

Peregrine

AssetCenter



Administration

© Copyright 2005 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.

If you need technical support for this product, or would like to request documentation for a product for which you are licensed, contact Peregrine Systems, Inc. Customer Support by email at support@peregrine.com.

If you have comments or suggestions about this documentation, contact Peregrine Systems, Inc. Technical Publications by email at doc_comments@peregrine.com.

This edition applies to version 4.4 of the licensed program

AssetCenter

Peregrine Systems, Inc.
3611 Valley Centre Drive San Diego, CA 92130
858.481.5000
Fax 858.481.1751
www.peregrine.com



Table of Contents

- Introduction 11
- Who is the Administration module intended for? 11
- What does the Administration module do? 11

- Chapter 1. AssetCenter Database Administrator - GUI 13
- Introduction 13
- Graphical interface 14

- Chapter 2. Standard database description files 21
- Foreword 21
- Definition of a database 22
- Structure of the database.txt and tables.txt files 23
- Description of the tables 24
- Description of fields 25
- Description of links 32
- Description of indexes 39

- Chapter 3. Creating, modifying and deleting an AssetCenter database 41
- Obtaining a license file 41
- Creating an empty shell with the DBMS 43
- Creating a connection with AssetCenter 53
- Creating the database structure with AssetCenter Database Administrator
. 55

Modifying the license	58
AssetCenter client languages	59
Changing your DBMS	62
Deleting an AssetCenter database	64
Chapter 4. Creating a database description file	67
Introduction	69
Database description parameters	69
Syntax for description templates	71
Information about certain database description parameters	82
Chapter 5. Diagnostics and repairing a database	105
Chapter 6. Updating the database structure	109
Chapter 7. Accessing the database via ODBC	111
Overview of access to the AssetCenter database	111
Example: Creating a report in Crystal Reports with the ODBC driver	114
Chapter 8. Archival	115
Enabling archival	115
Archival principles	117
Disabling archival	122
Chapter 9. Denormalization	123
When is it advisable to denormalize a value?	123
Denormalization principles in AssetCenter	124
Denormalizing a value	125
Example	125
Chapter 10. Import	127
Overview of importing data	128
Recommendations	129
Constraints to be respected when importing data into a field	130
Importing text files or an ODBC database	133
Saving and executing an import script	159
Executing an import script from a command prompt	161
Chapter 11. Controlling access to the database	163
Importance and overview of database access rights	164
How to ensure data security and confidentiality	165
Definitions in access management	165

Defining access conditions	166
Defining AssetCenter users	172
Managing user connections	174
Managing passwords	178
Chapter 12. AssetCenter Server	187
Overview of AssetCenter Server	187
Executing AssetCenter Server	188
Main screen of AssetCenter Server	196
General options of AssetCenter Server	197
Configuring the modules monitored by AssetCenter Server	199
Activating the verification immediately	222
Administering AssetCenter Server via the Web	223
Chapter 13. Messaging	229
Overview of messaging	229
Configuring AssetCenter to use messaging systems	230
Common connection problems	240
Chapter 14. Customizing planner viewer screens	243
Overview	243
Customizing how the planner viewer pages work	243
Adding a planner viewer page to a table	244
Adding a planner viewer page to a wizard	249
References: parameter syntax of the planner viewer pages	251
Chapter 15. Using AssetCenter as a DDE server	259
Definition of a DDE server	259
DDE call mechanisms	259
Introduction to DDE commands	261
Global commands	261
Commands associated with a table	266
Commands associated with a table and a field or a link	273
Introduction to Examples of DDE calls	280
First scenario: AssetCenter internal DDE calls	280
Second scenario - DDE calls from Microsoft Excel	282
Third scenario: DDE calls from Visual Basic	284
Chapter 16. Optimizing AssetCenter for use in a WAN environment	287
Options in the Edit/ Options menu	287
Lists	288
Stripping down screens	292

Connection cache	292
Access restrictions	292
Applying the configuration of one client to other clients	293
Chapter 17. AutoCAD integration	295
Presentation of AutoCAD/AssetCenter integration	295
Installation	296
Configuration	296
Using AutoCAD integration	297
Index	301



List of Figures

1.1. Database opening screen	16
2.1. Intermediary tables - utilization	35
2.2. Link between the Assets table and the Fixed assets table	35
2.3. Links between the History table and a target table	39
10.1. Import - Selecting the type	127
11.1. User profile - detail window	164
11.2. User profiles - Access restrictions	171
13.1. Messaging system - overview	230
14.1. Planner viewer - Example with work orders	248



