenter Database Administrator was executed in reference to a database description file. 	No
Connection	Name of the AssetCenter connection used to access the database.	Yes
AppInfo	Information about AssetCenter.	No
AppVersion	AssetCenter version number.	No

Property name	Description	Connection required
AppBuild	AssetCenter compilation number.	No
AppLanguage	AssetCenter language.	No
DbbVersion	The version number of the database structure.	No

Table instance

Properties

Table 4.2. Table instance properties

Property name	Description	Connection required
Create	This property may have two values: <ul style="list-style-type: none"> • 1: The login has create access rights for the table. • 0: The login does not have create access rights for the table. 	Yes
Delete	This property may have two values: <ul style="list-style-type: none"> • 1: The login has delete rights for the table. • 0: The login does not have delete rights for the table. 	Yes
ComputeString	Table description string.	No
InternalName	Internal name. This information has no particular use.	No
Label	Label	No
Desc	Description	No
SqlName	SQL name	No
FieldCount	Total number of fields in the table.	No
LinkCount	Total number of links in the table.	No
IndexCount	Total number of indexes in the table.	No
IsFirst	Indicates whether the item is located at the start of the collection, given the filter and the sort order: <ul style="list-style-type: none"> • 0: no • 1: yes 	No

Property name	Description	Connection required
IsLast	Indicates whether the item is located at the end of the collection, given the filter and the sort order: <ul style="list-style-type: none"> • 0: no • 1: yes 	No
Count	Count of the browsed items in the collection, given the filter.	No
CurrentIndex	Position of the item in the collection, given the filter and the sort order.	No
System	Indicates if this object is a system object (cannot be modified): <ul style="list-style-type: none"> • 0: no • 1: yes 	No
HasFeatureValueTable	Indicates if this table has a table of feature values associated with it: <ul style="list-style-type: none"> • 0: no • 1: yes 	No
IsFeatureValueTable	Indicates if this table is a table of feature values: <ul style="list-style-type: none"> • 0: no • 1: yes 	No
HasMemoField	Indicates if this table has a Memo -type field. <ul style="list-style-type: none"> • 0: no • 1: yes 	No

Objects

Table 4.3. Table instance 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.
FVSourceTable as Table	Source table of feature values.
IsValidScript as Script	Validity script
RelevantScript as Script	Relevance script
Base as Database	Described database

Field instance

Properties

Table 4.4. Field instance properties

Property name	Description	Connection required
Update	This property may have two values: <ul style="list-style-type: none"> • 1: The login has update rights for the field. • 0: The login does not have update rights for the field. 	Yes
Write	This property may have two values: <ul style="list-style-type: none"> • 1: The login has create access rights for the field. • 0: The login does not have create access rights for the field. 	Yes
Read	This property may have two values: <ul style="list-style-type: none"> • 1: The login has read rights for the field. • 0: The login does not have read rights for the field. 	Yes
UserType	By default, the data entry and display format is the format for the "Type" property. The "UserType" property enables you to specify the data entry and display format when it is verified.	No
Type	Storage format	No
UserTypeFormat	Additional information about the "UserType" parameter.	No
Size	Maximum size of the values in the field, in number of characters.	No

Property name	Description	Connection required
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"> • 1: The field can never be modified by users. • 0: The field may be modified by a user, if the user has the required access rights. 	No
Historized	<p>This field may have two values:</p> <ul style="list-style-type: none"> • 1: History is recorded for the field. • 0: History is not recorded for the field. 	No
ForeignKey	<p>This field may have two values:</p> <ul style="list-style-type: none"> • 1: The field is a foreign key. • 0: The field is not a foreign key. 	No
PrimaryKey	<p>This field may have two values:</p> <ul style="list-style-type: none"> • 1: The field is a primary key. • 0: The field is not a primary key. 	No
InternalName	<p>Internal name</p> <p>This information has no particular use.</p>	No
Label	Label of the field, as it appears in detail screens, for example.	No
Desc	Description	No
SqlName	SQL name	No
LongHelpComment	Comments on the use of the field.	No
LongHelpSample	Examples of values that may be assigned to the field.	No
LongHelpWarning	Important information for the field.	No
LongHelpDesc	Description of the field.	No
LongHelpCommentNoHTMLTag	Remarks about the utility of the field (HTML tags excluded).	No
LongHelpSampleNoHTMLTag	Examples of values that the field can accept (HTML tags excluded).	No
LongHelpWarningNoHTMLTag	Important information about the field (HTML tags excluded).	No
LongHelpDescNoHTMLTag	Description of the field (HTML tags excluded).	No

Property name	Description	Connection required
System	Indicates if this object is a system object (cannot be modified): <ul style="list-style-type: none"> • 0: no • 1: yes 	No
EmptyOnDup	Indicates if the default value is reapplied in case of duplication. <ul style="list-style-type: none"> • 0: no • 1: yes 	No
FieldCase	Indicates the relative case behavior for the field. <ul style="list-style-type: none"> • 0: No modification to the field value. • 1: Value modified in upper-case characters. • S2: Value modified in lower-case characters. • 3: Value modified in smartcase. 	No
Positive	Indicates if the numeric-type field is positive: <ul style="list-style-type: none"> • 0: no • 1: yes 	No

Objects

Table 4.5. Field instance 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.
ReadOnlyScript as Script	Read-only calculation script for the field.
HistoryScript as Script	History calculation script for the field.
RelevantScript as Script	Relevance calculation script for the field.

Possible values for the "Type" property

Table 4.6. 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 in which all characters are allowed.
7	date+time	Date and time
9	blob	Variable-length binary field (or BLOB Binary large object) can stores images and forms, for example, without size restriction.
10	date	Date only (no time).
12	memo	Variable-length text field.

Possible values for the "UserType" property

Table 4.7. 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 that specifies the described field.

Link instance

Properties

Table 4.8. Link instance properties

Property name	Description	Connection required
Update	This property may have two values: <ul style="list-style-type: none"> • 1: The login has update rights for the link. • 0: The login does not have update rights for the link. 	Yes
Write	This property may have two values: <ul style="list-style-type: none"> • 1: The login has create access rights for the link. • 0: The login does not have create access rights for the link. 	Yes
Read	This property may have two values: <ul style="list-style-type: none"> • 1: The login has read rights for the link. • 0: The login does not have read rights for the link. 	Yes
Type	Type of the link.	No
UserType	Type of information managed by the link.	No

Property name	Description	Connection required
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"> • 1: The target table is pre-defined. • 0: The target table is pre-defined. 	No
Historized	This field may have two values: <ul style="list-style-type: none"> • 1: History is recorded for the field. • 0: History is not recorded for the field. 	No
Cardinality	Cardinality of the link.	No
InternalName	Internal name This information has no particular use.	No
Label	Label.	No
Desc	Description	No
SqlName	SQL name	No
System	Indicates if this object is a system object (cannot be modified): <ul style="list-style-type: none"> • 0: no • 1: yes 	No
LongHelpComment	Comments on the use of the field.	No
LongHelpSample	Examples of values that may be assigned to the field.	No
LongHelpWarning	Important information for the field.	No
LongHelpDesc	Description of the field.	No
LongHelpCommentNoHTMLTag	Remarks about the utility of the field (HTML tags excluded).	No
LongHelpSampleNoHTMLTag	Examples of values that the field can accept (HTML tags excluded).	No
LongHelpWarningNoHTMLTag	Important information about the field (HTML tags excluded).	No
LongHelpDescNoHTMLTag	Description of the field (HTML tags excluded).	No

Objects

Table 4.9. Link instance 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
HistoryScript as Script	History calculation script for the link.
RelevantScript as Script	Relevance calculation script for the link.

Possible values for the "Type" property

Table 4.10. 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

Table 4.11. Possible values for the "UserType" property

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

Index instance

Properties

Table 4.12. 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"> • 1: You can create several records where the group of fields in the index can have exactly the same value. • 0: You cannot create more than one record where the group of fields in the index has a given value.
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"> • 1: You can create several records having an index of NULL. • 0: You can only create one record having an index of NULL.
SInternalName	<p>Internal name</p> <p>This information has no particular use.</p>
Label	Label as it appears in the detail screens.
Desc	Description

Property name	Description
SqlName	SQL name
System	Indicates if this object is a system object (cannot be modified): <ul style="list-style-type: none"> • 0: no • 1: yes

Objects

Table 4.13. Index instance objects

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

Script instance

Properties

Table 4.14. Script instance properties

Property name	Description
CalcMode	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: <ul style="list-style-type: none"> • 0: no • 1: yes • 2: script

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

Objects

Table 4.15. Script instance 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.

Instance FeatDesc

 Note:

You need to connect to the database in order to use this instance.

Properties

Table 4.16. Properties of the FeatDesc instance

Property name	Description
System	Indicates if this object is a system object (cannot be modified): <ul style="list-style-type: none"> • 0: no • 1: yes
Label	Label of the feature.
Desc	Description
SQLName	SQL name
Unit	Unit of the feature.
Type	Entry type of the feature.
Enum	Itemized list used to populate the value of the feature. This property is only useful if the feature's Type is Itemized list .
MinValue	The minimum value accepted by the feature (numeric type only).
MaxValue	The maximum value accepted by the feature (numeric type only).
IsConsolidated	Indicates if the value is calculated by consolidation. <ul style="list-style-type: none"> • 0: no • 1: yes
HasClass	Indicates if the feature is attached to a feature class: <ul style="list-style-type: none"> • 0: no • 1: yes

Objects

Table 4.17. Objects of the FeatDesc instance

Object name	Description
Class as FeatClassDesc	The class of the feature.

Instance FeatParamDesc



Note:

You need to connect to the database in order to use this instance.

Properties

Table 4.18. Properties of the FeatParamDesc instance

Property name	Description
LongHelpComment	Comments on the use of the field.
LongHelpSample	Examples of values that may be assigned to the field.
LongHelpWarning	Important information for the field.
LongHelpDesc	Description of the field.
LongHelpCommentNoHTMLTag	Remarks about the utility of the field (HTML tags excluded).
LongHelpSampleNoHTMLTag	Examples of values that the field can accept (HTML tags excluded).
LongHelpWarningNoHTMLTag	Important information about the field (HTML tags excluded).
LongHelpDescNoHTMLTag	Description of the field (HTML tags excluded).
LinkFilter	The AQL condition of the filter (link type feature).
IsInherited	Indicates if the feature is inherited: <ul style="list-style-type: none"> • 0: no • 1: yes

Property name	Description
CreationHistorized	History line created when the main record is created.
Write	This property can take the values: <ul style="list-style-type: none"> • 0: The login as write rights over the feature. • 1: The login does not have write rights.
Read	This property can take the values: <ul style="list-style-type: none"> • 0: The login as read rights over the feature. • 1: The login does not have read rights.
Update	This property can take the values: <ul style="list-style-type: none"> • 0: The login as update rights over the feature. • 1: The login does not have update rights.

Objects

Table 4.19. Objects of the FeatParamDesc instance

Object name	Description
DefaultScript as Script	Default value script for the feature.
MandatoryScript as Script	Mandatory entry script for the feature.
AvailableScript as Script	Availability script for the feature.
HistoryScript as Script	History script for the feature.
ForceDisplayScript as Script	Default display script for the feature.
Table as Table	Table concerned by the feature parameter.
Feature as FeatDesc	Feature associated with the parameter configuration of the feature.
ValueField as Field	The field in which the feature's value is stored (fVal, ValString or dtVal).

Instance FeatClassDesc

 Note:

You need to connect to the database in order to use this instance.

Properties

Table 4.20. Properties of the FeatClassDesc instance

Property name	Description
FullName	Full name of the feature class.
Name	Name of the features class.
HasParent	Indicates if the feature class has a parent class: <ul style="list-style-type: none"> • 0: no • 1: yes

Objects

Table 4.21. Objects of the FeatParamDesc instance

Object name	Description
ParentClass as FeatClassDesc	Parent feature class.

Instance CalcFieldDesc

 Note:

You need to connect to the database in order to use this instance.

Properties

Table 4.22. Properties of the CalcFieldDesc instance

Property name	Description
Label	Label of the calculated field.
Desc	Description
SQLName	SQL name
Formula	Formula of the calculated field.
UserType	Result type returned by the calculated field.

Property name	Description
Type	Type of calculated field.
IsVisible	Indicates if the calculated field is visible by default: <ul style="list-style-type: none"> • 0: no • 1: yes

Objects

Table 4.23. Objects of the CalcFieldDesc instance

Object name	Description
Table as Table	Table with which the calculated field is associated.
Script as Script	Calculation script of the calculated field.

Instance View

Properties

Table 4.24. Properties of the View instance

Property name	Description
InternalName	Internal name This information has no particular use.
System	Indicates if this object is a system object (cannot be modified): <ul style="list-style-type: none"> • 0: no • 1: yes
Label	Label of the screen.
Desc	Description
SQLName	SQL name
FuncDomain	Functional domain associated with the screen.

Property name	Description
HasSystemPage	Indicates if this screen has information displayed constantly (a banner above tabs): <ul style="list-style-type: none"> • 0: no • 1: yes
HasNotebook	Indicates if the screen has tabs: <ul style="list-style-type: none"> • 0: no • 1: yes
ScreenFilter	System filter of the screen.
CaptionList	The caption of the screen's list.
CaptionDetail	The caption of the screen's detail.

Objects

Table 4.25. Objects of the View instance

Object name	Description
Table as Table	The table containing the screen.
SysPage as Page	The banner above the screen's tabs.

Instance Page

Properties

Table 4.26. Properties of the Page instance

Property name	Description
InternalName	Internal name This information has no particular use.
System	Indicates if this object is a system object (cannot be modified): <ul style="list-style-type: none"> • 0: no • 1: yes
Label	The label of the page.

Property name	Description
Desc	Description
SQLName	SQL name

Instance PageItem

Properties

Table 4.27. Properties of the PageItem instance

Property name	Description
IsVerticalSplit	Indicates if this object is a vertical separator for the page: <ul style="list-style-type: none"> • 0: no • 1: yes
IsHorizontalSplit	Indicates if this object is a horizontal separator for the page: <ul style="list-style-type: none"> • 0: no • 1: yes
FrameLabel	The label of the frame.
SQLName	SQL name.
IsField	Indicates if this object is a field on the page: <ul style="list-style-type: none"> • 0: no • 1: yes
IsLink	Indicates if this object is a link: <ul style="list-style-type: none"> • 0: no • 1: yes
IsFeature	Indicates if this object is a feature: <ul style="list-style-type: none"> • 0: no • 1: yes
IsCalcField	Indicates if this object is a calculated field: <ul style="list-style-type: none"> • 0: no • 1: yes

Property name	Description
BaseType	If the object is a field, this relates to the type of field.

Objects

Table 4.28. Objects of the PageItem instance

Object name	Description
Field As Field	The field contained in the page.
Link as Link	The link contained in the page.
FeatParam as FeatParamDesc	The features in the page.
CalcField as CalcFieldDesc	The calculated field in this page.

Instance SysEnumValue

Properties

Table 4.29. Properties of the SysEnumValue instance

Property name	Description
StringValue	The text value of the system itemized-list.
NumValue	The numeric value of the system itemized-list.

Instance String

Properties

Table 4.30. Properties of the String instance

Property name	Description
Value	The value of the character string.

Instance ScriptField

Properties

Table 4.31. Properties of the ScriptField instance

Property name	Description
Name	Name of the field referenced by the script.

Global variables

Table 4.32. Global variables

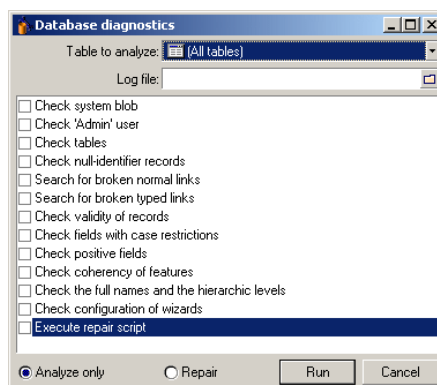
Property name	Description
Userlogin	Login you used when you connected to the database.
Time	Time at which AssetCenter Database Administrator was executed.
Date	Date on which AssetCenter Database Administrator was executed.
Dbb.Fullname	Full pathname of the database description file used.
Dbb.Shortname	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.

Property name	Description
Template.Name	Name of the database description template used (with the extension).
Template.Ext	Extension of the database description template used.

5 Diagnostic and repairs of a database

CHAPTER

The **Action/Diagnostics/Repair database** menu enables you to test the integrity of an existing AssetCenter database. In order to do this (and for the menu to even be available), you have to already be connected to the database in question.



Select **(No table)** to parse or repair the entire database.
You have two options for this:

- **Analyze only:** AssetCenter Database Administrator performs a simple diagnostic of the database, but does not perform any repairs in case of problems.
 - **Repair:** AssetCenter Database Administrator performs a diagnostic of the database and makes any necessary reparations.
-

 **Warning:**

The repair is conditioned by whether or not you activate the **Execute repair script**, in the same window. If this option is not activated, only the automatic reparation will take place. If you activated this option, however, the repair wizards will be launched after the automatic repairs.

AssetCenter Database Administrator checks and repairs the following items according to the options that you selected:

- Presence of the "sysblob" table (system data integrated with the database) in the database.
-

 **Note:**

Only the analysis of the "sysblob" table is possible.

- Existence of the "Admin" user (database administrator) in the table departments and employees.
 - Presence of all tables in an AssetCenter database.
 - Presence of records with null identifiers for all tables.
 - Search for broken links (simple or complex).
 - Validity of records.
-

 **Note:**

The repair is done using a wizard that is launched automatically.

- Broken fields (example: "user" instead of "User").
 - Positive fields.
-

 **Note:**

The repair is done using a wizard that is launched automatically.

- Coherency of features.
-

 Note:




The repair is performed automatically or by using a wizard according to the value of the feature (text, number, date).

- Full names and hierarchical levels: This mainly concerns the Employees table.
 - Configuration of wizards.
-

 Note:

Only the analysis is possible.

When you click **Run**, AssetCenter Database Administrator asks you to choose a log file to which will be written the operations and/or repairs performed during the diagnostic. During this diagnostic, the result of each test is indicated with an icon.

-  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.



6 | Updating your database structure

CHAPTER

AssetCenter Database Administrator enables you to convert a previous version of your database to a more recent version without having to use a migration script.

For further information about converting an AssetCenter database, refer to the **Migration** guide.

7 | Accessing the database via ODBC

CHAPTER

This chapter explains how to use external tools to access (read-only) the AssetCenter database via the ODBC driver developed for AssetCenter.

 Note:

The AssetCenter ODBC driver only supports read-only access to the database.

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. **Adbc.dll** is copied into the Windows "system32" folder.

 **Note:**

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.

 **Note:**

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:

Table 7.1. 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 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 DBMS in order to access the database.	You need to know the connection parameters required by the DBMS 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 enables you to view:

- Tables
- Standard fields
- Calculated fields
- Feature

All these objects are identified by their SQL name.

 **Note:**

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. You select the AssetCenter connection when you create and use the report. The standard connection dialog box under AssetCenter is used for this.

How to create your own ODBC connections

- 1 Launch the ODBC administrator.

- 2 Create a new connection by selecting the Peregrine AssetCenter Driver.
- 3 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 that 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.

8 Customizing the database

CHAPTER

AssetCenter Database Administrator enables you to perform different types of customizations.

- A customization of existing objects (tables, fields, links, screens, etc.).
- A customization by creating new objects.

Customizing existing objects

AssetCenter Database Administrator authorizes only limited customizations of the existing database objects. In order to avoid any problems while using AssetCenter, certain object values and objects are in read-only. Typically, you cannot modify the pages of an existing detail.

To customize an object in the database, edit the database description file (**gbbase.dbb**).

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

- Customization before creating the database.
- Customization after creating the database.

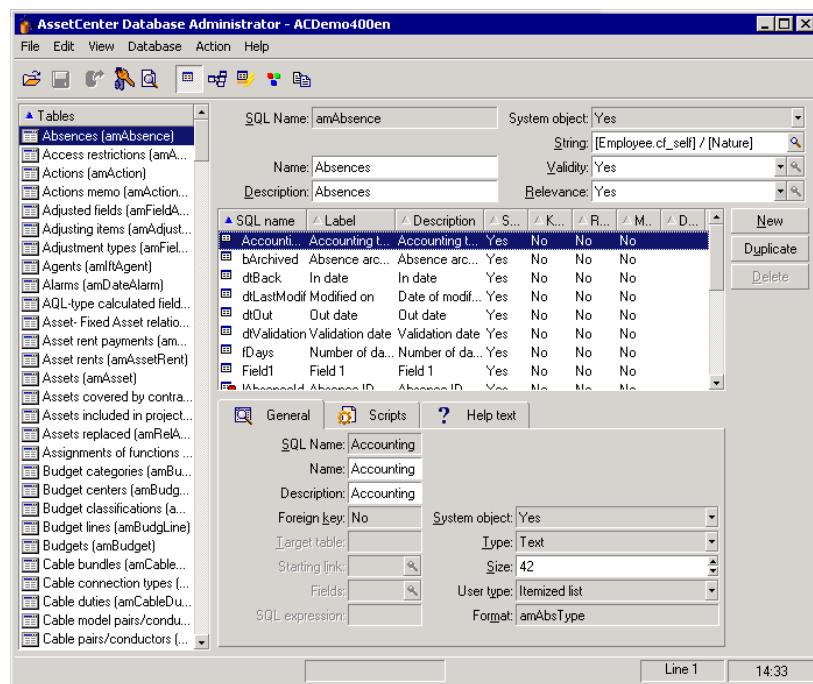
 **Note:**

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, indexes, screens, pages) of a table.

Figure 8.1. Administrator - main screen



Customizing tables

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 enables 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 enables 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 "Validity" field in the "amFeature" table:

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

 **Note:**

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

- The **Relevance** field applies to all the records of a database table. It enables you to delete records that are no longer pertinent (according to the criteria that you defined) and thus to rationalize the data in your database.
 - If this field is set to **Yes**, a table record is always relevant, whatever the value of its fields. This record will never be deleted.
 - If the field is set to **No**, a table record will never be relevant. It will systematically be deleted.
 - 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, there is little interest in conserving the records of the Comments table (amComment) that are empty (memComment field empty). This type of records is not relevant. For systematize their deletion, you can write the following Relevance script for the Comments table:

```
If [memComment]=" " Then
  RetVal= FALSE
Else
  RetVal = TRUE
End If
```


- The **Can contain features** option enables you to include features in a table. This option is irreversible.

Customizing objects


The second part of this customization pane enables you to read all the objects of a given 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 pane.


Note:


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.

Customizing a field, a link or an index

For this type of object, you can modify:

In the **General** tab

- The **Name** field.
- The **Description** field.
- The **Size** field enables 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:

 **Note:**

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).

 **Note:**

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 **Irrelevance** field conditions the display of the selected record in the detail of a list.
 - **Yes:** deletes the display of the object.
 - **No:** conserves the display of the object.
 - **Script:** indicates that a script is attached to the object to specify its display conditions.

```

If [bDepartment] = 1 Then
 RetVal = 1
Else
 RetVal = 0
End If

```

The selected object is not displayed if it is not relevant.

- The **Filter** field is not used in this version of AssetCenter.
- The **Formatting** field: enables 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.

 **Note:**

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

-
- The "Mandatory" field enables you to define conditions making a field mandatory.