List of Tables

PEREGRINE

2.1. Fields - type	27
2.2. Data types - Entry format	28
2.3. Value of the User types field	30
2.4. Various types of existing links	33
2.5. Nature of the information stored by links	34
3.1. Important server parameters when creating the Oracle instance	45
3.2. Important database and server parameters	47
3.3. Important server parameters	49
3.4. AssetCenter Windows clients - language depends on the nature of the object	61
3.5. AssetCenter Windows clients - display languages according to the record type	62
4.1. Database instance properties	83
4.2. Table instance properties	83
4.3. Table instance objects	85
4.4. Field instance properties	85
4.5. Field instance objects	87
4.6. Possible values for the "Type" property	88
4.7. Possible values for the "UserType" property	88
4.8. Link instance properties	89
4.9. Link instance objects	90
4.10. Possible values for the "Type" property	91
4.11. Possible values for the "UserType" property	91
4.12. Index instance properties	92
4.13. Index instance objects	92

4.14. Script instance properties	93
4.15. Script instance objects	94
4.16. Properties of the FeatDesc instance	94
4.17. Objects of the FeatDesc instance	95
4.18. Properties of the FeatParamDesc instance	95
4.19. Objects of the FeatParamDesc instance	96
4.20. Properties of the FeatClassDesc instance	97
4.21. Objects of the FeatParamDesc instance	97
4.22. Properties of the CalcFieldDesc instance	98
4.23. Objects of the CalcFieldDesc instance	98
4.24. Properties of the View instance	98
4.25. Objects of the View instance	99
4.26. Properties of the Page instance	100
4.27. Properties of the PageItem instance	100
4.28. Objects of the PageItem instance	101
4.29. Properties of the SysEnumValue instance	101
4.30. Properties of the String instance	102
4.31. Properties of the ScriptField instance	102
4.32. Global variables	102
7.1. Advantages of using the ODBC driver	112
12.1. AssetCenter Server program menus and icons	196
12.2. Days of monitoring	200



Introduction

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 AssetCenter databases. It enables you to perform a large number of operations, for example:

- Create a database.
- Customize 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 modify the same database concurrently 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.



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.
-



All present languages have the same importance. The creation language does not have a special status.

The AssetCenter Database Administrator GUI shares the same look and feel as AssetCenter, such as for consultation and creation. For more information, refer to the **User interface** guide.

Selecting a language

Situated to the right of the toolbar is a field that enables you to select the display language for multilingual database objects (field, screen, page labels, system itemized list values, etc.).

► [Display language according to the nature of the objects](#) [page 61]

This field is enabled if you have connected to a database (as opposed to opening a database description file).

This field offers the following languages:

- Language of AssetCenter Database Administrator used to create the database,
 - Languages inserted after creating the database (**Action/ Insert a language in the database**).
- [AssetCenter client languages](#) [page 59]

Selecting a table

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

Selecting 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 (at the 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 (at the bottom) contains the information about the selected object.

 Note:

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

Manipulating files

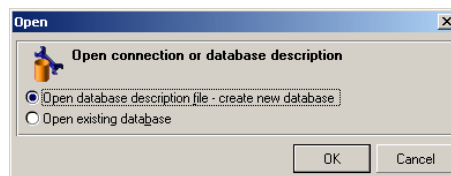
The **File** menu regroups the functions relating to loading and saving a file.

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 a data template during the creation of the database.

By validating this choice, you must provide AssetCenter Database Administrator the description file **gbbase.xml** located in the **Config** sub-folder of the AssetCenter installation folder.

 **Tip:**


In order to always have a copy of the standard description files at hand, we recommend making a backup copy of the **Config** folder.

 **Note:**

We also advise you to systematically generate a database description file for your databases (using the **File/ Export the database structure** 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.

This document does not deal with all the intricacies of editing the connection, accessible via the  button. For details, refer to the **User interface** guide.

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

There are two ways to save:

- The **File/Save database structure** menu enables you to save the modifications made to the database description file or to the database.
- The **File/ Export database structure** menu enables you to perform two distinct 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.

Exit the application

This menu enables you to quit AssetCenter Database Administrator. If you have made any modifications during the session, AssetCenter Database Administrator will ask you 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 to the database using external tools (via INSERT, DELETE, UPDATE statements or triggers). For the latter, we suggest you use the AssetCenter API.

Foreword

To import data, access the database using external tools, write queries, etc., you need to understand the structure of the database. This understanding will enable 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: These contain the complete structure of the database. They are located in the **doc\infos** sub-folder of the AssetCenter installation folder.

 Note:

These files describe the default database structure. They do not include any customizations you may have made.

In order for these description files to reflect customizations made to 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:
 - The AssetCenter database description files (**gbbase*.*** files) or a connection to the AssetCenter database.
 - A template (file with the **.tpl** extension), which describes which information is to be generated. We provide you with standard templates, and you can create your own. 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 the same database concurrently. The program uses a transactional method to update the data in the database and 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 `doc\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 `doc\infos` sub-folder of the AssetCenter installation folder.

The format of these files is as follows:

- 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.
- Tables are sorted by SQL name.
- For each table, the following information appears in this order:
 - 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

You use AssetCenter Database Administrator to define the description strings used for AssetCenter tables.

In an AssetCenter table, the description string describes those records that appear in drop-down lists and that enable you to select linked records.

It also defines the name that appears in the title bar of the window displaying 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 multiple levels.

Example:

The image shows a rectangular text box with a thin border. Inside the box, the text reads "String: [Name] [ModelName] [(AssetTag)]". The brackets are square and the text is in a monospaced font. There is a small magnifying glass icon at the end of the text.

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:

The image shows a horizontal rectangular box representing a window title bar. It contains the text "Detail of asset 'Asus AsusLX512 (1) '". The text is in a standard sans-serif font.

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 (Yes/No)
"d"	Date
"dt"	Date and time
"l"	Integer (32 bit)
"m"	Monetary
"p"	Percentage
"se"	System itemized list
"ts"	Duration
"mem"	Long text field
None	Text

- 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 **doc\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	Description
Integer (8 bit)	Integer (8 bit)	Integer from -128 to +127.
Integer (16 bit)	Integer (16 bit)	Integer from -32,768 to +32,767.
Integer (32 bit)	Integer (32 bit)	Integer from -2,147,483,647 to +2,147,483,646.
Floating point number	Floating point number	4-byte floating point number.
Double-precision number	Double-precision number	8-byte floating point number
Text	Text	Limited-length text field. All characters are accepted.
Binary field	Variable-length binary field.	Variable-length binary field (or BLOB Binary large object) can store images and forms, for example, without size restriction.
Long text field	Long text field	Variable-length text field (comments, for example). Certain DBMSs have limitations concerning these fields. Oracle for WorkGroups, for example, does not allow you to sort on this type of field.
Date and 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	Description
Default	By default	The information is displayed and entered just like it is stored in the database, i.e. depending on the field's Type .
Number	Number	Number.
Boolean (Yes/No)	Boolean (Yes/No)	Boolean.
Monetary	Monetary	Monetary values.
Date	Date	This field can only contain Date type values.
Date and time	Date and time	A field with this data-entry format contains Date+Time type values.
Time	Time	This field may only contain Time type values
Duration	Duration	Duration type values. The display format and default unit are defined by the User-TypeFormat 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	Custom 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.
Feature value	Feature value	Reserved. Do not use.
Basic script	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 types** field

Value of User type field in AssetCenter Database Administrator	Additional information
Duration	<p>This field enables you to define the display format and the default unit of a duration:</p> <pre data-bbox="534 409 1208 435">% [.D1] U1 [1] [d] [% [.D2] U2 [1] [d]] . . . [% [.Dn] Un [1] [d]]</pre> <p>Syntax:</p> <ul style="list-style-type: none"> ■ Ux is one of the following units: <ul style="list-style-type: none"> ■ Y: year ■ M: month ■ D: day ■ H: hour ■ N: minute ■ S: second ■ d is used to specify the unit to be applied if no unit is entered. Assign this parameter to one unit only. Example: If the timespan format is %M%Dd%H and you enter the value 10, this value will be interpreted as 10 days. ■ I is used to specify that the duration should be displayed its long format. If I is omitted, the unit will be displayed in its shortened form. Example: %YI displays year or years; %Y displays y. ■ .Dx is used to specify the number of decimals to display for the unit. Assign this parameter to the smallest unit of the format only. Example: If the duration format is %M%D%H%.2N and the stored duration is 15 seconds, the displayed value will be 0.25 min. <p>Example if the duration format is %MI%D%Hd%.1N:</p> <ul style="list-style-type: none"> ■ If you enter 50, AssetCenter takes this to mean 50 hours and stores 180 000 seconds. ■ If this stored value is 5 461 220 seconds, AssetCenter displays 2 months 3d 5h 2,3min.
System itemized list	Values taken by the itemized list.

Value of User type field in AssetCenter Database Administrator

Custom 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 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.

Exception

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 "lLocald".
- 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 "lLocald".
- 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 any 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	It represents a portion of contextual links.
Feature values	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 multiple records in table B. For example, the Contracts table is linked to the Assets table via a "1-->N" link: A contract can cover multiple 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

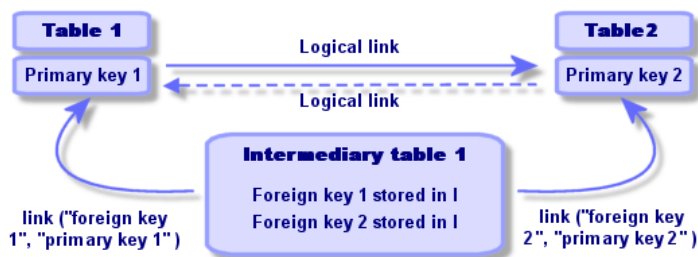
Intermediary tables are only used in the case of n-n cardinality logical links.

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.