 **Note:**

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

-
- The "Default" field specifies the field's default value. This value is automatically proposed by AssetCenter during the creation of a new record. The default values are defined using a Basic script.

 **Note:**

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**.

 **Note:**

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** shortcut menu) and can include up to

three sections. By default, these sections are entitled "Description", "Example" and "Precautions". The titles of these sections can be customized by modifying the labels of the links with SQL names "Comment", "Example" and "Warning" in the **Help on fields** table (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** pop-up menu command.

Customizing a detail

For this type of object, you can modify:

In the **General** tab


- The **Label** field.

In the **List/Detail** tab

- List/Detail proportions
- Title of the detail
- Title of the list
- System filter
- (Functional) Domain
- Columns in the list

Creating action buttons

The **Buttons** tab enables you to create action buttons that will be displayed in the detail. To create a button:

- Click .
- A line is added to the list of buttons exposed in the tab. Click on each of the line's cells to define the following button properties:
 - **Name:** internal name of the button. It enables you to identify the button in a unique manner.


- **Text** : the text appearing on the button.
- **Description**: label of the button as it is displayed in the graphical interface of AssetCenter.
- **Multiple selection**: Enables you to specify if the actions taken when you click the button can apply to several records.
- **No selection**: Enables you to specify if the actions taken when you click the button can be launched without any record having been selected.
- **Associated action**: Enables you to define the action executed when you click the button. Enter the SQL name of the action to execute.



```
<Type of action>:<SQL name of the action, of the view, etc. ...>
```

In this syntax, the type of action can take the following values:

- **A** for an action.
- **S** for a screen.
- **V** for a view.
- **F** for a form.
- **R** for a report.
- **Relevance**: enables you to specify whether the button is visible, invisible or whether it's visibility is conditioned by a script.

To delete an action button:

- 1 Select the button in the list displayed in the **Buttons** tab.
- 2 Click .
- 3 Click **Modify**.

To reorganize the order in which the buttons appear in the detail, use the  and  buttons.

Creating new objects

AssetCenter Database Administrator enables you to create new objects for the database.

 **Warning:**

We recommend that you work on an empty database. If you modify your working database, however, you must verify and repair the database (**Action/Diagnostics/ Repair database** menu) after having saved your changes.

Methods of creation

The following steps propose a method of creating new objects. These steps assume that you are creating the largest object: a table. Each step corresponds to a particular selection in this chapter.

- Create your table.
- Create the fields, links and index of your table.
- Create the details for your table.
- Create, if you want, action buttons present in the detail.
- Create the pages of your details.
- Add the pages to your details.
- Save your modifications.
- If this fails, propagate your modifications.

Creating a table

To create a new table:

- Select the **Database/Add a table** menu.
- AssetCenter Database Administrator displays a creation window.
- In this window, populate the standard fields associated with a table:
 - The **SQL name** field enables you to identify the new table in a unique manner and, notably, to reference it in a Basic script.

- The **Label** field contains the name of the table as it is shown in AssetCenter.
- The **Description** field.
- The **Primary key** field contains the SQL name of the field used as the table's primary key.
- If you want to associate features to your new table, select the **Add features** option. AssetCenter Database Administrator will automatically create the additional tables necessary to deal with the features.
- Click **Create**. AssetCenter creates the table as well as the field defined as the primary key for this table, and it puts you in position to edit the fields for this table.

Creating a field, link or index

To create a field:

- 1 Select the **Database/Add a field** menu.
- 2 AssetCenter Database Administrator displays a creation window.
- 3 In this window, populate the following fields:
 - **SQL name**
 - **Name**
 - **Description.**
 - **Type.**
- 4 Select the **Create an index for this field** option if you want to create an index.

After the field is created, you can no longer modify its size or format.

 **Note:**

Different types of fields:

- An itemized list type field must respect the following syntax:

```
Free text
```

The text entered (**Format** field) corresponds to the **Identifier** field in the detail of the itemized list.

- A **system-itemized list** type field must respect the following syntax:

```
To be validated|0|Validate|1|Refused|2
```

- A **duration** type field must respect the following syntax:

```
%Yl%Ml%Dld%H%N%S
```

- **Y, M, D, H, N, S** define the type of duration: respectively, year, month, day, hour, minute, second.
- **l** (long) defines whether the displayed duration unit is complete or not.
- **d** defines whether the duration is the default duration.

To create a link:

- 1 Select the **Database/Add a link** menu.
- 2 AssetCenter Database Administrator displays a creation window.
- 3 In this window, populate the following fields:
 - **Link type**
 - For the source table:
 - **SQL name**
 - **Name**
 - **Description.**
 - For the destination table:
 - **Table:** Indicate the destination table.
 - **SQL name**
 - **Name**
 - **Description.**

- 4 Select the **Create an index for this link** option if you want to create an index.

For more information about the different types of links, refer to this guide's chapter Standard database description files, section Description of links.

 **Note:**

Different types of links:

- The following are the types of 1-N links: Normal, Define, Own, OwnCopy, Own
 - The following are the types of 1-1 links: Normal1, OwnCopy1, Copy1, Own1, NeutDef1
 - The following are the types of N-N links: RelNN, NNCopy, CopyNN
-

To create an index:

- 1 Select the menu **Database/Add an index**.
- 2 Populate the different properties of each object. These properties are described in an exhaustive manner in the section **Customizing objects**, and are thus not detailed here.

Creating a hierarchical table

Creating a hierarchical table means creating a parent link in that table.

A parent link corresponds to a child link in the same table.

To create a parent link:

- 1 Select the **Database/Add a parent link** menu.
- 2 AssetCenter Database Administrator displays a creation window.
- 3 In this window, populate the following fields:
 - **SQL name**
 - **Name**
 - **Description.**
- 4 Populate the **Hierarchy** field according to the following syntax:

<Id of the parent link>,<name of the field to be used>
--

For example, for the Products table, the **Hierarchy** field of the **Parent** link is populated as follows:

```
lParentId, InternalRef
```

After these steps are completed, a **FullName** field is created. This is a unique identifier inside that table.


Creating a detail

A detail is a graphical view of the information stored in the objects of a table. The screens displayed in AssetCenter when you select a menu, click a toolbar icon or choose a view are all examples of a detail. Details are composed of several pages that are regrouped by tabs in the graphical interface of AssetCenter.

Note:

AssetCenter is provided with a set of standard details that are used to support the functions of the application. These details cannot be edited or modified. If you want to add a page to an existing detail, for example, you need to duplicate the detail, name it, then add the desired items.

To create a detail:

- 1 Select in the left-hand pane the table for which you want to create a new detail.
- 2 Select the View/Details menu or click the  button on the toolbar.
- 3 Select the **Database/Add a detail** menu or click **New**.
- 4 Populate the following fields in the **General** tab:
 - SQL name: SQL name of the detail. This name enables you to identify the detail in a unique manner and to reference it in scripts or queries.
 - Name: name of the detail.
- 5 Populate the following fields in the **List/Detail** tab:
 - Title of the list: defines the title of the detail as it appears in the list of screens (**Administration/ List of screens**, in AssetCenter).

- **Title of detail:** defines the title of the window displayed in AssetCenter. The **String** of the description is added to this title.
- **Domain:** functional domain of this detail. If the functional domain that you enter already exists in the database (created using the **Administration/Functional domains** menu), you will automatically have access to your screen in the functions pane under the corresponding functional domain.
- **List/detail proportions:** ratio between the detail and the list.
- **Columns of the list:** This field enables you to define the columns that will be displayed in a list. The syntax of this field is the following:

```
<SQL name of the field for the column 1>,<Proportional size of the column>,...
```


6 Click **Create**.

You just created the empty shell of your new screen. Now all you need to do is populate this screen by adding pages that contain fields or links, and possibly action buttons.

Creating action buttons

AssetCenter Database Administrator offers you to possibility of creating buttons in your details. These buttons enable you to trigger the execution of an action, the display of a screen, the printing of a report or form, or to open a view.

To create this button:

- 1 Select the detail to which you want to add a button.
- 2 Select the **Buttons** tab of the detail.
- 3 Click . A new item is created in the list of buttons for this tab.
- 4 You can directly edit the values for each property by clicking its value.
 - **Name:** SQL name of the button. This name enables you to identify the detail in a unique manner and to reference it in Basic scripts or queries.

- **Text:** contains the button's text.
- **Description:** label of the button as it is displayed in the graphical interface of AssetCenter when the cursor is over the button.
- **Multi-selection:** enables you to specify if the actions taken when you click the button can apply to several records.
- **Edit mode:** enables you to specify whether or not the button appears while saving an item.
- **No selection:** enables you to specify if the actions taken when you click the button can be launched without any record having been selected.
- **Associated action:** enables you to define the action that is executed when you click the button. The syntax of this field is the following:


```
<Type of action>:<SQL name of the action, of the view, etc.>
```







In this syntax, the type of action can take the following values:

- **A** for an action.
 - **S** for a screen.
 - **V** for a view.
 - **F** for a form.
 - **R** for a report.
- **Relevance:** enables you to specify the button's display parameters (yes, no or script).
- 5 Click **Modify** to validate your changes.

Creating a page

In AssetCenter, a page is graphically represented by a tab containing fields. To create this page:

- 1 Select in the left-hand pane the table for which you want to create a new page.
- 2 Select the **View/Pages** or click the  button on the toolbar.






- 3 Select the **Database/Add a page** menu or click **New**.
- 4 Populate the following fields in the **General** tab:
 - **SQL name:** SQL name of the page. This name enables you to identify the page in a unique manner and to reference it in scripts or queries.
 - **Name:** name of the page. This name appears as the name of the tab in the AssetCenter interface.
- 5 Click **Create**.
- 6 Click the **Contents** tab of the page's detail. You can now select the fields that will be available in the page that you are created. To do this:
 - Select in the list to the right (**Fields and links**) the field or the link that you want to add to the page, and click the  button to transfer it to the list on the left (**List of fields**). You can also double-click the field or link on the right to transfer it to the left.
 - If you want to delete a field from your page, select it in the list on the left and click the  button. You can also double-click the field on the left to perform this same operation.
 - The layout of this page and graphical organization of the fields on the page is automatically calculated by AssetCenter. You can, however, define the order of the fields' appearance on the page by ordering them in the left-hand pane using the buttons: , ,  and .
- 7 Click **Modify**.

Page layout

By default, the objects (fields or links) that you add to a page are displayed one after another in one single column. You can, however, define vertical and horizontal separators to modify this default behavior. These separators are defined by the following internal variables:

- **HSplitControl** for a horizontal separator.
- **VSplitControl** for a vertical separator.

To add a separator:

- 1 Open the **Contents** tab in the detail of a page.
- 2 Enter one of the two previously listed internal variables in the field above the list of fields and links.
- 3 Click 
- 4 The separator is transferred to the list of fields on the left. You can define the order in which these fields will appear on the page by using the , ,  and  buttons.

The following rules are applied when the page is displayed:







- All objects located between two **HSplitControl** variables will be part of the same horizontal block (one or more lines).
- The **VSplitControl** variable defines a separation in the column for the objects in a horizontal block.

For example, the contents of the General tab of an employee detail, as it is displayed by default in AssetCenter, can be defined as follows:

```
Location
HSplitControl
Phone
MobilePhone
VSplitControl
Fax
HomePhone
HSplitControl
EMail
Field1
Field2
HSplitControl
IDNo
dHire
dLeave
VSplitControl
CostCenter
BarCode
Field3
HSplitControl
memComment
```



Adding a page to a detail

To add a page to a detail:

- 1 Select the detail to which you want to add a page.
- 2 Select the **Pages** tab of the detail. You can now select the page(s) that you want to add to the detail. To do this:
 - Select in the list to the right (**Available pages**) the page(s) that you want to add to the detail, and click the  button to transfer them to the list on the left (**Pages in detail**). You can also double-click the pages on the right to transfer them to the left.
 - If you want to delete a page from your detail, select it in the list on the left and click the  button. You can also double-click the page on the left to perform this same operation.
 - To define the order of the tabs' appearance you need to order them in the left-hand pane using the buttons: , ,  and . The first page in the list will be the first tab displayed; the second page will be the second tab appearing, etc.

Saving modifications

AssetCenter Database Administrator uses the database's Smart update engine to save your customizations. To save these modifications:

- 1 Select the File/Save or click the  button on the toolbar.
- 2 AssetCenter Database Administrator displays a save window. You must define a **Log folder** in which the following information will be stored:
 - The operations performed while saving the modifications are stored in the **sdu.log**.
 - The SQL queries used to modify the database are stored in the **sdu.sql** file.
 - An XML file (**sdu.xml**) contains the structural differences between the initial database and the customized database.

- 3 Click **Update**. AssetCenter Database Administrator then updates your database by inserting the objects that you created.

Verifying your modifications

To check your modifications, you need to launch AssetCenter and connect to the customized database. Using the method previously defined, we created a new table. To access this new table:

- 1 Select the **Administration/ List of screens** menu.
- 2 Search for your detail in the list that appears on screen and click **OK**. Your new detail will appear.
- 3 You can also create a view of this new table if you want: Select the **Tools/Views/Create view from current window** menu.
- 4 Enter a **Name** for your view. This name will appear in the Functions and favorites panel.
- 5 Select a functional **Domain**. The **Name** of your view will appear under this functional domain the Functions and favorites panel.
- 6 You can now directly access your new detail.

Important note

In AssetCenter, all objects are defined by SQL names, in particular, the actions, view and screens. To open one of these objects from a menu or a function panel, AssetCenter uses the SQL name. If several objects, such as a view and a screen, have the same SQL name, AssetCenter tries to open the objects in the following order:

- View
- Screen

For example, let's say you create a detail for the Departments and Employees table. You name your detail amEmplDept. This is also the SQL name of the table, however. Now, when you select the **Portfolio/Departments and Employees** menu or click the **List of Employees** in the Functions and favorites panel, AssetCenter will open your detail and not the table.

9 Database options

CHAPTER

When you are connected to AssetCenter as the administrator, you can configure certain options inherent to AssetCenter and/or the database (via the **Administration/ Database options** menu).

Conventions

The edit screen (via the **Administration/ Database options** menu in AssetCenter) displays the assigned values for each option, and the minimum and maximum values allowed for certain options.

Values displayed in black cannot be edited.

The **Default value** column indicates the minimum value defined by Peregrine Systems

You can modify the **Current value** of the option, which is displayed in blue. 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).

The **Value used** column displays the default value, or the current value if it has been modified by the user.

 **Important:**

Any modifications you make in the **Current value** column are only taken into account when you restart AssetCenter.

Table 9.1. Database options

Section	Name of the option	Description	Example of values	Editable
Agent	Deactivate history during import	If this option is set to Yes , the history lines associated with records are not imported.	No	Yes
Procurement	Have the application server (aamsrv) create the items received	If this option is set to Yes , the ordered items are received in the amReceipt table but are not created in the portfolio. The corresponding records are stored in the intermediary amItemsReceived table and processed by the Create assets, consumptions, etc. corresponding to items received module of AssetCenter Server.	No	Yes
Wizards	Configuration script	This script contains Basic functions that can be called on in Wizard type actions.	N/A	Yes
Tips of the day	Strings	List of the "Tip of the day" tips displayed when AssetCenter is launched.	TIP2, "Shift+F9 starts the wizard debugger."	Yes
Authorization	Signature	Used by AssetCenter to verify that AssetCenter Server regularly connects to the database.	Updated by AssetCenter Server.	No
Authorization	Signature file	License file that you inserted in the database.		No

Section	Name of the option	Description	Example of values	Editable
Documents	Maximum size of documents that can be inserted into the database.	This size is expressed in bytes.	5 242 880	Yes
Features	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
Time zone	Server	Time zone of the server. The expected value is a numeric value specifying the difference with reference to GMT. An empty value means that no time zone is used. A null value (=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. An empty value means that no time zone is used. A null value (=0) defines the time zone as GMT.	-2 (time zone GMT-2)	Yes

Section	Name of the option	Description	Example of values	Editable
Event management	Expiration time for output events (in hours)	Output events are queues that can receive data and export it to external applications. The expiration time sets the value that determines when to purge these events.	336	Yes
Event management	Expiration time for input events (in hours)	Input events are queues that can receive data and import it from external applications. The expiration time sets the value that determines when to purge these events.	336	Yes
Full-text search	Enabled	Necessitates an initial configuration of the parameters by the database administrator.	No	Yes

10 Import

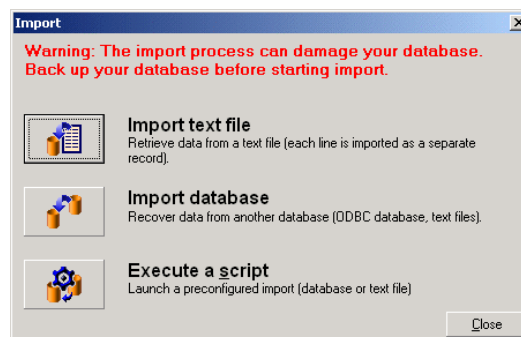
CHAPTER

This chapter explains how to import data with AssetCenter.

To import a single text file use the **File/ Import** menu, then the **Import text file** option.

To import several text files or a database use the **File/ Import** menu, **Import database** option.

Figure 10.1. Import - Selecting the type



Overview of importing data

An AssetCenter administrator can import data into the AssetCenter database from:

- A single text file.
- Several text files.
- A full database.

Importing data from a single text file

The text file is mapped to a table in the AssetCenter database.

The administrator maps each field in the text file to a database field, which can be located in the mapped table or in a linked table.

 **Note:**

The ISO character code used in the text file must be same as used in AssetCenter

Importing data from several text files

Each text file is mapped to a table in the AssetCenter database.

Each field in each text file is mapped to a database field, which can be located in the mapped table or in a linked table.

Importing data from a full database

You can import an ODBC database:

The administrator maps each table in the source database to a table in the target database.

The administrator maps each field in the source table is mapped to a field in a table of the target database. The target field can be located in the table that is mapped directly or in a linked table.

You can add or modify AssetCenter records. You cannot delete AssetCenter records.

All the information concerning the transfer can be saved as a script. You can use the script to import the data again without re-defining import settings.

The import module gives you the choice of several ways to handle errors and the possibility to write the results of operations carried out to a log file.

**Note:**

Protection of databases: Only an AssetCenter administrator can use the **File/Import** menu ("Admin" login or user with administration rights). This menu is disabled for other users, which protects access to the database.

**Note:**

Importing numeric data: Numeric fields must be formatted appropriately, regardless of the options defined in the Windows Control Panel: Only numeric characters without spaces are allowed, and the period "." is used as decimal separator. Numeric fields must be imported into numeric fields. This way, numeric data can be imported independently of the settings of the computer or computers used to perform the import.

Recommendations

Here are some recommendations before importing data into the AssetCenter database:

Default values for mandatory fields

The import module takes into account the mandatory nature of fields. If a record to be imported contains an empty field, and if this field is defined as mandatory in the target database, the import module rejects this record. To avoid records being rejected for this reason, we recommend assigning default values to mandatory fields in the target

database. If a value is given in the files to be imported this overrides the default value of the field.

Avoid using "Id" field as reconciliation keys

We advise against using the "Id" fields of tables as reconciliation keys if you want to reimport data you have exported. In effect, the corresponding identification numbers are not fixed and can be subject to modification. Use keys whose values are "changeless" such as the AssetTag of assets.

Backup the AssetCenter database

Since importing can make global modifications to your AssetCenter database, we recommend performing a backup before launching the import.

Avoid simultaneous access to the AssetCenter database

We advise against performing simultaneous imports on different machines or using AssetCenter on another machine during an import.

Constraints to be respected when importing data into a field

When importing data into the AssetCenter database, the import module verifies that the data is compatible with the structure of the database. If certain fields or links of data items to be imported are incompatible with the structure of the database, the outcome will be one of the following cases:

- The record is rejected as a whole.
- The value is truncated.
- The field is left as is.

You should make sure that the data you want to import is as compatible as possible to the structure of the database to avoid rejections.

Two possibilities are worth considering:

- You are importing a structured database:
 - If the source and target fields are of the same type (source-date and target-date for example), or if the fields are compatible (source-date+time and target-date for example), there are no particular constraints to be respected.
 - If the source field is a "text" type field and the target field is of a more specific type, you must respect the constraints of the target field.
- You are importing data from a text file:
 - You must respect the constraints of the target field.

According to the value of the "UserType" property of the target field

If the value is:	Then:
Default	You must respect the format defined by the "Type" property.
Number or Money	The source field must be a number. If the source field originates from a structured database and the field type is "number" or "money" then there is no constraint. If the source field is a "text" type field, you must present a number, use the period "." as a decimal separator and not use a separator for thousands.
Yes/No	The source field must display "1" (for "Yes"), or "0" (for "No").

If the value is:	Then:
Date	<p>If the source field originates from a structured database and the field type is "Date" or "Date+Time", then there is no constraint.</p> <p>If the source field is a "text" type field, you must respect the following constraints:</p> <ul style="list-style-type: none"> • The date format (order of year, month and day) must be the same for all records. Specify this format when importing. • You must systematically enter a day, month and year. • Use the same separator (of your choice) to separate days, months and years for all records. Specify this separator when importing.
System itemized list	<p>You must present one of the values of the itemized list only. Otherwise, the line will not be imported. You can identify the item from the itemized list by its clear value or as its number as stored in the database.</p> <p>Example</p> <p>If the itemized list is "Yes 1 No 0", importing "Yes" or "1" will give the same results.</p> <p>If you present an empty value, the import module attributes the value "0" to the field.</p> <p>We recommend presenting the stored numeric values, since these are more stable than their text equivalents from one version of AssetCenter to another and independent of the language version used.</p>
Custom itemized list	<p>The value of the field is selected in an itemized list, which can be modified by the user. You can therefore present the import module with one of the values of the itemized list. If, in addition, the itemized list is "open", you can present any value. This value will be added to the itemized list.</p>
Percentage	<p>Imported values must be the percentage value with or without the percentage sign "%" (for example: "10" or "10%").</p>
Time span	<p>You must respect the constraints defined by the "UserType" and "UserTypeFormat" properties with UserType is "Time span".</p>
Table or field SQL name	<p>You can import any alpha-numeric value. But if this value does not correspond to a valid SQL name of a field or table, you run the risk of corrupting the database.</p>

According to the value of the "type" property of the target field

You must respect the constraints linked to this property if the "UserType" property is set to "Default".

Particular case:

Special case: If the value is:	Then:
Date+Time	<p>If the source field originates from a structured database and the field type is "Date" or "Date+Time", then there is no constraint.</p> <p>If the source field is a "text" type field, you must respect the constraints defined by the "UserType" property when the UserType property is "Date" or "Date+Time".</p>

According to the value of other properties of the target field

If this property:	Is set to:	Then:
MandatoryType	Yes	If the source field is empty, the records that the import module should have added or modified are left untouched.
Size	Is populated	Values from the source field that are too long are truncated on import.
ReadOnly	Yes	It is not possible to import a value to a field with this property.

"dtLastModif" field (SQL name)

This field is updated when you modify or create records in the AssetCenter database via the user interface or via an import. It indicates the date of modification or creation of the record.

If you import a value to this field, this will overwrite the real import date.

Importing text files or an ODBC database

This section explains how to import one or more data files or an ODBC database.

To import a single text file: Use the **File/ Import** menu, then the Import text file option.

To import a set of text files: Use the **File/ Import** menu, then the Import database option, and finally the Text tab.

To import an ODBC database: Use the **File/ Import** menu, then the Import database option, and finally the ODBC tab

Before importing text files

Start by preparing the files containing the data to be imported. They must meet the following conditions:

- Each column represents a field.
- Each line represents a record.
- You may optionally include the field names at the start of the file. If you do not include them in the file, you may define them when you import the data.
- At the start of the file, before the optional line containing the field names, you can include up to 99 lines of comments that will not be imported: This can be done by populating the "First import line" field. These comment lines do not have to start with a special character.
- You may create the file using the OEM (DOS), UTF-8, UNICODE or Latin 1 character set.
- Columns can have either a fixed width or a variable width. In this case you select a character to act as a separator.
- Field contents may be delimited by any characters you choose.
- Field types may be numeric, character, or date.

- Imported values must follow the constraints related to the AssetCenter database structure (data-entry format, field type, link type, index type.)
- All characters are allowed except for those you use to delimit the text. There is no way to include the delimiter character in a text string.
- Date, date+time and duration data are subject to the same constraints as if they were directly entered into the program.
- We recommend that you build one text file for each main target table.

If your source data resides in a database whose engine is not supported by the import module, you must extract the data to text files and import them afterwards. If the database is recognized by AssetCenter you can import the information directly from the database using the **File/ Import** menu, then the **Import database** option.

Step 1: Select the text files or ODBC database to be imported

Importing a single text file

- 1 Select the **File/ Import** menu.
- 2 Select the **Import text file** option.
- 3 AssetCenter asks you to indicate the location of the structured text file containing the data to be imported.

Importing a set of text files


To select the text files you have prepared:

- 1 Select the **File/ Import** menu, **Import database** option, **Text** tab.
- 2 Click .
- 3 Once in the import module, select the **File/ Add file** menu. Indicate the names of the text files to be added.

Importing an ODBC database

Use the **File/ Import** menu and choose the option marked **Import text file** to select the source database.

Using the **ODBC** tab you can import an ODBC database.

- 1 Identify the data source, the user and the password. The  icon lets you directly create an ODBC data source, without having to go through the Windows Control Panel then the ODBC Data Source Administrator.
- 2 Click .

Step 2: Define the parsing of files or tables to be imported

Important:

When the file contains fixed-width fields, this step involves only one window. Two windows are necessary when the file contains separators between fields.

When importing a single text file

Once you have selected the text file in step 1, AssetCenter automatically asks you how it should be parsed.

When importing a set of text files

Once you have selected the text file in step 1, AssetCenter automatically asks you how it should be parsed. You may perform this step later by clicking , or populate this information at once.

There are different ways to recall this window when you are in the main screen of the import module:

- Double-click the source file.
- Or use the **Edit/ Properties** menu after having selected the source file.

The list of text files can be found in the "Source tables" column in the main screen of the import module.

When importing an ODBC database

Once you have opened the database in step 1, you can display the description of each table from the import module's main screen:

- Double-click the source table.
- Or use the **Edit/ Properties** menu after having selected the source table.

The list of source tables can be found in the "Source tables" column in the main screen of the import module.

First screen

Character coding

Indicate if the text uses the ANSI, OEM(DOS), UTF-8, UNICODE or Latin 1 character set.

First import line

Enter the line number containing the first line of data to be imported. Lines before this number are skipped by the import module.

If your document contains a line with field names and if this line is located just before the first data line, enter the number of the line with field names.

AssetCenter can skip up to 99 lines at the start of the file.

First imported line contains column names

Select this check box if the first line to be imported contains the field names (the column names). This way you do not have to enter the column names yourself.

If your file does not contain column names, you can define them in the following steps.

By separators

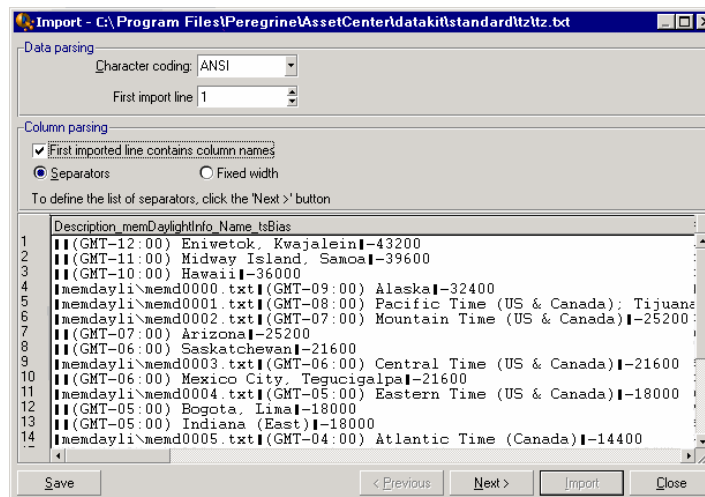
Check this check box if the field values are separated by a particular character.

Specify this character in the following screen.

Fixed width

Check this check box if all the values of each field are the same length. AssetCenter automatically displays column boundaries.

- To move a column boundary, use the mouse to select the boundary in the data area and drag it to the desired position. You cannot select the boundary in the title area.
- To remove a column boundary, use the mouse to select the boundary and drag it outside the table.
- To add a new column boundary, click with the mouse in the data area at the position where the separator should be inserted.




 **Note:**

The frame containing the records displays a partial preview of the file to be imported. A maximum of 25 lines are shown.

Possible second screen

This second screen is shown if in the first screen:

- 1 You specify that the field values are separated by a separator characters.
- 2 You click .

Column separators

Indicate the character used to separate two successive field values. We recommend using the semi-colon character ";" as separator.

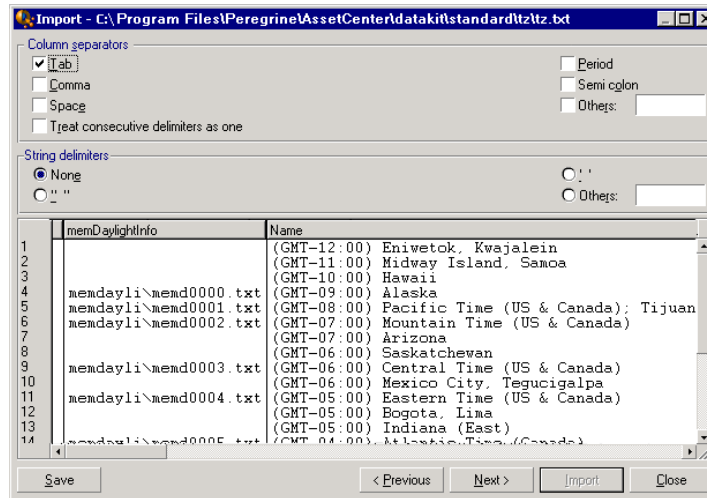
If you check the **Treat consecutive delimiters as one** option, AssetCenter will treat identical consecutive delimiters as one and will not create empty columns. If you want to create an empty column, uncheck this box use two consecutive delimiters in your text file.

String delimiters

If you use character to delimit text, indicate the character that you use. AssetCenter will remove these extra characters if they are present, before transferring the field to the database.


If, between two delimiters, AssetCenter encounters a column separator, it is considered as text. Specifying a string delimiter does not oblige you to use it systematically around all values. On the other hand, if you place a string delimiter before the start of a string this must be balanced by a delimiter at the end of the string.

You cannot import a string delimiter as a value.





Step 3: Describe the fields to be imported

When importing a single text file

Once you have parsed the text file as explained in step 2, click the  button to display the screen for describing the fields in the file.

When importing a set of text files

Once you have parsed the text file as explained in step 2, click the  button to display the screen for describing the fields in the file.

You can display this screen from the import module's main window by double-clicking the source file or by using the **Edit/ Properties** menu after selecting the file and clicking the  button once or twice, according to the "column parsing" selected in the previous step.

When importing an ODBC database

After selecting the database in step 1, you can display the tables' description from the import module's main screen: Either double-click

the source table or use the **Edit/ Properties** menu after having selected the source table. (The "Source tables" column in the main screen of the import module displays the list of source tables.)

Select the column to be configured by clicking anywhere in the column.

Number

The number of the selected column is displayed here.

If you did not select the **First imported line contains column names** option in step 2, you can select the column number directly, instead of clicking the table.

Name

The column name (or field name) appears here.

- 1 If you selected the **First imported line contains column names** option in step 2, you cannot modify the column name.
- 2 If you did not select that option, either leave the default name or modify it. This name makes it easy to identify columns during later steps.

Type

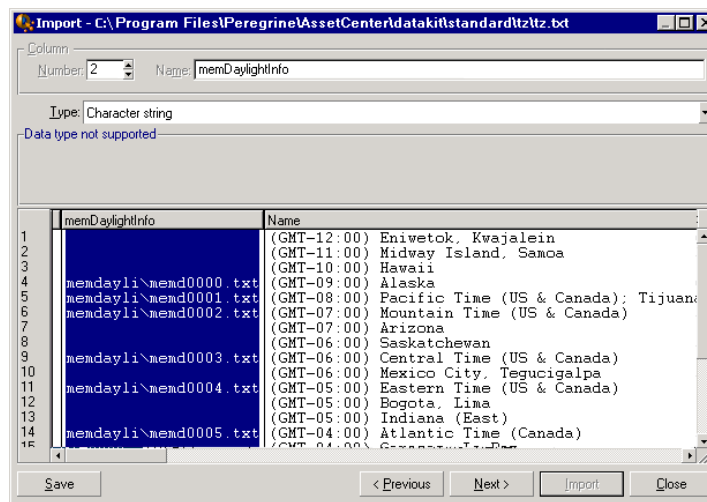
Enter the field type to be imported. AssetCenter lets you choose from the following possibilities:

- **Number:** All characters must be numbers. If AssetCenter finds other types of characters, this field's value is set to "0".
- **Character string:** All characters are allowed except those you used as text delimiters.
- **Date:** Only the date formats defined in the "Date format" frame are accepted. If AssetCenter finds other formats when importing, the field's value is set to null.

Date format

If you indicate that the field contains a "Date", AssetCenter asks you for the separator character between the day, month and year, as well as the order in which they appear.

Besides these two parameters, your dates may use all the possibilities available for entering dates in AssetCenter.




Note:

The table shows up to 25 lines of the file to be imported.

Step 4: Map source fields to target fields in the AssetCenter database

Importing a single text file

Once you have selected the source fields as described in step 3, click the  button to display the screen for mapping fields in the text file to AssetCenter database fields.

- 1 Start by mapping the text file to a table in the AssetCenter database using the "Target table" field.
- 2 Then map the fields in the text file to be imported (displayed in the "Target table" table) to a field in the AssetCenter database. (The fields in the target table and its linked tables are displayed in the list on the right under the "Target table" field.)

Importing a set of text files

- 1 Once you have selected the source fields as described in step 3, click the button to return to the import module's main window.
- 2 Map each text file to a target table.
- 3 Next, for each (text file, target table) pair in the table on the right, map each text field to an AssetCenter table: Either double-click the pair, or select it and use the **Edit/ Properties** menu.

Importing an ODBC database

- 1 Once you have selected the source fields as described in step 3, click the button to return to the import module's main window.
- 2 Map each text file to a target table.
- 3 Next, for each (text file, target table) pair in the table on the right, map the text fields to an AssetCenter database field: Either double-click the pair, or select it and use the **Edit/ Properties** menu.


Step 5: Map each text file or source table to a target table


 **Note:**

This section is relevant when importing a set of text files or an ODBC database.

Map each text file or source table to a target table

Click the file or the source table ("Source tables" column) and on the corresponding target table (in the "Target tables" column). Then do one of the following:

- Either, use the **Edit/ Map** menu.
- Or, click the  icon.
- Or, use the **Edit/ Map** menu: AssetCenter automatically maps the files or tables that have exactly the same name. This is achieved using the technical name of the field.

Use the **Edit/ Unmap** menu or the  icon to unmap a file or a source table from a target table.

Display the AssetCenter database structure

Double-click the target table in the "Target tables" column, or select it with the mouse and use the **Edit/ Properties** menu. AssetCenter displays the list of fields, their type and length.

Step 6: Map the fields to be imported to fields in the AssetCenter database

Source fields

This part of the screen displays the names you assigned to columns in step 3 (for text files) or the short-field description (for AssetCenter database).

Target table

When importing a single text file




Select the target table that will receive the data. AssetCenter displays the table's structure (fields in the table or in linked tables).

When importing a set of text files or an ODBC database

AssetCenter displays the structure of the target table mapped to the file or the source table (fields in the table or in linked tables).




Map source fields to target fields

There are several possibilities available to you:

- You can use the mouse to drag a "source field" to a "target field" to map them.
- You can select a "source field", select a "target field", then click the  icon to map them.
- The  icon lets you unmap a source field from a target field after having clicked the (source field, target field) pair.
- The  icon lets you to automatically map the source fields and target fields that share the exact same name. This is achieved using the technical name of the field.



Add additional calculated fields in the source file

AssetCenter allows you to add additional fields to your source file. These fields are not saved; just placed in memory.

Use the ,  and  icons to add, delete or display these additional fields.

Selecting keys

You may select one or more target fields to create record identification keys. An identifier key lets you identify a record in a table. If you select several keys, all these keys will let you identify the records.

Select the (source field, target field) pairs to qualify and click the  button to declare them as "keys". If this button is active, it appears as if it were pressed in and is lighter; the small icon to the left of the target field or link also looks like .

AssetCenter imports lines from the source file one by one, proceeding as follows:




- If there is a record in the database whose keys have exactly the same values, AssetCenter modifies the record according to the information contained in the text file.
- If several records exist with the same set of keys, the program stops at the first record it finds and ignores the others. It is therefore up to you to select the appropriate keys.
- If no records exist that match the keys, AssetCenter creates a new record in the database.

 **Note:**


We advise against using the "Id" fields of tables as reconciliation keys if you want to reimport data you have exported. In effect, the corresponding identification numbers are not fixed and can be subject to modification. Use keys whose values are "changeless" such as the AssetTag of assets.

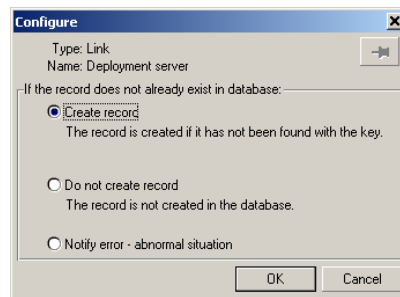
Configuring the creation of linked records

Let's say you want to import a file containing data that will be distributed to several different tables. This might be a file containing employee records and their corresponding portfolio-item records, for example. You would first select a main target table: the Employees table in our case. Then you would use links to indicate where the data should be distributed to other tables: the Portfolio items table in our case.

AssetCenter lets you configure how records will be created in the linked table if the record does not exist before the import process. Use the  icon to perform this configuration. This icon can only be used on links (not on the fields of linked tables). Links are represented by the  and  icons.

To display the configuration screen:

- 1 Map the field to be imported to the field in the linked table.
- 2 Click the corresponding link.
- 3 Click the  icon.



If the record is found using the specified identification keys, the program modifies the information in the record, if appropriate.

Create the record

The record is created if it was not found in the database using the specified identification keys.

Do not create the record

Records are not created, even if they are not found in the database using the specified identification keys.

Signal error - abnormal situation

AssetCenter generates an error message if the record is not found in the database using the specified identification keys.

Take into account only records that are already linked (📌)

The import module only considers records that are already linked to the main record if you attach the 📌 icon (pushpin) to the link.

Example

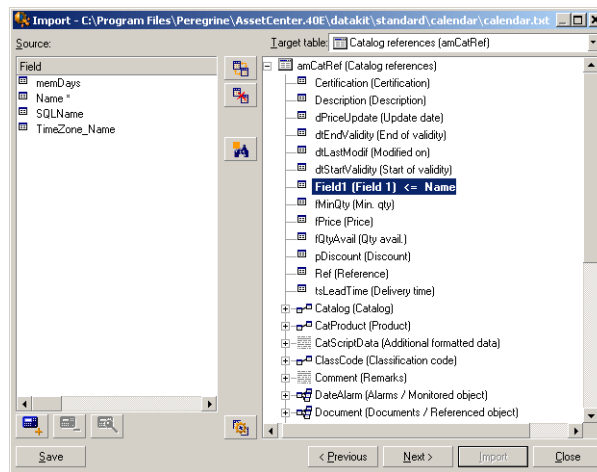
You are importing a list of employees and the portfolio items they use. The Employees and Departments table is the target table. You attach a pushpin to the link with Portfolio items table. For each portfolio item associated with an employee in the source file, the import module only takes into account the portfolio items that they are already using (**Portfolio** tab in employee detail).

In the case of "Own" type links, the pushpin is automatically placed and cannot be removed. "Own" type links are links for which the linked records are automatically deleted if the main record is deleted. The employees-training link is an example of this type of link: If you delete an employee, you will also delete all the training items linked to them. The effect of the pushpin depends on the types of links:

- If the target table is the Portfolio items table, and you attach a pushpin to the "user" link, the import module only looks for users linked to the portfolio items. Since there is only one user for a given portfolio item, it is thus possible to modify or create the user of a portfolio item without having to identify this user by a key. This is also very useful for modifying the value of a feature of a given portfolio item.
- If the target table is the Departments and Employees table, and you attach a pushpin to the "portfolio items" link, the import module only looks for the portfolio items linked to the employee. In this case, to modify or create the portfolio items of a user, you do have to have the appropriate identification keys, but they do not have to be as specific as they would be when not using the pushpin.

Note:

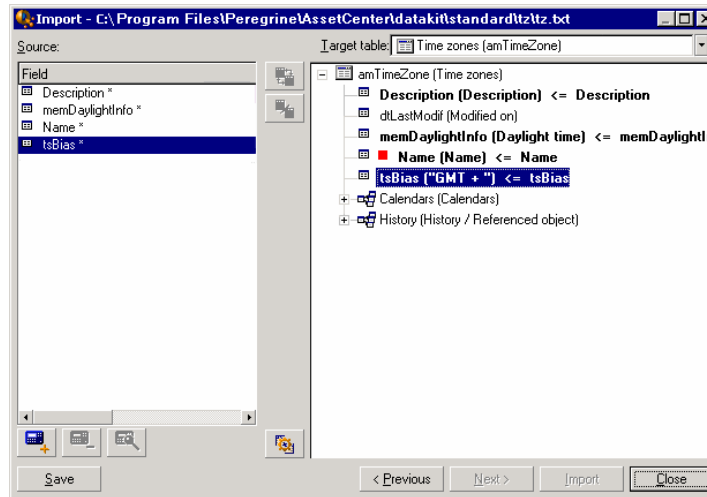
AssetCenter uses three kinds of links between records: n links: For example, a portfolio item can only be linked to one location, but a location can be linked to several portfolio items. 1 links: For example, a portfolio item can only be linked to one remark and a remark can only be linked to one portfolio item. n-n links: For example, a catalog can be linked to several distributor companies; a company can be linked to several suppliers.



Symbols used in the tree structure of target tables

- Indicates that the name following the symbol is a table (large symbol) or a field (small symbol).
- Indicates that the name following the symbol is a table linked to its parent table. You can only choose one record in the linked table from a record detail in the parent table. This type of link represents a field whose value can be entered using a "selection window" or a "drop down list".
- The name following the symbol is a table linked to its parent. You can choose several records in the linked table from a record detail in the parent table. This type of link represents a list of records located in a tab of the detail window of the parent table.

- This symbol is called the "pushpin". It can only be attached to a link to a table containing a field to which the imported field will be mapped. When a pushpin is set, the import module only looks among records linked to the record to which it is "pinned". The presence of the pushpin is determined by the options displayed using the 📌 icon.



Step 7: Add additional calculated fields in the source file

AssetCenter allows you to add additional fields to your source file. These fields are not saved; just placed in memory.

Use the 📌, 🗑️, and 🔍 icons to add, delete or display these additional fields.

Name

Give a name to this new field.

Formula field type

Enter the way the new field is to be calculated. The aspect of the screen changes according to the type you choose.

Concatenate

This mode lets you combine several fields in the source file. Select these fields one by one. You can separate these fields with the characters of your choice. Simply surround the characters with double quotes ".

Example: Field1" and "Field2.

Fixed extraction

This mode lets you extract part of a text field:

- 1 Select the source field (called "Main field").
- 2 Enter the **Number of characters to ignore**. AssetCenter will skip these characters.
- 3 Enter the **Number of characters to use**: AssetCenter will keep this number of characters after having skipped the "Characters to ignore".
- 4 The **Start extraction from end of field** check box tells AssetCenter to skip the **Number of characters to ignore** starting from the end of the field and keeps the **Number of characters to use** starting after the skipped characters and working backwards.

Example:

- 1 **Number of characters to ignore**: 3
- 2 **Number of characters to use**: 5
- 3 Value of field in source file: "REFIMP05A18500"
- 4 Value imported into the database: "IMP05" if the **Start extraction from end of field** check box was not checked, or "05A18" if it was checked.

Delimited extraction

This mode lets you extract part of a field in the source file:

- 1 Select the source field (called "Main field").
- 2 Indicate the **Separators** used within the values in the main field.
- 3 Enter the **Number of separators to ignore**. AssetCenter keeps all the data that follows these separators.

- 4 Enter the **Number of separators to include**. AssetCenter keeps all the information between the start of the text to retain and the separator that follows the last separator to include.
- 5 Check the **Start extraction from end of field** check box if you want AssetCenter to include the "Number of separators to ignore" and the "Number of separators to include" from the end of the field.

Example:

- 1 **Separator:** /
- 2 **Number of separators to ignore:** 2
- 3 **Number of separators to include:** 3
- 4 Value of field in source file: "1/2/3/4/5/6/7/8/9"
- 5 Value imported into the database: "3/4/5/6" if the **Start extraction from end of field** check box was not checked and "4/5/6/7" if it was checked.

Fixed value

This mode lets you include a combination of:

- Character strings enclosed with ".
- Variables. These are certain variables that result from functions used in default field values such as AmLoginName(), AmDate(), AmCounter().

Tree mode

This mode lets you build a tree structure from a single field in the source file.

- 1 Select the source field (called "Main field").
- 2 Indicate the "Separators" used between the sub-values in the field.

AssetCenter divides the source field into sub-values. The number of sub-values equals the number of character strings separated by separation characters. AssetCenter then creates a record for each sub-value and organizes them into a hierarchy.

Example:

- 1 Create a text file containing a column entitled "Name". The value of the one of the lines in the file is "/France subsidiary/Sales head office/Marketing department".
- 2 Configure the import module by creating a "Tree mode" type formula field (the separator is "/"). Its name is "FormulaField". Create a "Fixed value" type formula field (value = "1") and map it to the **Department** field (bDepartment) field (in order to create departments but not employees).
- 3 Map "FormulaField" to the **Name** field (SQL name: Name) in the table of employees.
- 4 Launch the import of the file
- 5 Result: 3 departments linked hierarchically are created: "France subsidiary", "Sales head office" and "Marketing department".

File

This mode allows you import a file into the database. This useful when importing images or long blocks of text.

Files can only be imported into the following types of field:

- Memo
- Blob


The following files formats are supported:

- ANSI text
- Images (all the image formats supported by AssetCenter can be imported).

In the field calculation formula, indicate the source field giving the full pathname of the file to be imported (path, name and extension). By default, the current folder is used as the path.

Script

This mode allows you calculate a value using a Basic script. This script can make reference to imported source fields.