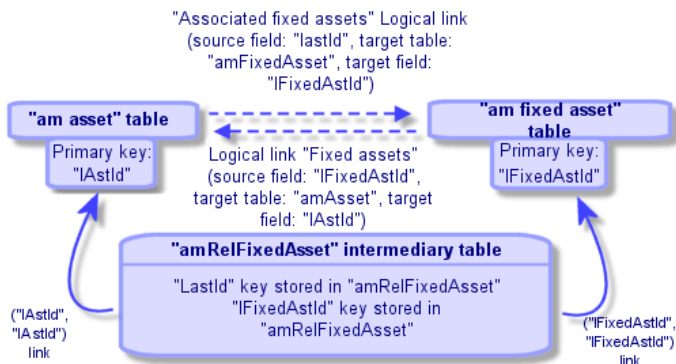
The following diagram explains how intermediary tables are used:

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 records (logical link **Associated fixed assets**):
 - Each record in the Assets table can be linked to multiple 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 multiple 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.

Overflow tables

These tables store:

- Additional information that is not contained in the original table
- The foreign key of overflow links (**Overflow** type link)

Overview of overflow tables

A table becomes an overflow table when it references an overflow link.

There are four overflow tables in the database structure:

- 1 The Software installations table (amSoftInstall)
- 2 The Telephones table (amPhone)
- 3 The Computers table (amComputer)

These tables are an extension of the Portfolio Items table (amPortfolio) and have its foreign key.

- 4 The Assets table (amAsset)

This table is an extension of the Portfolio Items table (amPortfolio): the foreign key is on the amPortfolio table and the PortfolioItems link shares the foreign key.

The links of an overflow table behave differently depending on whether they target a reference table or not.

The reference tables are:

- The Portfolio Items table (amPortfolio)
- The Contracts table (amContract)
- The Work Orders table (amWorkOrder)
- The Cables table (amCable)

- The Training table (amTraining)

For further information on the overflow table, refer to the **Portfolio** guide.

Behavior of an overflow link pointing to a reference table

For each overflow table pointing to a reference table:

- ◆ The creation of a record in the overflow table triggers the creation of a record in the reference table, except if the overflow link is irrelevant.

For example, on the Computers (amComputer) overflow table, the overflow link is irrelevant for computer groups. Therefore no record is created in the Portfolio Items table.

The creation of a record in the reference table triggers the creation of a record in the overflow table if:

- ◆ the **Also create** field of the nature of the model for this record references the overflow table.

Behavior of an overflow link pointing to other tables

For each overflow table pointing to a reference table:

- The creation of a record in the overflow table does not trigger the creation of a record in the reference table.
- The creation of a record in the reference table does not trigger the creation of a record in the overflow table.

Foreign key for an overflow table

For each overflow table pointing to a reference table:

- ◆ The foreign key is located in the overflow table

If you write a script in the reference table that queries an overflow table, remember to first execute a test query to check for the corresponding overflow link.

Implementation advice

When you create an overflow table, you must:

- Create the natures and models corresponding to the overflow link.
- Make sure the mandatory fields that are not populated by default are displayed in the page that depends on the overflow link.

Example of creating an overflow table

We are going to create a Cars table (amCar) as an overflow table of the Portfolio items (amPortfolio) table.

The steps are the following:

- Create the new table and the overflow link in AssetCenter Database Administrator,
 - Verify its behavior in AssetCenter.
- 1 To create the Cars table:
 - 1 Add a table to the database (**Database/ Add a table**)
 - 2 Populate the fields as follows:
 - **SQL name:** am Car
 - **Label:** Cars
 - 3 Add the fields of your choice
 - 2 To create the overflow link:
 - In the **amCar** table, create the link **Portfolio** (**Database/ Add a link**)
 - Populate the fields as follows:
 - **Link type:** Overflow
 - **Source** frame, **SQL name:** Portfolio
 - **Destination** frame, **Table:** Portfolio items (amPortfolio), **SQL name:** Car
 - Select **Create an index for this link**
 - 3 Create the page for the overflow table:

Populate the fields as follows:

 - **General** tab:
 - 1 **SQL name:** scrGenCar
 - 2 **Label:** General
 - **Content** tab:
 - 1 In the **List of fields** frame, add the link **Portfolio.Model** and the fields of your choice.
 - 4 To create the screen for the overflow link:

Populate the fields as follows:

 - **General** tab:
 - 1 **SQL name:** scrCar
 - 2 **Label:** Cars
 - 3 **Description:** Cars

- **Pages tab:**
 - 1 In the **Pages in detail** frame, add the following pages: scrGen car and PgGenDoc
- 5 Save your database.
- 6 Open AssetCenter and display the Natures screen
- 7 Verify that the **Also create** field includes a new itemized-list value: **Car**
- 8 Verify that for each item created in the Cars table (amCar) a corresponding item is created in the Portfolio Items table (amPortfolio)
- 9 Verify that for each portfolio item created a corresponding item is created in the Cars table (amCar)

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** (SQL name: 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.



3 | Creating, modifying and deleting an AssetCenter database

CHAPTER

This chapter explains how to:

- Create an AssetCenter database:
 - 1 Obtaining a license file [page 41]
 - 2 Creating an empty shell with the DBMS [page 43]
 - 3 Creating a connection with AssetCenter [page 53]
 - 4 Creating the database structure with AssetCenter Database Administrator [page 55]
- Modify certain parameters of an AssetCenter database:
 - Modifying the license [page 58]
 - AssetCenter client languages [page 59]
 - Changing your DBMS [page 62]
- Delete an AssetCenter database:
 - Deleting an AssetCenter database [page 64]

Obtaining a license file

The list of available functions depends on the license rights you have acquired.
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.