To build the calculation script, either enter the code directly or use click the  button to use the expression builder.

The script cannot reference fields in the database.

Test

 Note:

This field is only shown when the formula type of the field is set to "Fixed extraction" or "Delimited extraction".

Enter the field value of your choice.

Result

 Note:

This field is only shown when the formula type of the field is set to "Fixed extraction" or "Delimited extraction".

The simulated import value from the test data appears in this field.

Step 8: Special cases

Importing departments and employees

When importing records from the table departments and employees, it is sometimes necessary to specify if the imported record is a department or an employee.

There is a field that allows you to do this: the **Department** field (SQL name: bDepartment). It is set to "1" for departments and to "0" for employees. By default, the import module considers its value to be "0".

We recommend creating a "Fixed value" type formula field taking the value "1", and attaching this formula to the **Department** field when the imported item is a department.

Note:

The import module deduces that a record is a department when this record in turn has a child record in the table of departments and employees; employees cannot have child records.

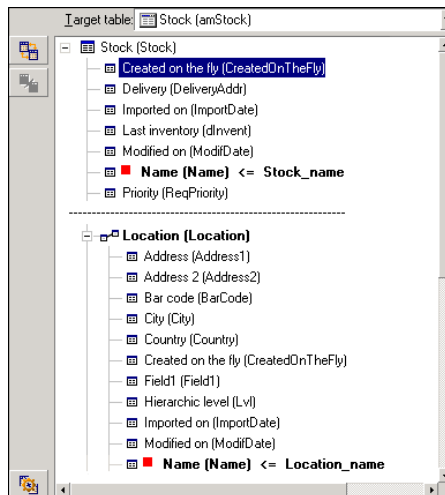
Importing documents

When importing documents you must specify a field to be imported in the **Table** field (SQL name: DocObjTable) in the table of documents. The Table field indicates the SQL name of the table to which the document is linked.

Step 9: Examples of using keys

Here is how AssetCenter interprets your choice of keys:

Example 1: Using a linked field as key of the main table



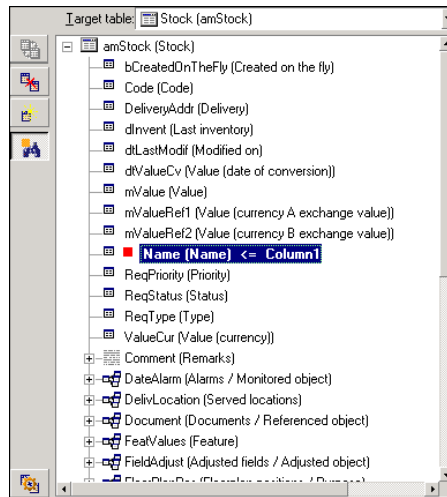
In this example, a stock is identified by two main keys:

- Stock.Name: Because the **Name (Name) <= Stock_name** field is declared as the reconciliation key and it is part of the main table.
- Location.Name: Because the **Name (Name) <= Location_name** linked field is declared as the reconciliation key in the table of locations and the **Location (Location)** link is declared as an reconciliation key.

In this same example, a location is identified by a main key:

- Location.Name: Because the **Name (Name) <= Location_name** field is declared as the reconciliation key in the Locations table.

Example 2: Defining a field as a key in a linked table, even though it is not a key in the main table



In this example, a stock is identified by a single key:

- Stock.Name: Because the **Name (Name) <= Stock_name** field is declared as a key and it is part of the main table.
- And the **Location (Location)** link is not a reconciliation key.


In this same example, a location is identified by a key:

- Location.Name: Because the **Name (Name) <= Location_name** key is declared as main key.

Conclusion

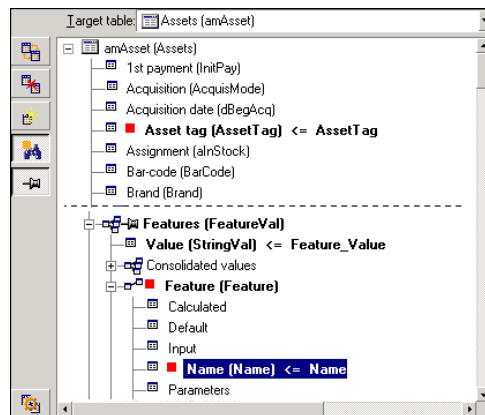
- You can define keys for the main table and other independent keys for tables linked to the main table. You can thereby import data in several tables from a single text file.
- To declare a field in a linked table as one of the keys in the main table, you must declare the field in the linked table AND the link as identification keys. If you only check the link, the key will only be used as key for the linked table itself.

Example 3: Keys that update feature values in database records





To update a record's feature value with a certain value, you must find the (record, feature) pair in the database and assign it the new value. If this pair does not exist, AssetCenter refers to the options defined using the  icon for the link, to determine whether or not it should create a linked record.

You can use either of these two methods:

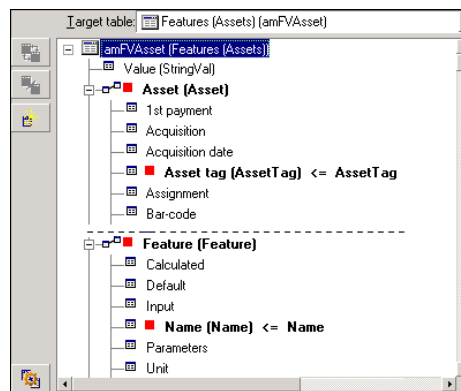
First method (example from the assets table):









- 1 The key on  Asset tag (AssetTag) identifies the asset to be modified.

- 2 The  icon on the  `Features (FeatureVal) *` link specifies that you only want to find the features of this asset.
- 3 The key on  `Feature (Feature) *` indicates that the (feature, value) pair is identified by the feature.
- 4 The key on  `Name (Name) <= Feature.Name *` indicates that the feature is identified by its name.
- 5 The new value appears in the `ValString (Value) <= Feature_Value` field.

Second method (example from the Asset feature values table):

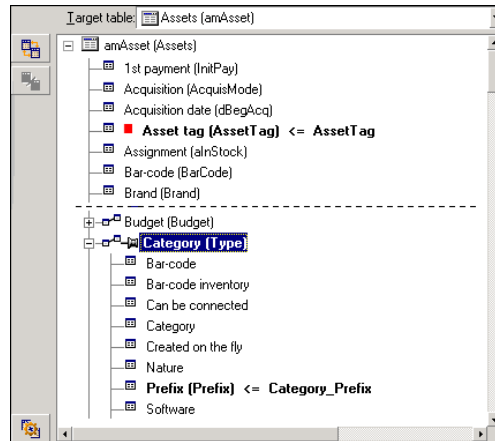




- 1 The ( `Feature (Feature) *`,  `Asset (Asset) *`) pair identifies the feature associated with the asset.
- 2 The key on  `Name (Name)` indicates which is the key field of the  `Feature (Feature) *` link.
- 3 The key on  `Asset tag (AssetTag)` indicates which is the key field of the  `Asset (Asset) *` link.
- 4 The new value appears in the `ValString (Value) <= Feature_Value` field.

Example 4: Modifying the contents of a linked record for which you have no reconciliation key

Example

You want to modify the prefix of the model that is linked to a given asset. In your import file, you have no key that lets you identify the model. You only know that the model is linked to a given asset.




- 1 The key on ■ `Asset tag (AssetTag) <= Asset.AssetTag` identifies the asset.
- 2 The  symbol on `Category (Type)` specifies that you only want to find models already linked to this asset.
- 3 The  `Prefix (Prefix) <= Category_Prefix` assigns a new value to the **Prefix** field (SQL name: Prefix) of the model.

 **Note:**

To optimize performance, we recommend selecting the keys from among the keys that constitute indexes in the table. (Warning: Certain indexes are made up of several fields.)

Step 10: Configuring the transfer

When importing a single text file

Once you have assigned the source fields to fields in the database in step 5, click the  button to display the transfer settings window.

When importing a set of text files or an ODBC database

Use the **Edit/ Options** menu from the import module's main screen.

"Error handling" frame

Select your preferred method for handling errors:

Stop import if error

The import process is halted as soon as an error is encountered.

Commit each imported line

The import module performs a database commit for each imported line. If any errors are detected in the line (either in the main table or in a linked table), the entire line is not imported. The program restores the database to its state before this line was imported.

Commit by group of lines

The import module processes the import by groups of lines; you specify the number of lines per group. If any errors are encountered in the group (either in the main table or in a linked table and for any line of the group), none of the lines in the entire group are imported. The program restores the database to its state before this group was imported. This guarantees database integrity.

"Log file" frame

Select the operations that should appear in the log file:

- Errors
- Appends and updates

Enter the name of the log file and its path. AssetCenter will create the file if it does not exist. Add the extension of your choice. We recommend you use **.log**.

 **Warning:**


The import module cannot create directories.

The log file also provides the following information:

- Time the job was dispatched.
- Job description.
- Errors encountered.

The log file is overwritten for each new import procedure.

Step 11: Data transfer

At each of the preceding steps, you can click the  button to start the data transfer if enough information is available.

Behavior at the record level

- AssetCenter imports data line by line in the order the data appears in the source file.
- AssetCenter can import data into several different tables from a single line.
- If part of an imported line cannot be imported, AssetCenter creates what it can create.
- AssetCenter searches the AssetCenter database for a record whose identification keys have exactly the same values as the source record. If such a record is found, AssetCenter modifies it according to the information contained in the text file.
- If no records are found that match the identifier keys, AssetCenter creates a new record in the database.
- If you do not define any reconciliation keys, the import module adds the imported records if the values to be imported respect the unique nature of the basic fields. Without any keys, the import module cannot update records.
- The import module works the same way for main items and linked items.

Behavior at the field level

- The import module does not automatically verify if a field is mandatory. You must verify yourself that mandatory fields are indeed present in the data to be imported.
- If the import module finds an unknown value for a field that is in a "system" itemized list, the line to be imported is rejected.
- If the import module finds a value that is not yet in a standard itemized list, the line to be imported is accepted and the new value is added to the itemized list if the list is "open". If the itemized list is "closed", the line to be imported is rejected.
- If a field value exceeds the maximum length, the end of the value is truncated.
- Fields with an empty value in the text file erase the existing value in the corresponding database field.
- If importing a line of data causes a new record to be created, AssetCenter inserts the default value in fields that do not appear as a column in the text file or the source table. If the column is present but no value is specified, AssetCenter inserts an empty field instead of the default value.

Saving and executing an import script

A script is a group of import settings saved under a given name. Import scripts let you reproduce similar import conditions without re-defining all the parameters. They help you save time.


Scripts are useful for:

- Executing the same import operation several times, until it works the way you want. (You can modify the source field on each execution, for example.)
- Performing regular database updates. (You can update the Employees table based on a file from the Human Resources department, for example.)

AssetCenter lets you save scripts and execute them later.

To save a script

When importing a single text file


- 1 Use the **File/ Import** menu to select the text file.
- 2 Define the import settings (name and location of the data file, its structure, etc.).
- 3 You can click the  button at any time to save these conditions as a script.

When importing a database or a set of text files

- 1 Use the **File/ Import** menu to select the database.
- 2 Define the import settings (location of the database or text files, mappings between fields, etc.).
- 3 You can save these settings as a script file at any time using the **File/ Save** or the **File/ Save as** menu.

To modify a script

When importing a single text file

- 1 Use the **File/ Import** menu.
- 2 Indicate that you are importing a text file.
- 3 In the "Open data file" dialog box, select "Import scripts *.scr" files in the "list of file types" field.
- 4 Open the script.
- 5 Modify the import settings (name and location of the data file, its structure, etc.).
- 6 You can click the  button at any time to save these new settings.


When importing a database or a set of text files

- 1 Use the **File/ Import** menu to import a database.
- 2 Populate the **ODBC** tab or the **Text** tab. Click **Open**.

- 3 Next, open the script using the **File/ Open script** menu.
- 4 Modify the import settings (name and location of the database, its structure, etc.).
- 5 You can save these settings as a script file at any time using the **File/ Save** or the **File/ Save as** menu.

To create a new script when you are defining import settings

Importing a single text file


Click the  button. Proceed as you would to create a new script.

Importing a database or a set of text files

Use the **File/ New script** menu. AssetCenter offers to save the current settings as a script before discarding them.

To execute a script

To execute an existing script:

- 1 Use the **File/ Import** menu.
- 2 Click the  icon.
- 3 Indicate the name of the script file.
- 4 Start the import procedure.

You may also proceed as if you were going to modify a script and then execute the import procedure from within the import dialog boxes.

Executing an import script from a command prompt

How it works

To execute the DOS program "online", you need to have created a script beforehand with the Windows import module.

You can execute an import command manually or automatically (by using a batch file, for example). Either way, you need to use an **amimpl.exe** file, which is located in the **bin32** folder of AssetCenter's installation folder.

Syntax

```
amimpl32 [-verbose] [-?|h|H] -src:<cnx> [-srcpass:<password>]
-dst:<cnx> [-dstlogin:<login>] [-dstpass:<password>] [-log:<file>]
```

-verbose: displays messages during import. Enabled by default.

-, -h or -H: displays help messages.

-src: according to the case, this parameter shows:

- The path and name of import script to execute.
- AssetCenter database connection name to import completely (as stated in the **File/ Manage connections** menu, **Name** field).
- The name of an AssetCenter database without connection:

```
[<EngineName>;<DatabaseLocation>;<User>;<Password>]
```

In this case, here is how to populate the different fields shown above between <>:

	Oracle	MS SQL Server	Sybase SQL Anywhere	Sybase SQL Server
EngineName	Oracle	ODBC	ODBC	Sybase
DatabaseLocation	Server name	Data source name	Data source name	ServerName:DatabaseName
User	Account name	MS SQL Server user name	Sybase SQL Anywhere user name	Account name

	Oracle	MS SQL Server	Sybase SQL Anywhere	Sybase SQL Server
Password	Account password	MS SQL Server user password	Sybase SQL Anywhere user password	Account password

-srcpass: password associated with the source database to be imported. In the case of an AssetCenter database, this is the password of the "Admin" account.

-dst: AssetCenter database connection name in which the data is to be imported (as stated in the **File/ Manage connections** menu, **Name** field).

-dstlogin: login name of AssetCenter account that will receive the imported data ("Admin" account or AssetCenter user with administration rights).

-dstpass: password associated with "dstlogin".

-log: full pathname of the import log file.

 **Note:**

If the strings between angle brackets <> contain spaces, you must place them within single quotes (').

Example:

```
amimpl32 -verbose -src:employee.scr -srcpass:Password
-dst:MainDBase -dstlogin:Gerald -dstpass:Password -log:'My
Log File.txt'
```

11 | User profiles

CHAPTER

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

 **Note:**

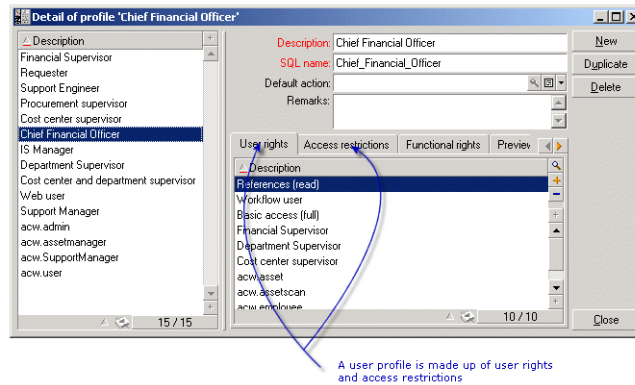
Access rules can only be managed by database administrators.

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

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

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

Use the **Administration/ Functional rights** menu to display the list of functional rights.

Figure 11.1. User profile - detail window

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:

- 1 Defining the data that each user can access, and defining the access conditions.
- 2 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.

- The Login field must be populated (Employees and Services table, Profile tab).
- The administrator must assign a "user profile" to the employee.

A user profile consists of user rights, functional rights and access restrictions.

A user right, a functional right or an access restriction can be used in several user profiles. A user right or functional right 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 a functional 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 that 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 that apply to the different tables in AssetCenter.

Definition of a functional right

A functional right is one of the component parts of a user profile. It concerns categories of functionalities (such as Procurement, Cable and Circuit, etc.) and is based on functional domains. As an administrator, you can assign rights to users according to their jobs and to what directly relates to them and their fields.

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 functional rights.
- Defining access restrictions.

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

Defining user profiles

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

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.

Object	Rights (RIU/L)
+ Product reservation (amProdReserv)	riU/CD
+ Product suppliers (amProdSupp)	riU/CD
- Products (amProduct)	riU/CD
+ (Fields)	RIU
+ (Links)	RIU
- (Features)	riU
+ Architecture (fv_Architecture)	RIU
+ Backup capacity (fv_BakCapacity)	RIU
+ BIOS asset tag ID (fv_BiosAssetTag...)	RIU
+ BIOS brand (fv_BiosBrand)	RIU
+ BIOS date (fv_dBiosDate)	RIU
+ BIOS identification string (fv_BiosSo...)	RIU
+ BIOS machine (fv_BiosMachine)	RIU
+ BIOS revision (fv_BiosRevision)	RIU


- 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 that determines 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 functional rights.

Functional rights can be managed via:

- The **Administration/ Functional rights** menu.
- Or by clicking the  button to the right of the "functional rights" list in a user profile detail (**Administration/ User profiles** menu).

The functional-right edition screen proposes a hierarchical list of all of AssetCenter's functional rights, which are defined by the user.

A functional right defines the functional domain of the application (**Item** column) in which a user can operate.

Example

In the Procurement functional domain, the person in charge of billing has access to the supplier invoices but not to reservations.

The logic of functional rights.

Each functional domain (**Item** column) is composed of a set of items to which a right is attributed (**Granted, Denied, Value of parent**).

Functional domains inherit the **Default value** of the right defined for the set of listed items (their domains and their items).

Each item of a functional domain inherits the default value of its parent (**Default value of the parent** field validated).

Each functional domain can have a different value from that in the **Default value** field. In this case, the items of these domains inherit its value (**Value of the parent** of the items in this validated domain).

Each item of a domain can have a different value from that of its domain. In this case, the value of each item is defined in the **Granted** or **Denied** field.

For example, for the set of items in a functional domains with the value **Denied**, the following conditions must be met:

- The **Default value** field has the value **Denied**.
- The functional domain has the **Value of the parent** value.
- The set of items in the domain have the **Value of the parent** value.


or

- The **Default value** field has the value **Granted**.
- The functional domain has the value **Denied**.

- The set of items in the domain have the **Value of the parent** value.
- OR
- The **Default value** field has the value **Denied**.
 - The functional domain has the value **Denied**.
 - The set of items in the domain have the **Value of the parent** value.

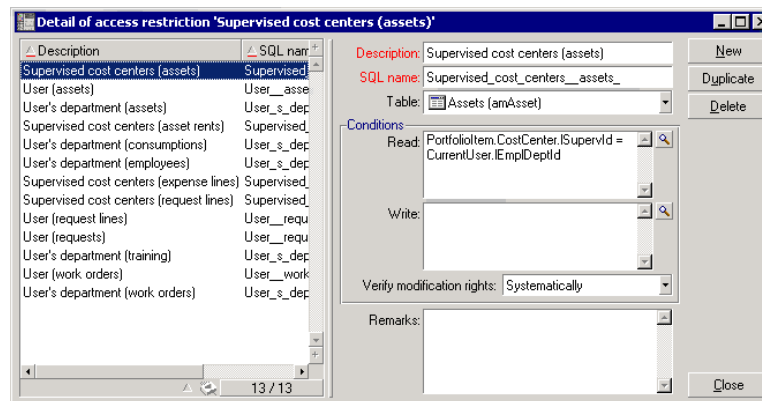
Defining access restrictions

Access restrictions can be managed via:

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

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

Figure 11.2. User profiles - Access restrictions



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

- Models or brands
- Departments or locations

- Insurance contracts

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

- Maintenance engineer in St. Louis
- Purchasing supervisor
- Etc.

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 that 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.

 **Warning:**

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

Verify modification rights

This field enables you to define the conditions for the modification rights. The use of this field depends on the AQL script in the **Write conditions** field.

Example of modification rights

Populate the **Write** field (WriteCond) with the following script:

```
seAssignment=1
```


If you select **Before any modification**, you can modify an item in stock or delete it.

If you select **After any modification**, you can insert an item in stock and modify it.

If you select **Systematically**, you can insert or modify an item in stock but not delete it.

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:

- 1 Create the user in the table of departments and employees.
- 2 Move to the **Profile** tab in the employee detail.
- 3 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.

- 4 Populate the **Login type** (SQL name: seLoginClass) field. Refer to the following paragraph to see the different options.

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

Note:

There is a default administrator in the table of departments and employees: 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.

Managing user connections

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

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 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.

Furthermore, it is 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.

This login type is restricted to users accessing the database via AssetCenter Web.

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.

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 **Administration/ Database options** menu. 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


The AssetCenter administrator can view connection slots via the **Administration/ Connection slots** menu.

A connection slot is defined by:

- The name of the application that 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:

- 1 Display the connection slot screen via the **Administration/ Connection slots** menu.
- 2 Select the appropriate connection slot.
- 3 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 that 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).

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:

- 1 AssetCenter searches for the date the slot was last modified.
- 2 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:

- 1 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.
- 2 Select those slots and delete them.

Disconnecting inactive users

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

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

- 1 The **Automatic disconnection** in the "Access control" section option allows you to specify if you want to activate the automatic disconnection procedure.
- 2 If you decide to activate the automatic disconnection procedure, then you need to define a value in the **Slot timeout in seconds** option of the "Access control" section.

 **Warning:**

In order for your modifications to be recognized, you need to disconnect, then reconnect.

 **Note:**

If you use a AssetCenter version 3.01 or earlier, the automatic disconnection is applied to the ensemble of users.

Managing passwords

This section explains how to manage passwords with AssetCenter.

Password for the Admin login

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

- 1 For the first installation of AssetCenter, this is the only login name available for accessing the AssetCenter database for all administrative operations.
- 2 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:**

Warning: Do not forget the password for the "Admin" login. Should this happen, you will have no other means of administering the database if the other administrative-rights records in the table of departments and employees 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.

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.

Modifications performed by the administrator

The 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 in AssetCenter. You therefore will need to contact Peregrine System support.

12 | AssetCenter Server

CHAPTER

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.

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.
- Searches for new workflow execution groups.
- Execution of workflow rules.
- Verification of time zones.
- Synchronization of data with AutoCAD.

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.

 **Warning:**

When you exit AssetCenter Server, all automatic monitoring functions are suspended if they have not been launched as services.

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 enables you to improve the performance of AssetCenter Server.

 **Note:**

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 performed 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 shortcut menu, you must disconnect AssetCenter Server from the database and then reconnect.

Launching AssetCenter Server

Manually launching AssetCenter Server

You can launch the AssetCenter Server program from the programs in the **Start** menu or in the AssetCenter program group.

 **Warning:**

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.

Automatically launching AssetCenter Service as a service

To launch AssetCenter Server as a service:

- 1 Manually launch AssetCenter Server.
- 2 Choose the connection to your database and select the **Use this connection in service mode** option.
- 3 Exit AssetCenter Server.
- 4 In the Windows configuration panel, select the AssetCenter Server service and set its launch to automatic mode.