Your rights are stored in a license file (**license.cfg**) that Peregrine Systems provides you.

This file is inserted once only at the database level; it does not have to be declared at the client level.

The insertion is performed at the step [Creating the database structure with AssetCenter Database Administrator](#) [page 55].

The license can be obtained from Peregrine Systems. Please contact technical support for information on who to contact. You will be required to provide information based on the following procedure:

- 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 of 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 addresses of each network card of this computer.

 **Note:**

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

 **Warning:**

If AssetCenter Server is running 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 that has a short response time with the DBMS.

Creating an empty shell with the 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
- Your business rules
- The database administration tools you usually use
- If the database will be used by several users concurrently, you should tune the DBMS (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:

- 370 tables
- 1 600 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 in 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	For Latin code-page category languages (English, French, German, Italian, Spanish, etc.), we recommend the WE8ISO8859P15 character set. With certain Oracle versions, this parameter cannot be modified once the database is created.
db_block_size	This parameter is expressed in bytes. We recommend using the value 8192 . With certain Oracle versions, this parameter cannot be modified once the database is created.
db_block_buffers	Size of the database memory cache. This size is expressed in units of db_block_size . Examples to obtain a 200 MB cache: <ul style="list-style-type: none"> ▪ <code>db_block_buffers = 25000</code> if <code>db_block_size = 8192</code> ▪ <code>db_block_buffers = 100000</code> if <code>db_block_size = 2048</code> For a medium-sized server, the memory reserved for the database cache should represent 20% to 25% of the server's total memory. This recommendation must be adjusted if several users access the system concurrently (the database cache can be slightly reduced in a way that the server's memory does not swap).

Parameter	Recommended value
<code>shared_pool_size</code>	9,000,000 to 1,500,000 bytes on an Oracle 8.0.x according to the number of users (10 to 50). If the user is installed in the multi-threaded MTS, then this parameter is much higher (together with the <code>large_pool_size</code> on Oracle 8i).
<code>log_buffer</code>	This parameter is expressed in bytes. We recommend using the value 163840 .
<code>processes</code>	<Number of concurrent users> + <Number of Oracle system processes>. 50 is acceptable for 10 users maximum. To define according to the rights granted by the Oracle license
<code>dml_locks</code>	500
<code>open_cursors</code>	<Number of concurrent users> * Max [30; <Maximum number of records updated in multiple selection>].
<code>optimizer_mode</code>	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 AssetCenter Server.)
<code>sort_area_size</code>	On non-MTS servers, <code>sort_area_size</code> corresponds to the number of bytes, dedicated for sorts in memory.
<code>sort_area_retained_size</code>	<code>Sort_area_retained_size</code> corresponds to the memory conserved at the end of the memory sort. As a basic guideline, you can set it to 65256 , then increase it if necessary according to use. On an MTS server, the sort area is global for all multiplexed user connections, and sorts are done in the memory space. Start with <code>65536*number of concurrent users</code> , then increase it if necessary according to use.

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 (between 128 and 256 MB for Windows 2000), and 128 to 256 MB with Windows 2000. The system must not swap.
auto create statistics	As set in the database template at database creation.	Database setting, to be enabled for all AssetCenter databases.
auto update statistics	As set in the database template at database creation.	Database setting, to be enabled 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 **User** login.

- 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 saturating 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 (allotted memory, etc.).
- `Select @@version`
Recover the server version.
- `Sp_helpsort`
Recover the sort order physically used by the indexes.

Preparing the 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 MB +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 performance associated with queries on large tables (such as the Histories or Comments tables) is deteriorating, you need to implement identified caches for these tables. Then, when you query 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 maintaining performance.

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 database 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
- DB2 UDB Command Center

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:

- ```
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 <Name of the node> REMOTE <Server name> SERVER 50000  
REMOTE_INSTANCE DB2 SYSTEM <Server name> OSTYPE <NT|SUN|AIX|Etc.>`
- `CATALOG DATABASE <Database name>; AT NODE <Name of the node>`
- If you wish to use the AssetCenter database with the archival function, DB2 will need to be able to handle 8K pages.  
By default, DB2 only supports 4K pages.  
To support the archival function, you must create a **SYSTEM** type tablespace and an 8K **REGULAR** type tablespace.  
Example:

```
CREATE Bufferpool ACBP DEFERRED SIZE 250 PAGESIZE 8 K ;

CREATE REGULAR TABLESPACE ACTS PAGESIZE 8 K MANAGED BY SYSTEM USING (
'F:\DB2\ACTS') EXTENTSIZE 16 OVERHEAD 10.5 PREFETCHSIZE 16 TRANSFERRATE
0.33 BUFFERPOOL ACBP;

CREATE SYSTEM TEMPORARY TABLESPACE ACTMPTS PAGESIZE 8 K MANAGED BY SY
STEM USING ('F:\DB2\ACTMPTS') EXTENTSIZE 16 OVERHEAD 10.5 PREFETCHSIZ
E 16 TRANSFERRATE 0.33 BUFFERPOOL ACBP;
```

## 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 access the database concurrently.

 **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 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
```

## MSDE



Note:

MSDE is supported for the demonstration databases only.

To create an empty shell, proceed as follows:

- 1 Make sure that the Windows service of the MSDE instance is started (**MSSQL\$ASSETCENTER** service for the MSDE instance installed with AssetCenter).
- 2 Open a DOS command prompt
- 3 Execute the following commands:
  - a Enter the command to start the SQL editor for the MSDE instance you are using.

Syntax:

```
osql -S <MSDE server name>\<MSDE instance name> -U <User ID> -P <User
password>
```

Example for the MSDE instance installed with AssetCenter:

```
osql -S (local)\ASSETCENTER -U sa -P saacpassword
```



- b Execute the command:

```
go
```

- c Enter the command to create a new database.

Syntax:

```
create database <Database name>
```

Example:

```
create database ACTest01
```

- d Execute the command:

```
go
```

- 4 Exit the SQL editor (**Exit**).
- 5 Close the DOS box (**Exit**).

## 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 multiple DBMS logins with restricted database rights. This makes it possible to control access to the database using external tools. The use of multiple logins is not documented here.

### Oracle example

Declare an Oracle login with at least **CONNECT**, **RESOURCE** and **CREATE SESSION** rights. If you have already created an AssetCenter database in 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;
```

## Creating a connection with AssetCenter

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

- 1 Start AssetCenter.
- 2 AssetCenter displays the **Connect to database** window.  
Exit this window without connecting to the database (**Cancel** button).
- 3 Display the list of connections **File/ Manage connections**.
- 4 Display a connection creation page (**New** button).
- 5 Populate the connection detail.
- 6 Create the connection (**Create** button).
  - ▶ [Useful information for creating a connection to an MSDE database \[page 54\]](#)
- 7 Do not test the connection at this stage (**Test** button) because the database structure has not yet been created.
- 8 Do not open the connection at this stage (**Open** button) because the database structure has not yet been created.
- 9 Close all the windows.
- 10 Exit AssetCenter.

---

 **Note:**

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

---

## Useful information for creating a connection to an MSDE database

Populate the following fields:

| Field      | Value                                                                                          |
|------------|------------------------------------------------------------------------------------------------|
| Name       | <Name of your choice>                                                                          |
| Engine     | Microsoft SQL Server                                                                           |
| Datasource | ACDemo44en (SQL Server)                                                                        |
| Database   | <Name of the empty shell created with the MSDE SQL editor><br>▶ <a href="#">MSDE [page 52]</a> |
| User       | sa                                                                                             |
| Password   | saacpassword                                                                                   |

| Field | Value                    |
|-------|--------------------------|
| Owner | <Leave this field empty> |

## Creating the database structure with AssetCenter Database Administrator

We are going to use AssetCenter Database Administrator to create the AssetCenter database structure inside the empty shell created with the DBMS. This structure is based on a description file installed with AssetCenter.

To create the database structure:

- 1 Start AssetCenter Database Administrator.
- 2 Open the database description file:
  - a Display the **Open** window (**File/ Open**).
  - b Select the **Open database description file - create new database** option.
  - c Validate the option (**OK** button).
  - d Select the file **gbase.xml** (located in the **config** sub-folder of the AssetCenter installation folder, in general **C:\Program Files\Peregrine\AssetCenter\config**).
  - e Open the file (**Open** button).
- 3 Start the wizard to create the database structure (**Action/ Create database** menu).
- 4 Populate the pages of the wizard (navigate through the wizard pages using the **Next** and **Previous** buttons).

**Generate SQL script / Create database page:**

| Fields   | Value                                                                                                                                    |
|----------|------------------------------------------------------------------------------------------------------------------------------------------|
| Database | Select the connection to the empty shell created with AssetCenter.<br><a href="#">▶ Creating a connection with AssetCenter [page 53]</a> |

| Fields                        | Value                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Creation                      | <ul style="list-style-type: none"> <li>■ <b>Create the database:</b> Select this option if you wish to create a database with the standard SQL script.</li> <li>■ <b>Generate the SQL script:</b> Select this option if you wish to generate the standard SQL database creation script in order to modify it.<br/>In this case: <ol style="list-style-type: none"> <li>1 Finish executing the database creation wizard, which will simply create a SQL script file.</li> <li>2 Modify the SQL script.</li> <li>3 Execute the database creation wizard once more, this time selecting the option <b>Create the database using a script</b>.</li> </ol> </li> <li>■ <b>Create the database using a script:</b> Select this option if you wish to create a database using a SQL script created using the <b>Generate the SQL script</b> option that you have subsequently modified yourself.</li> </ul> |
| Use advanced creation options | Select this option if you need to define the tablespaces for tables, indexes and LOBs.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |

#### Creation options page:



Note:

The tablespaces must be created beforehand using the DBMS.

#### Select script page:

| Fields | Value                                                 |
|--------|-------------------------------------------------------|
| File   | Enter the full path of the SQL script file to create. |

| Fields    | Value                                                                                                                                                                                                                                                                                                                                                                                                         |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Separator | <p>Select the separator for SQL statements:</p> <ul style="list-style-type: none"> <li>■ /: For an Oracle database</li> <li>■ <b>GO</b>: For all other databases</li> <li>■ <b>Any other separator</b>: as long as it is a valid separator (such as ;).</li> </ul> <p>You are free to define your own separator such as &lt;<b>MySeparator</b>&gt;, but the database creation script will no longer work.</p> |

### Creation parameters page:

| Fields           | Value                                                                                                                                                                                                                                                                                                                                                                                                   |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Creation script  | Enter the full path of the SQL script file to use.                                                                                                                                                                                                                                                                                                                                                      |