 **Warning:**

You need to have rebooted the computer on which AssetCenter Server is installed after installing the DBMS client layers of the base to which the AssetCenter Server service is connected. This is because the Windows service control manager reads the **PATH** environment variable when you start the computer. The AssetCenter Server service, however, needs an updated **PATH** variable in order to find the DLLS of the DBMS. This is especially true if the AssetCenter installation program installs SQL Anywhere runtime and if you want to run the AssetCenter Server service on a demonstration database (which uses SQL Anywhere).

From the DOS command prompt

You can automate the launching of AssetCenter Server by using the following command:

```
amsrvaamsrv23 -cnx:<connection name> -login:<login>  
-password:<login password>
```

 **Warning:**

The **login** is that of an AssetCenter administrator (either "Admin", or a user login with administration rights).

Strings between <> should have no spaces.

Example:

```
amsrv32 -cnx:BasePeregrine -login:Gerald -password:Password
```

This command can be placed in a batch file.


Executing AssetCenter Server manually under Windows

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. Only the administrator can connect to the database via AssetCenter Server.


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 dialog box that appears when starting.
- The **File/ Connect** menu.
- The  icon.

Disconnecting AssetCenter Server from a database

To disconnect AssetCenter Server from the database, use one of the following:

- The **File/ Disconnect** menu.
- The  icon.

Executing AssetCenter Server on Unix

Depending on the type of Unix used, you will have to modify the system configuration files to launch the **aamsrvl** executable at startup. The arguments of the executable are the following:

- **-webadmin** to administer the AssetCenter Server module via the Web,
- **-cnx: <Name of the connection>**

- **-login:** <User name>
- **-password:** <Password>
- The configuration options (default database, Web listening port, etc.) must first be modified in the **amsrvcf.ini** file, installed next to the executable.

Refer to the **Installation** guide for information about compatibility and installing AssetCenter on Unix.

Using the security-integrated Windows NT, 2000 or XP

Using the **integrated security** of Windows (also called **unified login**) consists of synchronizing the security information from AssetCenter with the information from the Windows User Manager.

With this synchronization:

- You can automatically import the list of employees declared in the User Manager into the AssetCenter database.
- The users who connect to an AssetCenter database don't have to populate the **Login** and **Password** fields when connecting to the database.

General overview

- It is the Connect-It application that synchronizes AssetCenter and the User manager.
- The Connect-It scenario that performs this task is programmed and launched by AssetCenter Server.
- Connect-It uses the following combination of fields to identify the users of the AssetCenter database:
 - **Name** (Name)
 - **First name** (FirstName)
 - **Login** (UserLogin)
- A user is authorized to connect to the AssetCenter database using a given login if the following items have the same value:

- **SID** field (Security Identifier) of the AssetCenter user corresponding to the Windows login.
- **SID** associated with the Windows login.

Main implementation steps

Here are the main steps you need to take to implement this system. We will elaborate on each one of these steps later on in the guide.

- 1 Create the users and groups in the Windows NT User Manager.
Objective: Prepare the information to transfer to AssetCenter.
- 2 Configure AssetCenter Server and Connect-It.
Objective: Define which NT domains retain and plan the transfers.
- 3 Trigger AssetCenter Server's **Add NT users to the database** module as well as any other modules that you might need.
Objective: Transfer the information for the first time and test the login.
- 4 Complete the description of the employees in the AssetCenter database.
- 5 Activate the automatic triggering of the AssetCenter Server modules.

Step 1: Create the users and groups in the Windows NT User Manager

The fields to populate are:

- For the users:
 - **Full name** field:

 **Tip:**

When you create users in the AssetCenter database, AssetCenter Server takes the value of the **Full name** field (from the User Manager) and looks for the first **space** character starting on the left. Everything it finds to the left of the first space is used to create the **First** field; everything to the first of the first space creates the **Name** field. If there is no space, only the **Name** field will be populated.

You thus do not have to use a space in first names having more than one word. You can instead use a hyphen, if you wish.

- **Description** field:
This field is used to populate the **Comment** field (Comment) in the AssetCenter database.
- For groups: **Name**.
This field is used to populate the **Name** field (Name) in the AssetCenter database.

Step 2: Configure AssetCenter Server and Connect-It

This step is described in chapter **AssetCenter Server**, section **Configuring the modules monitored by AssetCenter Server/ Add NT users to the database module (AddUser)** of this guide.

Step 3: Activate the updating process for the first time

- 1 Select the **Action/ Activate** menu.
- 2 Select the option corresponding to the **Add NT users to the database** task.
- 3 Click **OK**.

AssetCenter begins the update.

 **Note:**

If the number of users concerned by this operation exceeds the number of **Named** logins authorized by your license, AssetCenter Server declares the users as having **Floating** type logins.

Step 4: Complete the description of the employees in the AssetCenter database

Your database now contains one employee per NT user detected on the chosen domain. Their login is in the form:

[domain]\[user]

The users created do not have a valid password.

 **Note:**

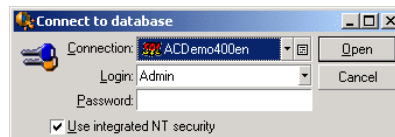
Following this operation, we advise you to verify that all the records created in the Departments and Employees table correspond to an AssetCenter user. You must notably re-enter the value of the **Password** field.

Step 5: Activate the automatic triggering of the AssetCenter Server modules

- 1 Select the **Tools/ Configure modules** menu.
- 2 Select the **Add NT users to the database** module.
- 3 Populate the fields of the **Verification schedules** frame.
- 4 Check the **Enabled** box.
- 5 Click **Modify**.

AssetCenter connection

Once the previous operation is completed, the NT users can directly access AssetCenter. During their first connection, all they need to do is select the **Use integrated NT security** option, then click **Open**.



No other authentication information will be asked for the next connections.

If a user wants to connect under another login, however, they can always do so by activating the connection box via the **File/ Connect to database** menu.

Reference information

Supported environments

- Windows domain: Windows NT 4 is supported. Windows Active Directory 2000 is not supported.
- Windows clients: Windows NT, 2000 and XP are supported. Windows ME, 95 and 98 are not supported.

Rules applied by AssetCenter during the creation/modification of users

- The AssetCenter login is created by concatenating the Windows **Domain** name and **User name** in the form of: **<Domain name>\<User name>**
- The following set of fields are used as the reconciliation key for the users of the AssetCenter database:
 - **Name** (Name)
 - **First name** (FirstName)
 - **Login** (UserLogin)
- When a user is imported, the user is only created if the reconciliation key does not yet exist in the database. All the fields corresponding to the Connect-It scenario are then populated.

If a record corresponding to the reconciliation key already exists, then all the fields in the Connect-It scenario will be updated except those constituting the reconciliation key.

 Note:

The **Password** field is populated with a special character that prohibits the user to access the AssetCenter database via any other means than the automatic login. You thus must populate this field manually if you want to authorize this user to access the database using a manual login (especially if the user wants to login on a different computer).

- Remember that there is an index in AssetCenter composed of the **Bar code** field. This index must be unique. You also need to verify that the **Bar code** field is populated with a different value for each user. This way, you can create different users who might share the very same name. For this, you can use the default value calculation script installed by AssetCenter.

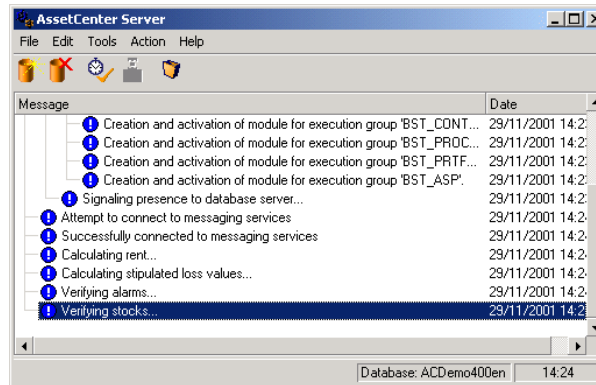
If you exit AssetCenter Server

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



The main window displays all events handled by AssetCenter Server. It enables you to access the following program menus and icons:

Table 12.1. AssetCenter Server program menus and icons

Icon	Menu	Function
	File/ Connect	Connect to a database.
	File/ Disconnect	Disconnect from a database.
	Action/ Activate	Selectively activate deadline monitoring agents.
	Action/ Empty list	Clear messages from main screen.
	Action/ Connect to messaging	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.

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. AssetCenter Server tries to reconnect to the database according to the frequency defined in the **Database** field in the **Reconnection delays** frame.

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 enables you to limit the size of the file that 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 tests 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 five 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 about the time zone of the server in the table of options of AssetCenter is updated (if the number $(n * 30\text{minutes})$ 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 about 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.

Configuring the modules monitored by AssetCenter Server

Introduction

To consult or configure a model monitored by AssetCenter Server, select the **Options/Configure** modules menu.

The window that appears enables you to display the list of modules and to define for each module:

- If the module is enabled or not.
 - Which task the module must perform.
 - How frequent each module is triggered.
-

Tip:

You can launch these modules on several machines by executing an AssetCenter Server session on each one of them. This can increase performances. Make sure that a module is only enabled on one machine at a time.

The following sections describe each one of the modules managed by AssetCenter Server. These modules are identified by the description and the name that is listed in AssetCenter Server.

Verification schedules

To define the verification schedules of a module:

- 1 Select the **Options/ Configure modules** menu.
- 2 Select the module to configure.
- 3 Populate the tab to the left of the **Verification schedules** zone.
- 4 If necessary, create and populate additional sub-tabs by right-clicking the labels of the sub-tabs and selecting the **Add a rule** menu.

The sub-tabs are used to enable you to define the days and times of monitoring.

Table 12.2. 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 Day , Month and Year check boxes. Example July 20

Value of the Days field

"The first"
 "The second"
 "The third"
 "The fourth"
 "The next to last"
 "The last"

Monitoring is performed

The days of the week defined using the **Day** check box, and for the month(s) and year(s) defined using the **Month** and **Year** 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 2000.

Monitoring times

Periodical

You can define two frequencies for the verification of a module that 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 configuration window for monitoring times. It features a 'Times' dropdown menu currently set to 'Periodical'. To the right of this menu is a schedule bar with a grid of days (3, 6, 9, 12, 15, 18, 21) and a time range of 8:00-19:00. Below the schedule bar, there are two input fields: 'During period:' set to '15min' and 'Outside of period:' set to '30min'. Both input fields have small arrows indicating they are adjustable.

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:

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.

Verify history lines module (History)

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.

Verify stocks module (Stock)

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 model:

- AssetCenter Server calculates the quantity of items actually available from the **Assignments** field in the detail of a portfolio item.

- 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** and **Management** tabs 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.

Verify alarms module (Alarms)

List of alarms monitored

At the asset level

Several key dates are monitored:

- The end-of-reservation date of an asset: This is shown in the **Reserv. end date** field (SQL name: dtEnd) in the **Portfolio/Reservations** 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).

- End-of-rent 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-of-reservation date 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:

- 1 Launch AssetCenter.
- 2 Select **Procurement/ Purchase requests**.
- 3 Select the purchase request reserving the consumable.
- 4 Display the composition of this purchase request.
- 5 Display the request line corresponding to the consumable.
- 6 Display the **Reservations** tab of the request line. This tab shows the list of reservations for consumables.
- 7 Display the detail of the reservation.

The monitored field is **Date fin** (Nom SQL : dtEnd).

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 of the **Rent** tabs.

At the purchase request 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 (**Acq. method** 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 alarms with 2 levels, the triggering of the second level alarm depends on the action carried out at the first level.

- 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.

Calculate rents module (Rent)

AssetCenter Server monitors periodic rent payments for contracts and assets. It calculates and/recalculates the amounts involved.

The **Calculate rents** module 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, which 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 enables you to recalculate erroneous expense lines in case of a change in tax rates, for example.

Parameters

The **User data item** field 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

 **Note:**

The maximum number of rent calculations made per transaction is specified by the `UserData` entry in the `Amsrv.cfg` configuration file.

Projected rent

The **User data item** field enables 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, 2001 through July 1, 2004.
- 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, 2002, AssetCenter Server is launched for the first time: it generates:

- Past rents from July 1, 2001 through July 1, 2002.
- The present rent on July 1, 2002.
- The projected rents from August 1, 2002 through July 1, 2003.

Following these calculations, the Recalculation effective from field indicates the date of the last non-projected expense line, i.e. July 1, 2002.

AssetCenter Server runs in the background: 2 months later on September 1, 2002, it generates:

- The projected rents from October 1, 2002 through September 1, 2003.

- 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, 2002 through October 1, 2002.

Calculate stipulated loss values module (LostVal)

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 the lease contract detail). In this way, loss values pertaining to any loss value rules that have been modified since the last time AssetCenter Server accessed the database are updated.

Split expense lines in cost centers module (CostCenter)

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 **Edit** category of the **Edit/ Options** menu 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 (seSplitStatus) so it displays "Not split" at the level of the expense lines highest up in the split operation. When these high-level expense lines were split, they 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 that, 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. It thus recalculates, using new parameters, all the expense lines that, 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.

Verify database server time zone module (TimeZone)

This module verifies the delay between the local time of the server and the client machines. This is useful if you specified a time zone for a client machine (menu **Administration/ Time zones**).

Signal presence to database server module (UpdateToken)

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.

Search for new workflow execution groups module (WorkflowFinder)

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 enables you to define verification timetables for each workflow execution group.
- Different workflow execution groups can be monitored by different instances of AssetCenter Server.

Execute workflow rules for execution group 'G' modules

Once a workflow execution group **G** 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 event's processing mode 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.
- Otherwise, the AssetCenter client machine executes the task.

Add the computers listed in the NT domain to the database module (AddCpu)

AssetCenter Server enables you to program the recovery of those computers declared in the NT domain.

The domain to analyze is specified at the Connect-It **addcpu.scn** scenario.

Before activating the **Add the computers listed in the NT domain to the database** module, you need to verify that the following parameters are correct:

- Parameters of the AssetCenter Server module **Add the computers listed in the NT domain to the database**.
- Parameters of the Connect-IT **addcpu.scn** scenario, which is located in the **scenario\ntsec\ntac40** sub-folder of the Connect-It installation folder.

Parameters of the module Add the computers listed in the NT domain to the database (User data item field).

Here is the parameter line by default:

```
"$connectit_exedir$/conitsvc.exe" -once
'$connectit_exedir$/../scenario/ntsec/ntac40/addcpu.scn'
-d:AssetCenter.SERVER=$cnx$ -d:AssetCenter.LOGIN=$login$
-d:AssetCenter.TEXTPASSWORD=$pwd$
```

Information about certain parameters in this line:

- `$connectit_exedir$` stores the path of the **conitsvc.exe** program in the Windows Registry.
You do not need to modify this parameter.
- **-once** indicates that **conitsvc.exe** must be executed one time (in other words, using Connect-It's **Once** scheduler.).

Do not modify this parameter; you define the programming at the level of the AssetCenter application.

- `$connectit_exedir$/../scenario/ntsec/ntac40/addcpu.scn` is the path to the Connect-It scenario to use..

Modify this parameter if you want to use another Connect-It scenario.

- `-d:AssetCenter.SERVER=cnx`
`-d:AssetCenter.LOGIN=$login$`
`-d:AssetCenter.TEXTPASSWORD=pwd` stores the AssetCenter connection name opened by AssetCenter Server, as well as the login and password used for the connection.

These parameters overwrite the values defined at the level of the AssetCenter connector in the **addcpu.scn** scenario.

You have no need to modify these parameters.

Parameters of the addcpu.scn Connect-It scenario

To modify the **addcpu.scn** scenario:

- 1 Execute the Connect-It scenario editor.

- 2 Open the **addcpu.scn** scenario, which is located in the **scenario\ntsec\ntac40** sub-folder of the Connect-It installation folder.
- 3 Select the **NT Security** connector in the **Scenario diagram** window by clicking on the title bar of the **NT security** box; not inside of the box itself.
- 4 Select the **Tools/ Configure** menu.
- 5 Click **Next**.
- 6 Populate the **Domain(s)** field with the name of the domain from which you want to import the computers.

 **Warning:**

The computers that can be recovered are:

- Those with the same domain as the one to which AssetCenter Server is connected.
- Those domains that are **trusted** by the one to which the AssetCenter Server is connected.

 **Tip:**

To find out whether or not a computer has been recovered:

- 1 Execute the Windows explorer on the AssetCenter Server machine.
- 2 Display the network neighborhood.
- 3 The computers that you see are those from which AssetCenter Server could recover information.

To find out which information is populated in the AssetCenter database, examine in detail the **addcpu.scn** scenario.

The module populates the **Next scan** field (dtNextScan) in the **Computers** table (amComputer) with the date of the module's execution at 00:00.

To learn how the **NT Security** connector works, refer to the Connect-It **User's guide**, chapter **Application connectors**, section **NT Security connector**.

To learn how the **AssetCenter** connector works, refer to the **Connect-It User's guide**, chapter **Peregrine Systems connectors**, section **Asset Management connector**.

Add NT users to the database module (AddUser)

AssetCenter Server enables you to program the recovery of the users declared on the NT domain.

This is essentially used to populate the **Departments and employees** table with the information useful for connecting to an AssetCenter database that uses integrated NT security.

The domain to analyze is specified at the Connect-It **adduser.scn** scenario.

Before activating the **Add NT users to the database** module, you need to verify that the following parameters are correct:

- Parameters of the AssetCenter Server module **Add NT users to the database**.
- Parameters of the Connect-It **adduser.scn** scenario, which is located in the **scenario\ntsec\ntac40** sub-folder of the Connect-It installation folder.

Parameters of the module Add NT users to the database (User data item field).

Here is the parameter line by default:

```
"$connectit_exedir$/conitsvc.exe" -once
'$connectit_exedir$/../scenario/ntsec/ntac40/adduser.scn'
-d:AssetCenter.SERVER=$cnx$ -d:AssetCenter.LOGIN=$login$
-d:AssetCenter.TEXTPASSWORD=$pwd$
```

Information about certain parameters in this line:

- `$connectit_exedir$` stores the path of the **conitsvc.exe** program in the Windows Registry.

You do not need to modify this parameter.

- **-once** indicates that **conitsvc.exe** must be executed one time (in other words, using Connect-It's **Once** scheduler.).

Do not modify this parameter; you define the programming at the level of the AssetCenter application.

- `$connectit_exedir$/../scenario/ntsec/ntac40/adduser.scn` is the path to the Connect-It scenario to use..

Modify this parameter if you want to use another Connect-It scenario.

- `-d:AssetCenter.SERVER=cnx`
`-d:AssetCenter.LOGIN=$login$`
`-d:AssetCenter.TEXTPASSWORD=pwd` stores the AssetCenter connection name opened by AssetCenter Server, as well as the login and password used for the connection.

These parameters overwrite the values defined at the level of the AssetCenter connector in the **adduser.scn** scenario.

You have no need to modify these parameters.

Parameters of the **addcpu.scn** Connect-It scenario

To modify the **adduser.scn** scenario:

- 1 Execute the Connect-It scenario editor.
- 2 Open the **adduser.scn** scenario, which is located in the **scenario\ntsec\ntac40** sub-folder of the Connect-It installation folder.
- 3 Select the **NT Security** connector in the **Scenario diagram** window by clicking on the title bar of the **NT security** box; not inside of the box itself.
- 4 Select the **Tools/ Configure** menu.
- 5 Click **Next**.
- 6 Populate the **Domain(s)** field with the name of the domain from which you want to import the computers.

Tip:

If you want to explore several domains, we recommend creating a Connect-It scenario and AssetCenter Server module per domain.

 **Warning:**

The users that can be recovered are:

- Those on the same domain as the one to which the AssetCenter Server user is connected.
 - Those domains that are **trusted** by the one to which the AssetCenter Server user is connected.
-
-

 **Tip:**

To find out if a machine can be recovered:

- 1 Execute the Windows explorer on the AssetCenter Server computer.
 - 2 Make a folder into a shared folder.
 - 3 Define the permissions for this shared folder.
 - 4 Add a user to the permissions.
 - 5 The users that you see are those from which AssetCenter Server could recover information.
-

To find out which information is populated in the AssetCenter database, examine in detail the **adduser.scn** scenario.

To learn how the **NT Security** connector works, refer to the *Connect-It User's guide*, chapter **Application connectors**, section **NT Security connector**.

To learn how the **AssetCenter** connector works, refer to the *Connect-It User's guide*, chapter **Peregrine Systems connectors**, section **Asset Management connector**.

Create assets, consumptions, etc. corresponding to items received module (Delivery)

Prerequisites

This module cannot be executed unless you have already done the following:

- Execute AssetCenter

- Select the **Administration/ Database options** menu.
- Select the **Procurement/Have the application server (aamsrv) create the items received** option.
- Set this option to **Yes**.

Task performed by the module

This module is used to process the records from the **Items received** table in order to create received items (assets, consumptions, etc.) in their respective tables.

Utility of this mode

Assigning this task to AssetCenter Server rather than the AssetCenter application can increase the performances of those users receiving orders.

Frequency of execution

We recommend that you execute this module several times a day if you want the users to be able to quickly access the items received in their respective tables.

Send the scanner to the computers module (SendScan)

This module enables you to download and execute the InfraTools Desktop Discovery **scan32.exe** inventory program on a selection of workstations.

The **scan32.exe** program enables you to create an **.fsf** file that stores the results of the machine scan.

The list of machines on which the **scan32.exe** file is copied is defined by taking all the records from the **Computers** table (amComputer) whose **Next scan** field (dtNextScan) is not empty and is inferior to the module's execution date.

 **Note:**

This module is based on the assumption that the list of machines has already been created using the **Add the computers listed in the NT domain to the database** module.

Before activating the **Send the scanner to the computers** module, you need to verify that the following parameters are correct.

 **Note:**

The default parameters of the Connect-It **sendscan.scn** scenario, located in the **scenario\autoscan\ac40** sub-folder of the Connect-It installation folder, don't need to be modified in order for the **Send the scan to the computers** module to work.

Parameters of the module **Send the scanner to the computers (User data item field)**.

Here is the parameter line by default:

```
"$connectit_exedir$/conitsvc.exe" -once
'$connectit_exedir$/../scenario/autoscan/ac40/sendscan.scn'
-d:AssetCenter.SERVER=$cnx$ -d:AssetCenter.LOGIN=$login$
-d:AssetCenter.TEXTPASSWORD=$pwd$
```

Information about certain parameters in this line:

- `$connectit_exedir$` stores the path of the **conitsvc.exe** program in the Windows Registry.
You do not need to modify this parameter.
- **-once** indicates that **conitsvc.exe** must be executed one time (in other words, using Connect-It's **Once** scheduler).
Do not modify this parameter; you define the programming at the level of the AssetCenter application.
- `$connectit_exedir$/../scenario/autoscan/ac40/sendscan.scn` is the path to the Connect-It scenario to use.
Modify this parameter if you want to use another Connect-It scenario.

- `-d:AssetCenter.SERVER=cnx`
`-d:AssetCenter.LOGIN=$login$`
`-d:AssetCenter.TEXTPASSWORD=pwd` stores the AssetCenter connection name opened by AssetCenter Server, as well as the login and password used for the connection.

These parameters overwrite the values defined at the level of the AssetCenter connector in the **sendscan.scn** scenario.

You have no need to modify these parameters.