| Password         | <p>Password to associate with the administrator login.</p> <p><b>Note:</b></p> <p>The AssetCenter database administrator is the record in the <b>Departments and employees</b> (amEmplDept) table for which the <b>Name</b> (Name) field is set to <b>Admin</b>.</p> <p>The database connection login is stored in the <b>User name</b> (UserLogin) field. The administration name is <b>Admin</b>.</p> |
| Confirm password | Confirm the value entered in the <b>Password</b> field.                                                                                                                                                                                                                                                                                                                                                 |
| License file     | <p>Select the license file <b>license.cfg</b> provided with AssetCenter.</p> <p>► <a href="#">Obtaining a license file</a> [page 41].</p>                                                                                                                                                                                                                                                               |

### Create system data page:

| Fields         | Value                                                                                                                                                                                                                     |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Use time zones | <p>Select this option in order to be able to manage the time-zone differences between the database server, the date field and the AssetCenter clients.</p> <p>► <b>Advanced use</b> guide, chapter <b>Time zones</b>.</p> |

| Fields                  | Value                                                                                                                                                                    |
|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Server                  | Select the time zone of the AssetCenter database server.<br><br>This is the time zone defined, for example, at the level of the Windows Control Panel of the server.     |
| Data                    | Select the time zone in which the dates stored in the database will be expressed.                                                                                        |
| Use help on fields      | Select this option to insert the information describing the database fields and links.                                                                                   |
| Use AutoCAD integration | Select this option if you plan on using the AutoCAD integration functionality in AssetCenter.<br><br>► <b>Administration</b> guide, chapter <b>AutoCAD integration</b> . |

### Data to import page:

| Fields               | Value                                                                                                                                                                                                                                                                                                                                                                                                                     |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Available data       | Select the sample data or line-of-business data to import into the database.<br><br>For a production database, we recommend that you: <ul style="list-style-type: none"> <li>■ Select the line-of-business data for the modules you wish to use.</li> <li>■ Do not select the sample data.</li> </ul> <p><b>Note:</b></p> <p>The data that you do not import when creating the database can be easily imported later.</p> |
| Stop import if error | Select this option for the import to stop if a problem is encountered.                                                                                                                                                                                                                                                                                                                                                    |
| Log file             | Full name of the file to which all import operations, including errors and warnings, are logged.                                                                                                                                                                                                                                                                                                                          |

- 5 Execute the options defined using the wizard (**Finish** button).

## Modifying the license

You must change the license file for your database in the following cases:

- If you change the network cards on the computer on which AssetCenter Server is running
- If your license expires
- If you acquire additional rights

AssetCenter Database Administrator enables you to perform this modification while connected or disconnected from the database.

## 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.
- 4 Select the new license file in the dialog box that appears.

---

## AssetCenter client languages

### Overview

Certain AssetCenter database objects are multilingual (► [Display language according to the nature of the objects](#) [page 61]).

In order for multilingual objects from the AssetCenter database to be available at the client level, you must:

- 1 Start by adding the corresponding language parameters to the database (► [Adding language parameters](#) [page 60]).
- 2 Next, install a corresponding language version of the AssetCenter client on the user workstation; Multilingual database objects are automatically displayed in the language of the client (► [Displaying a given language at the AssetCenter client level](#) [page 60]).

It is possible to use any or all of the languages supported by the AssetCenter software suite.

For example, you may access your database using an English or a Spanish client.

## Adding language parameters

Databases are created with AssetCenter Database Administrator using the parameters corresponding to the language version of AssetCenter Database Administrator.

To add other language parameters:

- 1 Determine which languages you want to support.
  - 2 Install AssetCenter in each one of these additional languages.
- 



**Tip:**

It is possible to just install AssetCenter Database Administrator.

---

- 3 Start AssetCenter Database Administrator in the language of your choice.
  - 4 Open your database (**File/Open/ Open existing database** menu).
  - 5 For each of the languages to be added:
    - 1 Select **Action/ Insert a language in the database**.
    - 2 Select the folder in which the appropriate language version of AssetCenter is installed.  
Example: **C:\Program Files\Peregrine\AssetCenter-English**
    - 3 Click **OK** to confirm.
- 



**Warning:**

If you add the parameters of a language that is already used, any customizations will be overwritten, including those that refer to the added objects.

---

## Displaying a given language at the AssetCenter client level

Once the language parameters have been added to the AssetCenter database by the administrator:

- 1 Install an AssetCenter Windows client in the required language.
- 2 Start the AssetCenter client.