Parameters of the sendscan.scn Connect-It scenario

To modify the **sendscan.scn** scenario (this should not be necessary):

- 1 Execute the Connect-It scenario editor.
- 2 Open the **sendscan.scn** scenario, which is located in the **scenario\autoscan\ac40** sub-folder of the Connect-It installation folder.
- 3 Select the **Send scan** connector in the **Scenario diagram** window by clicking the title bar of the **Send scan** box, but not inside the box itself.
- 4 Select the **Tools/ Edit mapping** menu.
- 5 Select the **amComputerSrc-CommandDst/ Command (CommandDst)** line.
- 6 Click the **magnifier**.
- 7 Select the **Arg** element.
- 8 Here is the default mapping script:

```
Dim strArg As String
strArg = "-cmd:scan -certifile:iftscan.cfg
-certifpwd:icwpwd"
strArg = strArg & " -ipaddr:" & [TcpIpHostName]
strArg = strArg & " -ipport:738"
strArg = strArg & " -pwd:password"
strArg = strArg & " -logfile:scanners/scanw32.exe"
strArg = strArg & " -remfile:c:/scanw32.exe"
strArg = strArg & " -remparam:-oC:\" & [WorkGroup] & "_"
& [Name] & ".fsf"
```

Here are some indications on the script parameters:

- `-certiffile:iftscan.cfg -certifpwd:icwpwd` These files are installed in the **bin32\scanners** subfolder of the Connect-It installation folder by the **Automatic inventory** package, supplied on the AssetCenter CD-ROM. They go with the **scanw32.exe** inventory program. They grant you the rights necessary to scan the remote computer.
- `-ipport:738` and `-pwd:password:` These parameters indicate the port and the password of the InfraTools Remote Control agents on the remote computers. Check that the agents on the remote computers are using this port and password or modify these parameters accordingly. If you want to use a different port and password on certain remote computers, modify the mapping script. The script is written in Basic.
- `-locfile::` This parameter defines the full path of the **scanw32.exe** file, installed by the **Automatic inventory** package, supplied on the AssetCenter CD-ROM.
- `-locfile::` This parameter indicates the full path to where the **scanw32.exe** file should be copied on the remote computer.
- `-remparam::` This parameter defines the full path of the **fsf** file, produced when **scanw32.exe** is executed on the remote computer. By default, this file is created at the root of **c:**, and the name of this file is made up of the domain name followed by the name of the remote computer.

 **Warning:**

This mapping script must perfectly match the mapping script of the **getfsf.scn** scenario.

To find out which information is populated in the AssetCenter database, examine in detail the **sendscan.scn** scenario.

The module populates the **Next scan** field (`dtNextScan`) in the **Computers** table (`amComputer`) with:

- If the scan was performed successfully: the date of the module's execution + 7 days.

- If the scan was performed unsuccessfully: the date of the module's execution + 1 day.

 **Tip:**

A history of scans performed successfully or unsuccessfully is kept in the **Scans** field (ScanHistory) in the **Computers** table (amComputer).

For more information about the **Command line** connector, refer to the Connect-It **User's guide**, chapter **Protocol type connectors**, section **Command line connector**.

For more information about the **AssetCenter** connector, refer to the Connect-It **User's guide**, chapter **Peregrine Systems connectors**, section **Asset Management connector**.

To obtain a documentation of the **sendscan.scn** scenario, refer to the Connect-It **User's guide**, chapter **Scenario documentations**, section **Creating a scenario documentation**.

Troubleshooting

If you have any difficulties in executing this module, make sure of the following:

- An InfraTools Remote Control Agent is installed on the remote computer.
- The port and password of the InfraTools Remote Control Agent are identical to those in the **sendscan.scn** scenario mapping.
- The name of the remote computer is the same at the level of the InfraTools Remote Control Manager, and in the **Name** field (Name) in the **Computers** table (amComputer).
- The TCP/IP layers are correctly installed on the remote computer.

To scan computer that are not running 32-bit Windows

The **Send the scanner to the computers** module only enables you to scan 32-bit Windows computers. To scan computers running other platforms, you must manually use one of the scanners installed by the **Automatic inventory** package, provided on the AssetCenter CD-ROM.

These scanners are created in the **bin32\scanners** subfolder of the Connect-It installation folder. The names of these scanners enable you to identify the platform for which they are intended.

To find out what is inventoried by the scanners

The **Automatic inventory** package, installed from the AssetCenter CD-ROM installs the **settings.txt** file in the **bin32\scanners** subfolder of the Connect-It installation folder.

This file contains a description of what the scanners, located in the same subfolder, inventory during their execution.

Get the results of the scanners module (GetFsf)

AssetCenter Server enables you to program the recovery of the **.fsf** files produced by InfraTools Desktop Discovery (these files store the results of the machine scan).

The list of machines on which the **.fsf** files are recovered is defined by taking all the records from the **Computers** table (amComputer) whose **Last hardware scan** field (dtHardScan) is inferior to the **Last scan** field (dtLastScan).



Note:

This module is based on the assumption that the machine scan has already been performed using the **Send the scanner to the computers** module.

Before activating the **Get the results of the scanners** module, you need to verify that the following parameters are correct:

- Parameters of the AssetCenter Server module **Get the results of the scanners**.
- Parameters of the Connect-It **getfsf.scn** scenario, which is located in the **scenario\autoscan\ac40** sub-folder of the Connect-It installation folder.

Parameters of the module **Get the results of the scanners (User data item field).**

Here is the parameter line by default:

```
"$connectit_exedir$/conitsvc.exe" -once
'$connectit_exedir$/../scenario/autoscan/ac40/getfsf.scn'
-d:AssetCenter.SERVER=$cnx$ -d:AssetCenter.LOGIN=$login$
-d:AssetCenter.TEXTPASSWORD=$pwd$
```

Information about certain parameters in this line:

- `$connectit_exedir$` stores the path of the **conitsvc.exe** program in the Windows Registry.

You do not need to modify this parameter.

- **-once** indicates that **conitsvc.exe** must be executed one time (in other words, using Connect-It's **Once** scheduler.).

Do not modify this parameter; you define the programming at the level of the AssetCenter application.

- `$connectit_exedir$/../scenario/autoscan/ac40/getfsf.scn` is the path to the Connect-It scenario to use..

Modify this parameter if you want to use another Connect-It scenario.

- `-d:AssetCenter.SERVER=cnx`
`-d:AssetCenter.LOGIN=$login$`
`-d:AssetCenter.TEXTPASSWORD=pwd` stores the AssetCenter connection name opened by AssetCenter Server, as well as the login and password used for the connection.

These parameters overwrite the values defined at the level of the AssetCenter connector in the **getfsf.scn** scenario.

You have no need to modify these parameters.

Parameters of the **getfsf.scn** Connect-It scenario

To modify the **getfsf.scn** scenario:

- 1 Execute the Connect-It scenario editor.

- 2 Open the **getfsf.scn** scenario, which is located in the **scenario\autoscan\ac40** sub-folder of the Connect-It installation folder.
- 3 Select the **Fet FSF** connector in the **Scenario diagram** window by clicking on the title bar of the **Get FSF** box; not inside of the box itself.
- 4 Select the **Tools/ Edit mapping** menu.
- 5 Select the **amComputerSrc-CommandDst/ Command (CommandDst)** line.
- 6 Click the **magnifier**.
- 7 Select the **Arg** element.
- 8 Here is the default mapping script:

```
Dim strArg As String
strArg = "-cmd:getfile -certiffile:iftscan.cfg
-certifpwd:icwpwd"
strArg = strArg & " -ipaddr:" & [TcpIpHostName]
strArg = strArg & " -ipport:738"
strArg = strArg & " -pwd:password"
strArg = strArg & " -remfile:c:/" & [WorkGroup] & "_" &
[Name] & ".fsf"
strArg = strArg & " -locfile:../fsf/" & [WorkGroup] & "_"
& [Name] & ".fsf"
```

Here are some indications on the script parameters:

- **-certiffile:iftscan.cfg -certifpwd:icwpwd** These files are installed in the **bin32/scanners** subfolder of the Connect-It installation folder by the **Automatic inventory** package, supplied on the AssetCenter CD-ROM. They go with the **scanw32.exe** inventory program. They grant you the rights necessary to scan the remote computer.
- **-ipport:738** and **-pwd:password**: These parameters indicate the port and the password of the InfraTools Remote Control agents on the remote computers. Check that the agents on the remote computers are using this port and password or modify these parameters accordingly. If you want to use a different port and

password on certain remote computers, modify the mapping script. The script is written in Basic.

- `-remfile`: This parameter indicates the full path of the `.fsf` files to recover from the remote computers. By default, you'll notice that these files are created at the root `c:\`, and that the names of these files are made up of the domain name, followed by the names of the remote computers.
- `-locfile`:: This parameter defines the full path of the `.fsf` files, copied onto the computer running Connect-It. By default, these files are created in the `\fsf` subfolder of the Connect-It installation folder, and that the names of these files are composed of the domain name followed by the names of the remote computers.

 **Warning:**

This mapping script must perfectly match the mapping script of the `sendscan.scn` scenario.

To find out which information is populated in the AssetCenter database, examine in detail the `getfsf.scn` scenario.

For more information about the **Command line** connector, refer to the *Connect-It User's guide*, chapter **Protocol type connectors**, section **Command line connector**.

For more information about the **AssetCenter** connector, refer to the *Connect-It User's guide*, chapter **Peregrine Systems connectors**, section **Asset Management connector**.

To obtain a documentation of the `getfsf.scn` scenario, refer to the *Connect-It User's guide*, chapter **Scenario documentations**, section **Creating a scenario documentation**.

Troubleshooting

If you have any difficulties in executing this module, make sure of the following:

- An InfraTools Remote Control Agent is installed on the remote computer.

- The port and password of the InfraTools Remote Control Agent are identical to those in the **sendscan.scn** scenario mapping.
- The name of the remote computer is the same at the level of the InfraTools Remote Control Manager, and in the **Name** field (Name) in the **Computers** table (amComputer).
- The TCP/IP layers are correctly installed on the remote computer.

Update the database with the results of the scanners module (IddAc)

AssetCenter Server enables you to program the recovery of the **.fsf** files produced by InfraTools Desktop Discovery (these files store the results of the machine scan).

The folder containing the **.fsf** files is specified at the level of the Connect-It **iddac.scn** scenario.

Note:

This module is based on the assumption that the machine scan has already been performed.

Before activating the **Update the database with the results of the scanners** module, you need to verify that the following parameters are correct:

- Parameters of the AssetCenter Server module **Update the database with the results of the scanners**.
- Parameters of the Connect-It **iddac.scn** scenario, which is located in the **scenario\idd\iddac40** sub-folder of the Connect-It installation folder.

Parameters of the module Update the database with the results of the scanners (User data item field).

Here is the parameter line by default:

```
"$connectit_exedir$/conitsvc.exe" -once  
'$connectit_exedir$/../scenario/idd/iddac40/iddac.scn'
```

```
-d:AssetCenter.SERVER=$cnx$ -d:AssetCenter.LOGIN=$login$
-d:AssetCenter.TEXTPASSWORD=$pwd$
```

Information about certain parameters in this line:

- `$connectit_exedir$` stores the path of the **conitsvc.exe** program in the Windows Registry.
You do not need to modify this parameter.
- **-once** indicates that **conitsvc.exe** must be executed one time (in other words, using Connect-It's **Once** scheduler.).
Do not modify this parameter; you define the programming at the level of the AssetCenter application.
- `$connectit_exedir$/../scenario/idd/iddac40/iddac.scn` is the path to the Connect-It scenario to use..
Modify this parameter if you want to use another Connect-It scenario.
- `-d:AssetCenter.SERVER=cnx`
`-d:AssetCenter.LOGIN=$login$`
`-d:AssetCenter.TEXTPASSWORD=pwd` stores the AssetCenter connection name opened by AssetCenter Server, as well as the login and password used for the connection.
These parameters overwrite the values defined at the level of the AssetCenter connector in the **iddac.scn** scenario.
You have no need to modify these parameters.

Parameters of the **iddac.scn** Connect-It scenario

To modify the **iddac.scn** scenario:

- 1 Execute the Connect-It scenario editor.
- 2 Open the **iddac.scn** scenario, which is located in the **scenario\idd\iddac40** sub-folder of the Connect-It installation folder.
- 3 Select the **Desktop Discovery** connector in the **Scenario diagram** window by clicking on the title bar of the **Desktop Discovery** box; not inside of the box itself.
- 4 Select the **Tools/ Configure** menu.
- 5 Click **Next**.

6 Populate the following fields:

- **FSF files folder** field: the folder path that contains the **.fsf** files to get (machine scan files).
This folder must be the same as the one specified in the **getfsf.scn** scenario **mapping**.
- **SAI files folder**: the folder path that contains the **.sai** files (correspondence files between the scanned software files and the corresponding software).
This folder must be the same as the one specified in the **getfsf.scn** scenario **mapping**.
- **User entry file**: the path of the **.cdt** file. (This Connect-It file lists of fields present in the InfraTools Desktop Discovery scan formula in order that they be present in the Connect-It mapping.)

To find out which information is populated in the AssetCenter database, examine in detail the **iddac.scn** scenario.

The module populates the **Last hardware scan** (dtHardScan), **Last network scan** (dtNetworkScan) and **Last software scan** (dtSoftScan) fields in the **Computers** table (amComputer) with the date of the scan.

For more information about the **Desktop Discovery** connector, refer to the **Connect-It User's guide**, chapter **Peregrine Systems connectors**, section **InfraTools Desktop Discovery connector**.

For more information about the **AssetCenter** connector, refer to the **Connect-It User's guide**, chapter **Peregrine Systems connectors**, section **Asset Management connector**.

To obtain a documentation for the **iddac.scn** scenario, refer to the **Connect-It User's guide**, chapter **Scenario documentations**, section **Creating a scenario documentation**.

Verify null-identifier records module (History)

This module verifies integrity of the records whose primary keys are null.

These records are automatically created in all the tables when the database is created.

They are used by AssetCenter to perform certain administrative tasks (which is transparent to you).

This module verifies that these records still exist, and will recreate them if necessary.

We recommend that you execute this module at least once every day to maintain the integrity of the database.

Purge the input-events table module (PurgeEventInTable)

This module deletes the records of the **Input events** table (amInputEvent) according to the information in the:

- **Status** field (seStatus) of the **Input events** table (amInputEvent).
- **Deletion** field (seStatus) of the **Input events** table (amInputEvent).
- Expiration time defined by the **Events management/ Expiration time for input events (in hours)**, accessible via the **Administration/ Database options** menu in the AssetCenter application.

Purge the outgoing-events table module (PurgeEventOutTable)

This module deletes the records from the **Input events** table according to the information in the:

- **Status** field (seStatus) of the **Output events** table (amOutputEvent).
- **Deletion** field (seStatus) of the **Output events** table (amOutputEvent).
- Expiration time defined by the **Events management/ Expiration time for output events (in hours)**, accessible via the **Administration/ Database options** menu in the AssetCenter application.

Update statistics for tables module (Stats)

This module updates the database statistics.

These statistics are used by all the DBMSs supported by AssetCenter to optimize SQL query plans.

If these statistics are not updated, the DBMS will not know which indexes are the most efficient.

We recommend that you execute this module once a week, or every night if your database is heavily modified.

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 under AssetCenter Server.

Indicate which verifications to carry out by checking the appropriate boxes.

Administrating 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 a service

During the installation of AssetCenter:

- AssetCenter Server is installed as a service that is not launched.
- 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.)

 **Warning:**

You need to have rebooted the computer on which AssetCenter Server is installed after installing the DBMS client layers of the base to which the AssetCenter Server service is connected. This is because the Windows service control manager reads the **PATH** environment variable when you start the computer. The AssetCenter Server service, however, needs an updated **PATH** variable in order to find the DLLs of the DBMS. This is especially true if the AssetCenter installation program installs SQL Anywhere runtime and if you want to run the AssetCenter Server service on a demonstration database (which uses SQL Anywhere).

Enabling Web access to the AssetCenter Server service

To enable Web access:

- Select a connection to a database.
- Select the **Use this connection in service mode** option.

You can also:

- Edit **amsrvcf.ini**, which can be found in the **\amsrv\bin** 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.
-

 **Important:**

In Unix, the value of the port that is used must be superior to 1024. This is due to inherent reasons of security in the operating system.

Start the AssetCenter Server service

To start the AssetCenter Server NT service:

- 1 In Control Panel, click the **Services** utility.
- 2 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:

- 1 Click the **Startup** button.
- 2 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

 **Note:**

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:

- 1 Launch your usual Web browser.
- 2 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".

- 3 The Home Page is opened. Click the **Connect** button in this page.
- 4 A window authorizing access to the AssetCenter Server service is displayed. Use it to enter:
 - 1 A "UserName": "WebAdmin".
 - 2 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:

- 1 Check the **Reconnect at startup** option.
 - 2 Enter the name of the database connection to which the AssetCenter Server service must reconnect automatically.
 - 3 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 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.

13 | Messaging

CHAPTER

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 details the configuration required to use the messaging system depending on the protocol used.

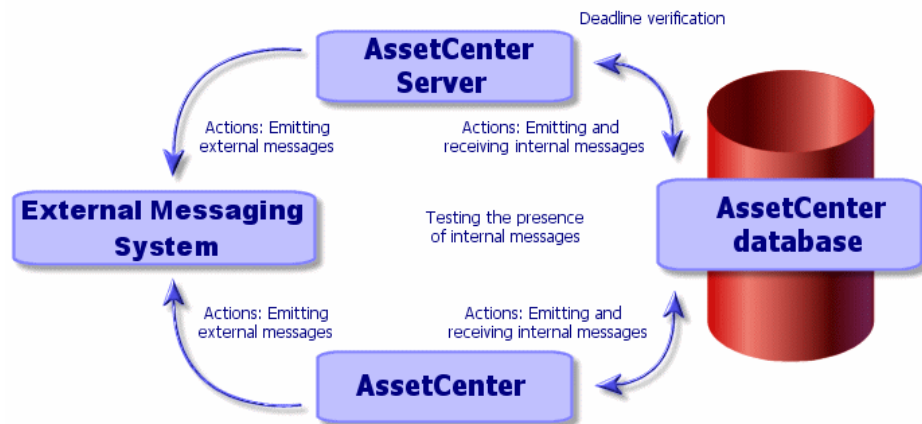
Overview of messaging

AssetCenter manages message-sending by using the following protocol types:

- AM (AssetCenter)
- SMTP
- MAPI
- VIM

When receiving messages, AssetCenter only manages those messages using the AM (AssetCenter) protocol.

Figure 13.1. Messaging system - overview



Refer to the **Advanced use** guide, section **Messaging**, for information about how to send, consult and receive messages.

Configuring AssetCenter to use messaging systems

The configuration of AssetCenter depends of the type of protocols used. 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.
- Populate certain fields of an action.
- Configure and activate AssetCenter Server.
- Configure the frequency of testing for new messages.
Using the **Edit/ Options** menu, **Messages** tab.

Warning:

When you use AssetCenter, you cannot use another messaging protocol other than the one already used.

Only the internal AssetCenter (AM) messaging protocol can be jointly used with another messaging protocol.

Multiple recipients

Whatever protocol you use, you have to respect the following syntax in order to send a message to several recipients:

```
SMTP:[name@adresse.domain], SMTP:[name2@address.domain]
```

For example:

```
SMTP:[andreoucassidou@peregrine.com],  
SMTP:[administrator@prgn.com]
```

SMTP protocol

Detail of the user

In order to send a message, AssetCenter needs to know the account of the sender, specified in the Departments and Employees table (**Messaging** tab), and the messaging address of the recipient, specified in an action.

- The account of the sender is identified by the following fields:
 - **Account** (SQL name: MailLogin): this must be in the form:

```
SMTP:[ name ]
```

- **Password** (MailPassword): Leave this field empty, unless your SMTP server requires a login.

The screenshot shows the 'Messaging' tab of a user profile form. The 'Name' field is 'Hartke' and 'First' is 'Richard'. The 'Title' is 'Building Maintenance' and 'Mr/Mrs' is 'Mr.'. The 'Department' is 'Administration'. Below the tabs, the 'Account' field is 'SMTP:hartke' and the 'Password' field is masked with asterisks.

- Populate the **Email** field of the **General** tab with the following syntax:

SMTP : [name@address . domain]

The screenshot shows the 'General' tab of a user profile form. The 'Address' section includes 'Location' (/San Mateo site/Building 02/2nd Floor/001- Office/), 'Telephone' ((650) 572-9000), 'Fax' ((650) 572-9099), 'Mobile tel.', and 'Home tel.'. The 'E-Mail' field is 'SMTP:hartke@taltek.com'. The 'ID #' is 'DEMO-M031' and 'Cost center' is 'Common Line'. The 'Hire date' is '21/12/1995' and 'Bar code' is 'DEMO-U004'. There are also fields for 'Leaving date', 'Field 1', 'Field 2', and 'Field 3', and a 'Remarks' text area.

- The recipient is identified by the **To** field (SQL name: MsgTo) in the detail of a messaging-type action.

Populate the **To** field (and the optional **Cc** and **Ccc** fields, if you wish) with the appropriate address, in the following manner:

SMTP : [name@address . domain]

Or using a calculated string.

The screenshot shows a user detail window with the following fields and values:

- Name: Email
- Context: Employees (anEmpDept)
- Type: Messaging
- Referenced object: (empty)
- Priority: Normal
- Acknowledgment: (unchecked)
- To: [EMail]
- Cc: (empty)
- Bcc: (empty)
- Subject: (empty)
- Message: (empty)

In this example, the **To** field is populated with the contents of the **Email** field, in the **General** tab of a user's detail. The administrator is put in copy.

Win.ini file

You must add certain command lines in the **win.ini** file of your Windows folder to be able to use the messaging system in AssetCenter.

```
[mail]
SMTP=1
SMTPserver=[server name]
email=[messaging address@domain name]
displayname=[user's full name]
```

Example

```
[mail]
SMTP=1
SMTPserver=mail.prgn.com
email=sblaine@prgn.com
displayname=Steven Blaine
```

MAPI protocol

Detail of the user

In order to send a message, AssetCenter needs to know the account of the sender, specified in the Departments and Employees table (**Messaging** tab), and the messaging address of the recipient, specified in an action.

- The profile of the sender is identified by the following fields:
 - **Account** (SQL name: MailLogin): Populate the field in the following manner:

```
MAPI:[user profile name]
```

To find out the name of the user profile, look in the Windows **Configuration Panel** and click the **Mail** icon. In the window that appears, click **Show Profiles**.

- **Password:** (MailPassword): Populate the field with your messaging password.

The screenshot shows a user profile configuration window. At the top, there are fields for Name (Hartke), First (Richard), Title (Building Maintenance), and Mr/Mrs (Mr.). Below these is a Department field (Administration). A tabbed interface is visible with tabs for Training, Portfolio, Costs, Projects, Profile, Messaging, and Appli. The Messaging tab is active, showing an Account field with the value 'MAPI:MS Exchange Settings' and a Password field with masked characters '*****'.

- Populate the **Email** field of the **General** tab with the following syntax:

MAPI:[mailbox name]

General Training Portfolio Costs Projects Profile Messag

Address

Location: /San Mateo site/Building 02/2nd Floor/001- Office/

Telephone: (650) 572-9000 Fax: (650) 572-9099

Mobile tel.: Home tel.:

E-Mail: MAPI:gcolombo

Field 1:

Field 2:

ID #: DEMO-M031 Cost center: Common Line

Hire date: 21/12/1995 Bar code: DEMO-U004

Leaving date: Field 3:

Remarks:

- The recipient is identified by the **To** field (SQL name: MsgTo) in the detail of a messaging-type action.

Populate the **To** field (and the optional **Cc** and **Ccc** fields, if you wish) with the appropriate address, in the following manner:

MAPI:[mailbox name]

Or using a calculated string.

Name: Email

Context: Employees (anEmplDept)

Type: Messaging Preview...

Description Messaging Features History

Referenced object:

Priority: Normal

Acknowledgment

To: [EMail]

Cc: MAPI:administrator

Bcc:

Subject: ✓ ✗

Message:

In this example, the **To** field is populated with the contents of the **Email** field, in the **General** tab of a user's detail.

VIM protocol



Note:

AssetCenter does not manage accented characters for the sender's identifier with the VIM protocol.

Detail of the user

In order to send a message, AssetCenter needs to know the account of the sender, specified in the Departments and Employees table (**Messaging** tab), and the messaging address of the recipient, specified in an action.

- The account of the sender is identified by the following fields:
 - **Account** (SQL name: MailLogin): this must be in the form:

VIM: [name/domain]

The screenshot shows a user detail form with the following fields and values:

- Name: Listel
- First: Susan
- Title: Switchboard Manager
- Mr/Mrs: Mrs
- Department: Administration

The **Messaging** tab is selected, showing:

- Account: VIM:slistel/TALTEK
- Password: *****

- **Password:** (MailPassword): Populate the field with your messaging password.
- Populate the **Email** field (SQL name: Email) of the **General** tab with the following syntax:

```
VIM: [name/domain]
```

The screenshot shows a software window with several tabs: General, Training, Portfolio, Costs, Projects, Profile, and Messag. The 'General' tab is active. The form contains the following fields and values:

- Location: /San Mateo site/Building 01/Hall/001 - Reception/
- Telephone: (650) 572-9000
- Fax: (650) 572-9099
- Mobile tel.:
- Home tel.:
- EMail: VIM:slistel/TALTEK
- Field 1:
- Field 2:
- ID #: DEMO-M041
- Cost center: Common Line
- Hire date: 21/12/1995
- Bar code: DEMO-U005
- Leaving date:
- Field 3:
- Remarks:

- The recipient is identified by the **To** field in the detail of a messaging-type action.

Populate the **To** field (and the optional **Cc** and **Ccc** fields, if you wish) according to the following syntax:

VIM: [name/domain]

The screenshot shows a user detail page for an email. The 'Name' field is 'Email', 'Context' is 'Employees (amEmplDept)', and 'Type' is 'Messaging'. The 'To' field is populated with '[EMail]'. The 'Cc' field is 'VIM:administrator/PRGN'. The 'Subject' and 'Message' fields are empty.

In this example, the **To** field is populated with the contents of the **Email** field, in the **General** tab of a user's detail. The administrator is put in copy.

Windows configuration

You must configure the Windows **Path** in order for AssetCenter to be able to manage the VIM protocol.

Windows 2000

- Note the access path to the **vim32.dll**.
- Edit the properties of your computer by right-clicking the icon of your workstation (system).
- In the **Advanced** tab, click **Environment variables**.
- In the **System variables**, edit the **Path** variable by adding the Lotus Notes path.

Windows NT4

- Note the access path to the **vim32.dll** file of your Lotus folder.

- Edit the properties of your computer by right-clicking the icon of your workstation (system).
- In the **Environment** tab, edit the **Path** variable by adding the Lotus Notes path.

Windows 95, 98, ME

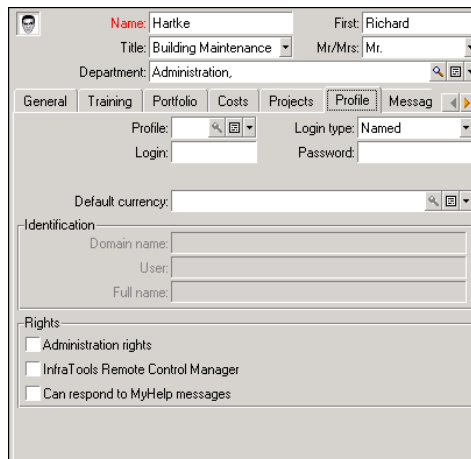
- Note the access path to the **vim32.dll** file of your Lotus folder.
- Locate and edit the autoexec.bat file at the root of your hard drive.
- Add the instruction.

```
SET PATH=[Lotus Notes path]
```

Then specify the access path to the **vim32** of your Lotus Notes folder.

AM protocol

- In order for an internal message to be sent, the user must be registered in the Departments and Employees table (amEmplDept) and have a login:
 - The login of the user is specified in the Departments and Employees table (**Profile** tab).



The screenshot displays a user profile configuration window. At the top, there are fields for Name (Hartke), First (Richard), Title (Building Maintenance), and Mr/Mrs (Mr.). Below these is the Department (Administration). The window has several tabs: General, Training, Portfolio, Costs, Projects, Profile, and Message. The Profile tab is currently selected, showing fields for Profile, Login type (Named), Login, and Password. There are also fields for Default currency, Identification (Domain name, User, Full name), and Rights (Administration rights, InfraTools Remote Control Manager, Can respond to MyHelp messages).

- You must also populate the **Email** field in the **General** tab of the user with the following syntax:

```
AM:[user login]
```

- The messaging address of the recipient, specified in a messaging-type action, must be entered:
- Populate the **To** field (and the optional **Cc** and **Ccc** fields, if you wish) in the following manner:

```
AM:[login of recipient]
```

Or using a calculated string.


The screenshot shows the 'Email' field in the 'General' tab of a user's detail page. The field is populated with the text '[E-Mail]'. The 'Context' is set to 'Employees (anEmpDept)' and the 'Type' is 'Messaging'. The 'Referenced object' is empty, and the 'Priority' is 'Normal'. The 'Acknowledgment' checkbox is unchecked. The 'To' field is populated with '[E-Mail]', the 'Cc' field with 'AM:Brodin', and the 'Bcc' field is empty. The 'Subject' and 'Message' fields are also empty.

In this example, the **To** field is populated with the contents of the **Email** field, in the **General** tab of a user's detail.

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

- 1 Launch AssetCenter Server.
- 2 Connect to a database.
- 3 Click .

Test to perform if a problem is encountered

- 1 Create a new **Messaging Type** (SQL name: seActionType) action specifying a given recipient.
- 2 Trigger the action using the **Tools/ Actions** menu.
- 3 Verify that the recipient has received the message and that the "router" has not sent an error message to your messaging system (unknown recipient).
- 4 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 (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.

 **Warning:**

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"

A password is required with "VIM" messaging systems. Add this password under your messaging system. Then indicate it in AssetCenter in the **Password** field (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 (MailPassword), in the **Messaging** tab of the employee detail.

"Message container is corrupt" or "The configuration of your station is invalid".

The VIM protocol recovers the information linked to the name and the location of the Lotus Notes **notes.ini** file. If this information is incorrect, you cannot send messages. Edit and modify the parameters in this file accordingly.

14 | Using AssetCenter as a DDE server

CHAPTER

This section describes in detail the DDE calls recognized by AssetCenter when used as a DDE server.

In this section, each theoretical description is followed by a practical example.

Definition of a DDE server

DDE means Dynamic Data Exchange and designates dynamic data exchange mechanisms between Windows applications. In the case being described, DDE is used to execute AssetCenter commands from another application

DDE call mechanisms

The DDE mechanisms are based on "services". In order to execute a DDE mechanism, you need to define a "topic" to give the necessary

context in which the "commands" are to be executed. For reasons of integrity, each time you change context, you must terminate the previous context.

This chapter contains information about the following topics:

- DDE service
- DDE topic
- DDE command

DDE service

In most cases, the "service" is the name of the executable loaded in memory. When using AssetCenter as a DDE server, the service is "aam".

DDE topic

The topic allows you to define the context in which the action is to be executed. For AssetCenter, this topic is "AssetCenter".

DDE command

These are commands that will be sent to AssetCenter to be executed. They can be divided into several groups:

- Global commands, which do not require a table name or field name to be executed.
- Commands associated with a table, requiring the SQL name of a table as a parameter in order to be executed.
- Commands associated with a table and field or link, requiring the SQL name of a table and field or link as parameters in order to be executed.

Commands belonging to these groups can be of two types:

- "Execute", which enables you to execute a task in AssetCenter.
- "Request", which enables you to ask AssetCenter for information.

How to find the SQL name of a table, field or link

When you right-click any field of a table, AssetCenter displays a shortcut menu. If you select the **Configure object** menu, the window displayed by AssetCenter gives the SQL name the table and SQL name of the object (link or field) on which you have clicked.

Introduction to DDE commands

Steps to follow

Three steps are necessary for the proper execution of a DDE command:

- 1 Clearly define the context of the command's execution by specifying the "Service" and the "Topic" to be used. When using AssetCenter as a DDE server, the "Topic" is always "AssetCenter".

 **Note:**

Once the context is defined, it is used by default in all following DDE commands until a new context is defined.

- 2 Launch the command itself. Two types of command are available.
 - **Execute:** <command>(<parameters>)
 - **Request:** <command>(<parameters>)
- 3 Close the previously defined context.

Particularities

Each application under Windows has its own way of sending and receiving DDE commands. The following sections of this document include:

- An exhaustive list of DDE commands that AssetCenter can receive. The syntax of each of these commands is described in detail.

- Examples of how AssetCenter can be driven by DDE mechanisms, which illustrate the use of these commands. Each of these examples uses a different programming language.

Global commands

"Global" commands do not depend on a given table or field. In particular, they do not require the SQL name of a table or a field as an argument. This chapter contains information about the following commands:

- Connect(Cnx, User, Password)
- Disconnect()
- ExecuteAction(ActionName)
- ListAllTables([Mask])
- ListOpenTables([Mask])
- OpenView(ViewName)

Connect(<Cnx>, <User>, <Password>)

Type of action

Execute

Description

Connects to a database using the following parameters:

<Cnx>

This argument can either contain:

- A connection name defined under AssetCenter (which can be found in the **amdb.ini** file).
- The full definition of a connection to a database according to the syntax described below:

```
[<Database engine>;<Database localization>;<Login for the
database engine>;<Password for the database engine>]
```

<User>

This argument contains the name of the AssetCenter user that you want to use to connect to the database.

<Password>

This argument contains the password associated with the login (value of "<User>" argument).

Examples

The following command allows you to connect to an Oracle database for which a connection is already defined under AssetCenter. The name of this connection is "TDemo". The "Admin" login is used for the connection. The password is "password".

```
Execute:Connect(TDemo, Admin, password)
```

The following command performs the same connection without the connection being defined under AssetCenter. The "TDemo" database is located on the server "Joshua". The password for the Oracle connection is "Root".

```
Execute:Connect([Oracle;Joshua;TDemo;Root], Admin, password)
```

Disconnect()

Type of action

Execute

Description

Disconnects AssetCenter from the current database.

Example

The following command terminates the connection to the AssetCenter database:

```
Execute:Disconnect()
```

ExecuteAction(<ActionName>)

Type of action

Execute

Description

Triggers the action called "<ActionName>".

<ActionName>

This argument contains the name of the action as defined under AssetCenter in the **Name** field (SQL name: Name) of the action detail.

Example

The following command triggers the action "Reminder: work order not completed":

```
Execute:ExecuteAction(Reminder: Work order not completed)
```

ListAllTables([Mask])

Type of action

Request

Description

Lists all tables in the database. This list, composed of the SQL names of tables, can be filtered by using the "<Mask>" argument.

<Mask>

This argument allows you to filter data using the following:

- A question mark ("?") can be used as a wildcard for any single character.
- An asterisk ("*") can be used to represent any character or group of characters.

Examples

The following command returns the list of SQL names of all the tables in the current database:

```
Request:ListAllTables()
```

The following command returns the list of SQL names of all the tables in the current database whose SQL names start with "amA":

```
Request:ListAllTables(amA*)
```

The following command returns the list of SQL names of all the tables in the current database containing the letter "v":

```
Request:ListAllTables(*v*)
```

The following command returns the list of SQL names of all the tables in the current database starting with "am" and whose fourth character is "t":

```
Request:ListAllTables(am?t*)
```

ListOpenTables([Mask])

Type of action

Request

Description

Lists the SQL names of all open tables of the database. This list can be filtered using the "<Mask>" argument.

<Mask>

This argument allows you to filter data using the following:

- A question mark ("?") can be used as a wildcard for any single character.
- An asterisk ("*") can be used to represent any character or group of characters.

Examples

The following command returns the SQL names of all open tables in the current database:

```
Request:ListOpenTables()
```

The following command returns the SQL names of all open tables in the current database whose SQL name starts with "amA":

```
Request:ListOpenTables(amA*)
```

Let's suppose that the three following tables are open under AssetCenter: "amAsset", "amAction", "amModel". The previous command returns the SQL name of the two tables beginning with "amA", i.e.: "amAsset" and "amAction".

OpenView(<ViewName>)

Type of action

Execute

Description

Opens a view defined under AssetCenter.

<ViewName>

This argument contains the SQL name of the view as it is defined under AssetCenter.

Example

The following command opens the view called "Leased assets":

```
Execute:OpenView(Leased assets)
```

Commands associated with a table

These commands depend on a table. They require the SQL name of a table as an argument in order to be executed.

This chapter includes information about the following commands:

- OpenTable(Table)
- CloseTable(Table)
- Table.GetRecordCount()
- Table.SetViewMode(Mode)
- Table.SetRecordMode(Mode)
- Table.ListAllFields([Mask])
- Table.ListAllLinks([Mask])
- Table.SetFilter(Condition)
- Table.SetSelection(Condition)
- Table.GetSelectionId()

OpenTable(<Table>)

Type of action

Execute

Description

Opens the table with SQL name "<Table>".

<Table>

This argument contains the SQL name of the table to be opened.

Example

The following command opens the table of assets (SQL name: amAsset):

```
Execute:OpenTable(amAsset)
```

CloseTable(<Table>)

Type of action

Execute

Description

Closes the table previously opened under AssetCenter.

<Table>

This argument contains the SQL name of the table to be closed.

Example

The following command closes the table of assets (SQL name: amAsset):

```
Execute:CloseTable(amAsset)
```

<Table>.GetRecordCount()

Type of action

Request

Description

Returns the number of records in the table with SQL name "<Table>". The table concerned by this command must first be opened in order for this command to work.

<Table>

This argument contains the SQL name of the table for which you want to know the number of records.

Example

The following command returns the number of records in the table of asset (SQL name: amAsset):

```
Request : amAsset . GetRecordCount ( )
```

<Table>.SetViewMode(<Mode>)**Type of action**

Execute

Description

Defines the view mode of a table that has been opened.

<Table>

This argument contains the SQL name of the table concerned.

<Mode>

This argument can be set to one of the following values:

- "Arbo": Records in the table "<Table>" are displayed in tree view.
- "List": Records in the table "<Table>" are displayed in list view.
- "ListOnly": Only the records from the table "<Table>" are displayed.
- "DetailOnly": Only the detail of the selected record in the table "<Table>" is displayed.
- "ListDetail": Displays both the list of records in the table "<Table>" and the detail of the selected record in this list.

Example

This following command puts the table of products (SQL name: amCatProduct) in tree-view:

```
Execute:amCatProduct.SetViewMode(Arbo)
```

<Table>.SetRecordMode(<Mode>)

Type of action

Execute

Description







Defines the mode of interaction with records from an open table.



<Table>

This argument contains the SQL name of the table concerned.

<Mode>

This argument can be set to one of the following values:

- "New": Starts the creation of a new record in the table "<Table>". The command corresponds to clicking the  button.
- "Duplicate": Duplicates the selected record in the table "<Table>". The command corresponds to clicking the  button.
- "Delete": Destroys the record selected in the table "<Table>". The command corresponds to clicking the  button.
- "Modify": Confirms the modifications made to the selected record in the table "<Table>". The command corresponds to clicking the  button.
- "Create": Confirms the creation of a new record in the table "<Table>". The command corresponds to clicking the  button.
- "CreateContinue": Combines a creation and a duplication. The command corresponds to clicking the  button.

- "Cancel": Cancels the creation of a new record or the modifications made to the selected record. The command corresponds to clicking the  button.
- "Close": Closes the previously opened table "<Table>". The command corresponds to clicking the  button.

Example

The following example opens the table of assets (SQL name: amAsset), starts the creation of a new record, then cancels this creation:

```
Execute:OpenTable(amAsset)
Execute:amAsset.SetRecordMode(New)
Execute:amAsset.SetRecordMode(Cancel)
```

<Table>.ListAllFields([Mask])

Type of action

Request

Description

Returns the SQL names of all the fields in the previously opened table "<Table>".

<Table>

This argument contains the SQL name of the table concerned.

<Mask>

This argument allows you to filter data using the following:

- A question mark ("?") can be used as a wildcard for any single character.
- An asterisk ("*") can be used to represent any character or group of characters.

Example

The following command returns the SQL names of all the fields in the table of assets:

```
Request:amAsset.ListAllFields
```

The following command returns the SQL names of all the fields in the table of assets (SQL name: amAsset) whose SQL name starts with "se":

```
Request:amAsset.ListAllFields(se*)
```

<Table>.ListAllLinks([Mask])

Type of action

Request

Description

Returns the SQL names of all the links in the previously opened table "<Table>".

<Table>

This argument contains the SQL of the table concerned.

<Mask>

This argument allows you to filter data using the following:

- A question mark ("?") can be used as a wildcard for any single character.
- An asterisk ("*") can be used to represent any character or group of characters.

Example

The following command returns the list of SQL names of all the links of the table of assets (SQL name: amAsset):

```
Request: amAsset.ListAllLinks
```

The following command returns the list of SQL names of all the links of the table of assets (SQL name: amAsset) whose SQL name starts with "se":

```
Request: amAsset.ListAllLinks(se*)
```

<Table>.SetFilter(<Condition>)

Type of action

Execute

Description

Filters the table "<Table>" according to the "<Condition>" argument.

<Table>

This argument contains the SQL name of the table concerned.

<Condition>

This argument contains the condition applied to the command. It is an AQL clause.

Example

The following command filters the table of assets (SQL name: amAsset). This filter displays only those records modified before 8/28/98 at 15:00:00:

```
Execute: amAsset.SetFilter(dtLastModif<[98/08/28 15:00:00])
```

<Table>.SetSelection(<Condition>)

Type of action

Execute

Description

Selects one or more records from the previously opened table "<Table>", according to the argument " <Condition>".

<Table>

This argument contains the SQL name of the table concerned.

<Condition>

This argument contains the condition applied to the command. It is an AQL clause.

Example

The following command selects assets whose asset tag is greater than or equal to "7":

```
Execute:amAsset.SetSelection(AssetTag>='7')
```

<Table>.GetSelectionId()

Type of action

Request

Description

Returns the list of identifiers of selected records in the table "<Table>".

<Table>

This argument contains the SQL name of the table concerned.

Example

The following example selects those records in the table of assets (SQL name: amAsset) whose asset tag is greater than or equal to "7", then returns the list of identifiers of the selected records:


```
Execute:amAsset.SetSelection(AssetTag>='7')
Request:amAsset.GetSelectionId()
```

Commands associated with a table and a field or a link

These commands depend on a field in a table. They require the SQL name of a table and the SQL name of a field or a link in this table as argument in order to be executed.

This chapter contains information about the following commands:


- Table-Objet.AddLink()
- Table-Object.GetValue()
- Table-Object.Highlight()
- Table-Object.RemoveLink()
- Table-Object.SetFocus()
- Table-Object.SetValue(Value)
- Table-Link.SetValueWhere(Condition)
- Table-Object.Show()

<Table>:<Object>.AddLink()

Type of action

Execute

Description

Simulates clicking the  button of a list zone. This command allows you to add a link to a record in a table.

<Table>

This parameter contains the SQL name of the table concerned by the operation.

<Object>

This parameter contains the SQL name of the object concerned.

Example

The following command adds a value in an itemized list.

```
Execute:amItemizedList:ItemListVals.AddLink()
```

<Table>:<Object>.GetValue()

Type of action

Request

Description

Returns the value of an "<Object>" (field or link) of the table "<Table>" for the selected record.

<Table>

This argument contains the SQL name of the table concerned.

<Object>

This argument contains the SQL name of the field or link of the table "<Table>" for which you want to recover the value.

Example

The following command returns the values of the **Field1** field (SQL name: Field1) in the table of assets (SQL name: amAsset):

```
Request:amAsset:Field1.GetValue()
```

The following command returns the value of the **Model** link (SQL name: Model) of the table of assets (SQL name: amAsset):

```
Request:amAsset:Model.GetValue()
```

The following command returns the value of the **Remarks** link (SQL name: Comment) of the table of assets (SQL name: amAsset):

```
Request:amAsset:Comment.GetValue()
```

<Table>:<Object>.Hilight()

Type of action

Execute

Description

Sets the focus on a field and highlights it.

<Table>

This parameter contains the SQL name of the table concerned by the operation.

<Object>

This parameter contains the SQL name of the field concerned by the operation. This command does not work for links.

Example

The following command highlights the **Bar code** field (SQL name: Barcode) in the table of assets (SQL name: amAsset):


```
Execute:amAsset:Barcode.Hilight()
```

<Table>:<Object>.RemoveLink()

Type of action

Execute

Description

Simulates clicking the  button of a list zone. This command allows you to delete a link to a record in a table.

<Table>

This parameter contains the SQL name of the table concerned by the operation.

<Object>

This parameter contains the SQL name of the object concerned.

Example

The following command deletes the selected value in an itemized list.

```
Execute:amItemizedList:ItemListVals.RemoveLink()
```

<Table>:<Object>.SetFocus()

Type of action

Execute

Description

Sets the focus on the field or link "<Object>" of the table "<Table>" for the selected record.

<Table>

This argument contains the SQL name of the table containing the field or link to which you want to set the focus.

<Object>

This argument contains the SQL name of the field or link or the table "<Table>" to which you want to set the focus.

Example

The following command sets the focus on the **Model** link (SQL name: Model) of the table of assets (SQL name: amAsset):

```
Request:amAsset:Model.SetFocus()
```

<Table>:<Object>.SetValue(<Value>)

Type of action

Execute

Description

Populates the field "<Field>" in the table "<Table>", for the selected record, with the value "<Value>".

<Table>

This argument contains the SQL name of the table containing the field that you want to populate.

<Field>

This argument contains the SQL name of the field in the table "<Table>" that you want to populate.

<Value>

This argument contains the value you want to assign to the "<Field>" field in the table "<Table>" for the selected record. In the case of a "Date" or "Date+Time" type field, this argument must be expressed using the international date format (yy/mm/dd hh:mm:ss).

Example

The following command attributes the value "Test" to the **Field1** field (SQL name: Field1) in the table of assets (SQL name: amAsset), for the selected record:

```
Execute:amAsset:Field1.SetValue(Test)
```

The following command attributes the value "08/28/00" to the **Install. date** field (SQL name: dInstall) in the table of assets (SQL name: amAsset), for the selected record:

```
Execute:amAsset:dInstall.SetValue(00/08/28)
```

<Table>:<Link>.SetValueWhere(<Condition>)

Type of action

Execute

Description

Populates the link "<Link>" in the table "<Table>", for the selected record, according to the condition "<Condition>".

<Table>

This argument contains the SQL name of the table containing the table you want to populate.

<Link>

This argument contains the SQL name of the link of the table "<Table>" that you want to populate.

<Condition>

This argument allows you to identify the target record of the link. It is an AQL clause.

Example

The following example assigns the value "Test" to the **Model** link (SQL name: Model) from the table of the assets (SQL name: amAsset) for the selected record. The model "Test" must exist in order for the DDE command to execute correctly.

```
Execute:amAsset:amModel.SetValueWhere(Name='Test')
```

<Table>:<Object>.Show()

Type of action

Execute

Description

Moves the focus to a field or a link that is not visible on the screen. The table containing the field or link must be open.

<Table>

This parameter contains the SQL name of the table concerned by the operation.

<Object>

This parameter contains the SQL name of the object concerned.

Example

The following command moves the focus to the **A/C Code** field (SQL name: AcctCode) in the table of assets (SQL name: amAsset):

```
Execute:amAsset:AcctCode.Show()
```

Introduction to Examples of DDE calls

In order to illustrate this area of functionality, we have described several scenarios.

- The first scenario gives a theoretical description of DDE calls.
- The second is a practical example of using DDE calls under Excel. The associated example is written in Visual Basic for Applications.

- The third scenario presents an application written in Visual Basic 5.0 that will enable you to gain some hands-on experience with DDE calls.

First scenario: AssetCenter internal DDE calls

The objective of this scenario is to identify, for each one of the actions to be carried out, the appropriate DDE command and arguments. This scenario deals with the theoretical aspects of using DDE mechanisms; the practical aspects are dealt with in the following examples.

In this example, we are going to create a "numerical" type feature, called "RAM". Here are the actions carried out:

- 1 The table of features is opened;
- 2 The title of the feature is populated.
- 3 The data-entry type of the feature is populated.
- 4 The unit is populated.
- 5 The feature is created.

Answering the call

We must first identify the service and topic needed to execute the DDE commands.

We are in the general context of the AssetCenter application, therefore:

- Service = "aam"
- Topic: "AssetCenter"

We now just have to execute the command that opens the table of features:

- Command: `OpenTable()`
- Parameter: The SQL name of the table; in this case "amFeature"

The command is written as follows:

```
OpenTable(amFeature)
```


AssetCenter opens the table of features. We now need to initiate the creation of a new record for this table:

- Command : SetRecordMode()
- Prefix of the command: The SQL name of the table, i.e. "amFeature"
- Parameter: The date entry mode; in this case "New"

The command is written as follows:

```
amFeature.SetRecordMode(New)
```

Entering data

Next, we need to give AssetCenter the appropriate commands for the fields that interest us:

- **Title** field (SQL name: TextLabel). The command to use and its arguments are as follows:
 - Command : **<Table>:<Object>.SetValue(<Value>)**
 - * Argument "<Table>": The SQL name of the table, i.e. "amFeature"
 - * Argument "<Object>": The SQL name of the field, i.e. "TextLabel"
 - * Argument "<Value>": The value of the field, i.e. "RAM"

```
amFeature:TextLabel.SetValue(RAM)
```

- **Input type** field (SQL name: seDataType). The command to use and its arguments are as follows:
 - Command: **<Table>:<Object>.SetValue(<Value>)**
 - * Argument "<Table>": The SQL name of the table, i.e. "amFeature"
 - * Argument "<Object>": The SQL name of the field, i.e. "seDataType"
 - * Argument "<Value>": The value of the field, i.e. "Numerical"

```
amFeature:seDataType.SetValue(Numeric)
```

- **Unit** field (SQL name: Unit). The command to use and its arguments are as follows:
 - Command: **<Table>:<Object>.SetValue(<Value>)**

- * Argument "<Table>": The SQL name of the table, i.e. "amFeature"
- * Argument "<Object>": The SQL name of the field, i.e. "Unit"
- * Argument "<Value>": The value of the field, i.e. "MB"

```
amFeature:Unit.SetValue(MB)
```

Creating the feature

In order to create the feature, we just need to create the record in the table of features:

- Command : <Table>.SetRecordMode(<Mode>)
 - Argument "<Table>": The SQL name of the table, i.e. "amFeature"
 - Argument "<Mode>": The mode of creation of a record, i.e. "Create"

```
amFeature.SetRecordMode(Create)
```

Second scenario: DDE calls from Excel

This example is based on an Excel worksheet "TestDDE.xls", which can be found in the **amdemo** folder of AssetCenter. This file contains the necessary macros for this example.

This example is identical to the previous one. Only the DDE calls are different and are in conformity with VBA (Visual Basic for Applications).

This section contains the following information:

- Description of the macro
- Source code of macro

Description of the macro

Go to the "Data_entry" worksheet. It contains a table with three columns named **Title**, **Input type**, and **Unit** respectively, and a button (labeled "Create the feature") to which a macro called "Create" has been assigned:

You just need to enter the appropriate values in the **Title** (cell B6 in this example), **Input type** (cell C6), **Unit** (cell D6) fields, and click "Create the feature" in order for Excel to give issue the orders to AssetCenter to perform the following actions:

- 1 Open the table of features.
- 2 Enter the title of the feature.
- 3 Enter the data entry type of the feature.
- 4 Enter the unit.
- 5 Create the feature.

Source code of macro

For reference, here is the source code of the macro that executes the DDE calls. For further information about DDE calls under Excel or Word and the syntax of VBA language, please refer to the respective online documentation.

```
Sub CreateFeature()
Set Label = Worksheets("Data_entry").Range("B6")
Set Type = Worksheets("Data_entry").Range("C6")
Set Unit = Worksheets("Data_entry").Range("D6")
Context = Application.DDEInitiate(app:="aam",
topic:="AssetCenter")
Application.DDEExecute Context, "OpenTable(amFeature)"
Application.DDEExecute Context, "amFeature.SetRecordMode(New)

Application.DDEExecute Context,
"amFeature:TextLabel.SetValue("+Label+)"
Application.DDEExecute Context,
"amFeature:seDataType.SetValue("+Type+)"
Application.DDEExecute Context,
"amFeature:Unit.SetValue("+Unit+)"
Application.DDEExecute Context,
"amFeature.SetRecordMode(Create)"
Application.DDETerminate Context
End Sub
```

Third scenario: DDE calls from Visual Basic

This scenario presents a utility that enables you to experiment with DDE mechanisms via a simple graphical interface. It is also a good illustration of how to program DDE mechanisms under Visual Basic.

For this example, execute **DDE TestCenter.exe**, which can be found by following this path: **samples\DDE\Program**. This program allows you to execute "Execute" and "Request" type DDE commands.

Note:

AssetCenter must be launched in order to be able to receive DDE commands issued by the Basic program.

This chapter contains information about the following topics:

- Before starting
- Executing the program
- Source code of the program

Source code of the program

The commented source code to this program, in the form of a Visual Basic project, can be found by following this path:
sample\DDE\VisualBasic.

Before starting

Installation

In order to use this program, Visual Basic must be correctly installed on your machine. In particular, certain ActiveX controls must be correctly registered. If DDE TestCenter returns an error such as "Control XXXX is not registered", try the following:

- 1 Search for the control on your computer and go to its folder.
- 2 Execute the following command:

```
regsvr32 XXXX
```

- 3 Relaunch DDE TestCenter. If this fails, please refer to the documentation of Visual Basic.

Recommendations

In order to make the most of this example, we recommend that you:

- 1 Launch AssetCenter, then reduce the size of the application window to about half the size of your screen.
- 2 Launch "DDE TestCenter.exe" and move the application window so that is next to that of AssetCenter.

 Note:

In this way, you will be able to see the results of the orders issued by **DDE TestCenter.exe** in AssetCenter directly.

Syntax

This example is similar to the previous one. Only the DDE calls are different and correspond to the Visual Basic standard.


Executing the program

"Execute" type DDE command

Go to the "Execute" tab:

Enter the command to be executed in the "Command" field. Use the following syntax:

```
Command=<command> (<arguments>)
```

Click the  button to execute the command. Any errors are displayed in the "Last DDE Error" field.

Example n°1:

The following "Execute" command opens the table of features:

```
OpenTable(amFeature)
```

Example n°2:

The following "Execute" commands open the table of budgets, create a new record and populate the "Name" field (SQL name: Name) in the detail screen. Execute these commands sequentially:


```
OpenTable(amBudget)
amBudget.SetRecordMode(New)
amBudget.Name.SetValue("Test")
```

"Request" type DDE command

Go to the "Request" tab:

Enter the command to be executed in the "Command" field. Use the following syntax:

```
Command=<command> (<arguments>)
```

Click the  button to execute the command. The results of the request are displayed in the "Request Result" field. Any errors are displayed in the "Last DDE Error" field.

Example n°1:

The following "Request" command displays the list of SQL names of all the tables in the current connection:

```
ListAllTables()
```

Example n°2:

The following "Request" command displays the list of SQL names of all the fields of the table of features (SQL name: amFeature) opened beforehand:

```
amFeature.ListAllFields()
```

15 | Optimizing AssetCenter for use in a WAN environment

CHAPTER

WAN networks are often characterized by:

- Low bandwidth
- High latency

It is possible to configure AssetCenter in order to minimize these disadvantages. However, these configurations are made to the detriment of certain functions of AssetCenter.

This chapter provides some tips that can be used to accommodate the limitations of WAN networks. All the same, it is important to perform tests to measure the trade-off between faster reaction times and loss of functionality.

Options in the Edit/ Options menu

You can limit the length of accesses made to the database using the following options:

- **Type-ahead after** option in the **Navigation** category: You can either deactivate Type-ahead, or attribute a high value (for example, Type-ahead after 10 000 ms).
- **Trees in drop-down lists** option in the **Navigation** category: You can deactivate this option, since tree-views are more costly in terms of performance than list views.

However, you lose the user-friendly nature of tree-views in drop-down lists.

You can limit the exchange of information between the client machine and the database server by adjusting the following options:

- **Do not load for longer than** and **Do not load more than** options in the **Lists** category (main lists or others): We recommend limiting the number of lines loaded (for example, you can specify a maximum value of 50 lines in main lists and 15 in tab lists). It is up to you to determine the number of lines to be loaded - according to any filters you apply to the lists being displayed - and the likelihood of finding the required information in a given number of lines.
- **Test new messages, Messages** category: You can configure this option in order to test for messages on connecting to the database only, or space out the verification intervals (every 10 minutes, for example).
- **Caches** category: You can increase the length of intervals used to define the frequency of refreshing the caches (**Every** column), or even disable cache refreshing during a session. In this case, caches are loaded on connecting to the database only.

It is likely that if you do not refresh the caches on a regular basis, the data being displayed will not be up to date. However, the majority of cached data items are generated when AssetCenter is installed and are not subject to regular variations (itemized lists, dictionary of features, currencies, business calendars, etc).

Lists

Setting the parameters for a particular list

Main list and tab lists can be configured via the **Configure list** menu in several cases:

- Lists displayed via the menus allowing you to access tables (**Portfolio/ Assets and batches** menu, for example).
- Lists displayed by views (**Tools/ Views** menu).
- Selection lists (**Select link** shortcut menu).
- Lists that appear in certain tabs of details.

Sorting lists

These lists can be sorted in several ways:

- By selecting your own sort conditions (**Sort** column).
- By using predefined indexes (**Sort by index** field).

The resulting performance of these two options sometimes differ. It is not possible to predict which option offers the best solution.

You need to try both solutions for each list in AssetCenter before deciding on which method is the best for your database.

Filters

Lists can also be filtered.

The time it takes to display a list increases with:

- The number of filtering criteria.
- The distance of the tables to which the filter criteria applies (distance from the table whose list displays the contents).
- The number of **OR** clauses in the filter query.

Selecting the columns to display

The time it takes to display a list increases with:

- The number of columns to display.
- The distance of the tables containing the fields and links are to be displayed.

Displaying in table or tree view.

The **tree** view takes longer to display than the **table** view.

Displaying icons in lists

Icons take longer to appear than text.

Setting list parameters at the database level

Certain interface options can affect the time it takes to display a list.

To access these options:

- 1 Select the **Edit/ Options** menu.
- 2 If necessary, modify the following options:
 - **Lists/ Other lists/ Do not load for longer than**
 - **Lists/ Other lists/ Do not load more than**
 - **Lists/ Main lists/ Do not load for longer than**
 - **Lists/ Main lists/ Do not load more than**

By reducing the number of lines loaded in a list, you can decrease the time it takes to display a list.

By reducing the maximum amount of time allotted for displaying a list, you thus limit the number of lines displayed according to what you deem a reasonable amount of loading time.

Note:

These options are stored in the AssetCenter database and are the same for all users who access it.

To learn more about these interface options, refer to the **Introduction** guide, chapter **Customizing a client workstation**, paragraph, **General AssetCenter interface options**.

Setting list parameters at the AssetCenter client level

The **ArrayFetchingSize** parameter is used by all the DBMSs supported by AssetCenter for an AssetCenter data connection.

To learn about this parameter's utility, you need to know that a DBMS sends the requested records in packets from an AssetCenter client. The size of these packets (expressed by number of records) is defined by the **ArrayFetchingSize** parameter.

This parameter is defined at the level of each AssetCenter client in the **amdb.ini** file and for each AssetCenter connection. This file is located at the root of the Windows installation folder.

If the **ArrayFetchingSize** parameter is not in the **amdb.ini** file, then its default value is **30**.

This parameter is integrated with the following parameters of the **Edit/Options** menu:

- **Lists/ Other lists/ Do not load more than**
- **Lists/ Main lists/ Do not load more than**

Non-optimized example for WAN

- Let's suppose that the **Do not load more than** option is set to **20**.
- Let's also suppose that **ArrayFetchingSize** is set to **30**.
- Then AssetCenter will fetch the list in 7 rounds (since $200 / 30 = 6,7$). This requires more time than if the list was fetched just once.

 **Tip:**

This amount of time is generally insignificant for a LAN but can be quite noticeable for a WAN.

Optimized example for a WAN

If it takes a long time to display a list, you need to configure the application to fetch records just once.

Here's the rule you need to apply:

```
ArrayFetchingSize = Do not load more than + 1
```

 **Note:**

We tested this scenario with a WAN using a 250 ms ping. By optimizing this parameter, we gained 1.5 s upon displaying a list of 200 records.

Example of application:

- **Do not load more than** equals 200.
- Set the value of **ArrayFetchingSize** to 210.
- Thus, AssetCenter fetches just once.

To modify amdb.ini

- 1 Edit the **amdb.ini** file.
- 2 Search for the selection [`<AssetCenter connection name to optimize>`]
- 3 Find out if a line already exists in the section that starts with `ArrayFetchingSize=`.
If this is the case, modify the value of the existing parameter.
- 4 If this is not the case, add a complete line in the section:
`ArrayFetchingSize=<Parameter value>`

 **Tip:**

This must be done on each client workstation.

Stripping down screens

To increase the reaction times of the application, you can restrict the number of data items appearing on screen, by only displaying those columns, lists and tabs that are strictly necessary.

Connection cache

You can also activate the connection cache, via the **Cache** tab in the connection detail screen:

Activating the connection cache:

- Reduces connection time to the database.
- Also saves time if you use images and icons.

In general, the default cache sizes are well optimized.

Access restrictions

It takes longer to display detail and list windows when there are access restrictions for the given login being used. This is due to the fact that AssetCenter performs a test before displaying the data.

If you have a doubt, display the list or detail with an unrestricted login and compare the display speed.

Delete any optional access restrictions as you see fit.

Applying the configuration of one client to other clients

Once you have optimized the performance of a given workstation, you now need to propagate the configuration modifications to the other client workstations.

In order to save time, you can copy the **amdb.ini** file corresponding to the modifications you have made.

16 | AutoCAD integration

CHAPTER

Presentation of AutoCAD/AssetCenter integration

You have the possibility of dynamically linking and tracking your AssetCenter data with the data in AutoCAD and vice versa.

The migration and tracking of your data is done using a Connect-It scenario.

 **Note:**

The provided Connect-It scenario is an example of data organization. This must be configured so that it can correctly replicate your data.

The AutoCAD functionality of AssetCenter adds two new menus in AutoCAD that enable you to import and view locations, employees and assets attached from AssetCenter.

Installation

Creating an AssetCenter database compatible with AutoCAD

 **Note:**

An AssetCenter demonstration database that supports AutoCAD is created during the AssetCenter installation. This database is located in the **Peregrine/AssetCenter/acadi/db** folder.

AssetCenter Database Administrator enables you to create an AssetCenter database that supports AutoCAD integration.

To create a database that supports AutoCAD:

- 1 Open a description file (.dbb).
- 2 Choose **Create a database** from the **Action** menu.
- 3 Perform the steps described in chapter Creating an AssetCenter database, section Creating an empty AssetCenter database using your DBMS.
- 4 Select the **Use AutoCAD** integration option.
- 5 Click **Create**.

FacilityCenter users

If you already have the AutoCAD integration module provided with FacilityCenter, use the scenario provided with AssetCenter to synchronize the AutoCAD data.

Configuring Connect-It

The Connect-It scenario provided with AssetCenter is located in the **Peregrine/AssetCenter/acadi/scenario** folder.

You must configure the AutoCAD connectors and AssetCenter to be able to use the **acadi.scn** scenario.

Configuring the AssetCenter connector

To configure the AssetCenter connector:

- 1 Open the **acadi.scn** scenario.
- 2 In the **Scenario diagram** screen, right-click the Asset Management connector and select **Configure connector** from the shortcut menu that appears.
- 3 In the wizard that appears, populate the **Name** field and **Description** field, if necessary, then click **Next**.
- 4 Select your AssetCenter database (**Connection** field) and populate the **Login** and **Password** fields of the ODBC connection before clicking **Next**.
- 5 Click **Finish**.

Configuring the AutoCAD connector

To configure the AutoCAD connector:

- 1 In the **Scenario diagram** screen, right-click the AutoCAD connector and select **Configure connector** from the shortcut menu that appears.
- 2 In the wizard that appears, populate the **Name** field and **Description** field, if necessary, then click **Next**.
- 3 Select your AssetCenter database that is compatible with AutoCAD (**Connection** field) and populate the **Login** and **Password** fields before clicking **Next**.
- 4 Click **Finish**.

Using AutoCAD integration

Graphic interface

The AutoCAD module installed by AssetCenter adds two menus in AutoCAD: DrawTools and AssetCenter.

Overview

You need to have already correctly configured the data synchronization in Connect-It before starting to work on your AssetCenter applications or on AutoCAD.

Synchronizing data with Connect-It

The **acadi.scn** Connect-It scenario enables you to synchronize and migrate data between AssetCenter and AutoCAD.

This scenario enables you to replicate data between AssetCenter and AutoCAD, which you can do periodically using a certain programming in Connect-It. Example: Data replication between AssetCenter and AutoCAD is performed each Friday. Refer to the Connect-It user's guide to find out more about this scenario.

You must replicate the AssetCenter data in AutoCAD (**Produce** contextual menu in the AssetCenter connector) to be able to use it in AutoCAD.

Using AssetCenter data in AutoCAD

To access your AssetCenter database that is compatible with AutoCAD:

- 1 Use the **AssetCenter/Property Portfolio** menu.
- 2 Select your AssetCenter database.
- 3 Choose your location (building, floor, room, etc.) and click **Attach**.
- 4 AutoCAD tells you when it is connected to the database, and you can starting working on the selected map.

After having worked in AutoCAD and saved your modifications - but before having used AssetCenter - launch the Connect-It scenario in order to synchronize your data (**Produce** contextual menu in the AutoCAD connector).

Data organization for the Connect-It scenario

For the provided Connect-It scenario to work properly:

- The locations must be organized as follows:
 - Campus
 - Building
 - Floor
 - Room/ Sublocation
- The **LocationType** field must be populated with: "Campus", "Building", "Floor", "Room/Sublocation".
- The assets and employees are those populated at the **Room/Sublocation** level.
- The assets are:
 - Nature.seBasis = portfolio item
 - Nature.seMgtConstraint = unique asset tag
 - Nature.bSoftLicense = 0

 **Note:**

The provided scenario can be entirely customized.

If you want to customized the provided scenario, we recommend that you contact Peregrine Systems support.

Using AutoCAD data in AssetCenter

After having migrated your AutoCAD data to Connect-It, the **Employees**, **Locations** and **Assets** tables will be populated or updated with the data originating from AutoCAD.

This data is populated in:

- The **General** tab, as well as in the **Portfolio** tab of a location detail. The AutoCAD drawing of a floor is visible in the **Document** tab of the floor of a location.
 - The **General** tab, as well as in the **Documents** tab of an employee detail.
 - The **General** and **Portfolio** tabs of a location detail.
-

 **Note:**

The AutoCAD module enables you to view data relating to assets and employees. To view other types of data, you must create a specific Connect-It scenario or contact Peregrine Systems customer support.

17 | Using a test database

CHAPTER

When you customize an AssetCenter database, we recommend that you work on a copy of that database to preserve the integrity of the original. This section details the different steps necessary for transferring the test database to the working database (of the same version of AssetCenter).

Here are the steps you must follow:

- 1 Make a copy of your working database. This will be your test database.
- 2 Develop and test the test database.
This test database contains the new data that will be categorized and transferred to the working database.
- 3 Export the additional data and the structure of the test database.
- 4 Import the additional data and the structure of the test database into the working database.
- 5 Migrate the data (the records) from the test database to the working database with Connect-It.

Copying your working database

 **Warning:**

The way you copy your working database depends on your database engine. We recommend that you refer to the documentation provided with your database engine.

AssetCenter Database Administrator enables you to copy your database in two different ways:

- By copying the database to an empty database.
- By performing a dump.

To copy a database to an empty database:

- 1 Connect to your working database (**File/Open/ Open existing database** menu).
- 2 Choose the empty shell in which you want to transfer the contents of your working database (**Action/Copy database to empty database** menu).

For more information about how to perform a dump, refer to this guide's chapter Creating an AssetCenter database, section Changing your DBMS.

Categorizing data

You need to categorize the different types of data in AssetCenter.

These types of data are:

- Records: wizards, workflows, actions, chargeback rules, etc.
- Additional data: default values, editable scripts in AssetCenter Database Administrator.
- Database structure: fields, links.

 **Note:**

When we say "data", we do not mean raw data like inventories, employees, locations, etc.

Exporting the database structure

AssetCenter Database Administrator enables you to export the structure of your test database.

To export the structure of your test database:

- 1 Create a folder to hold the database structure.
- 2 Connect to your working database (**File/Open/ Open existing database** menu).
- 3 Export the structure of this database (**File/Export the database structure** menu).

The structure and the additional data are exported in the following types of files:

- **.bin**
- **.cfg**
- **.dbb**
- **.dsd**
- **.str**
- **.stt**
- **.usr**
- **.wiz**

Importing the new database structure

 **Warning:**

We advise you to make a copy of your test database and your working database and save them. Importing the new structure is irreversible, and these copies will come in handy in case of any problem.

Importing the structure of your test database into your working database enables you to:

- Update your working database with the structure of the test database.
- Update the working database with the additional data from the test database.

To import the structure of your test database:

- 1 Connect to your working database (**File/Open/Open existing database** menu).
- 2 Choose the menu **Action/Update the database structure**.
- 3 Select the **.dbb** file in which you want to export the structure of the test database structure for the update.

Migrating data with Connect-It

Launch the Connect-It scenario corresponding to the data type to migrate. Then include a WHERE clause to distinguish the modified data from the original data.