- 3 Connect to the database.

The multilingual database objects are automatically displayed in the language of the client.



 Note:

At the level of an AssetCenter client, multilingual objects can only be displayed in one language: that of the client itself.

## Display language according to the nature of the objects

The display language of an object depends on the nature of the object:

Table 3.4. AssetCenter Windows clients - language depends on the nature of the object

| Nature of the object                                                                                                                                                                                                                                | Storage of text of the object | Display language by the AssetCenter client                                  |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------|-----------------------------------------------------------------------------|
| <ul style="list-style-type: none"> <li>■ Labels and descriptions of tables, fields, links and indexes</li> <li>■ Screen label</li> <li>■ Page label</li> <li>■ System itemized lists value</li> <li>■ Extended help on a field or a link</li> </ul> | Database                      | Language of the AssetCenter client connected to the database                |
| ◆ Data stored in the database                                                                                                                                                                                                                       | Database                      | Language in which the user enters the data                                  |
| ◆ Tip of the day                                                                                                                                                                                                                                    | Database                      | Language of AssetCenter Database Administrator, used to create the database |
| ◆ Program message                                                                                                                                                                                                                                   | Client program                | Language of the AssetCenter client connected to the database                |

Certain database records require additional information:

Table 3.5. AssetCenter Windows clients - display languages according to the record type

| Record type             | Comment                                                                                                                                                                                                                                                                                                 |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Crystal Reports         | <p>You can display a report regardless of the language used to create it.</p> <p>However, labels are displayed in the language that was used for the creation of the report.</p> <p>If you want the labels to be available in other languages, you must duplicate the report and translate it.</p>      |
| Forms                   | Same principle as the reports.                                                                                                                                                                                                                                                                          |
| Actions and wizards     | Same principle as the reports.                                                                                                                                                                                                                                                                          |
| Queries                 | They are language independent                                                                                                                                                                                                                                                                           |
| Demonstration databases | <p>They are monolingual (installation language of AssetCenter).</p> <p>You can add other language parameters in the same way you would for your production database.</p>                                                                                                                                |
| Workflow schemes        | <p>You can execute a workflow scheme, no matter what language was used to create it.</p> <p>However, labels are displayed in the language that was used for the creation of the scheme.</p> <p>You cannot duplicate a workflow scheme because the workflow instances would then also be duplicated.</p> |
| Online help             | The online help can be used in any one language. The default language is that of the client installed on the user's desktop.                                                                                                                                                                            |
| Documentation           | During the installation of AssetCenter on a client desktop, only <b>pdf</b> files matching the installation language are installed. If you need <b>pdf</b> files in other languages, simply copy them from a desktop on which they are installed.                                                       |

## Changing your DBMS

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

To change your DBMS, you must export the structure and contents of your database from the old DBMS, then import it into the new one.

---

 **Warning:**

The database versions must be the same. You cannot, for example, export a AssetCenter 3.x database to a AssetCenter 4.4 database.

---

Follow these steps to perform the dump:

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

 **Note:**

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

- [Creating an empty shell with the DBMS](#) [page 43],
  - [Creating a connection with AssetCenter](#) [page 53],
  - [Creating the database structure with AssetCenter Database Administrator](#) [page 55].
- 

## Exporting the database structure and its data

To export:

- 1 Start AssetCenter Database Administrator.
- 2 Open your AssetCenter database (**Open/ Open existing database** menu in AssetCenter Database Administrator).
- 3 Select the **Action/Export database structure and data** menu.
- 4 Indicate the name of your export and click **Save**.

AssetCenter generates the export files.

---

 **Note:**

The export 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 export.

---

## Importing the database structure and data

Once you have completed the export, and after you have created the empty shell and its connection declaration, you can now import the exported files:

- 1 Use the **Action/ Import database structure and data** menu.
- 2 In the dialog box that is displayed, select the empty shell you have created.
- 3 Select the first export file (**xxx01.ar**) generated by AssetCenter Database Administrator.

AssetCenter imports the export files into your new database.

## Deleting 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**.
- In the case of MSDE:

- 1 Make sure that the Windows service of the MSDE instance is started (**MSSQL\$ASSETCENTER** service for the MSDE instance installed with AssetCenter).

- 2 Open a DOS command prompt

- 3 Execute the following commands:

- a Enter the command to start the SQL editor for the MSDE instance you are using.

Syntax:

```
osql -S <MSDE server name>\<MSDE instance name> -U <User ID> -P <User password>
```

Example for the MSDE instance installed with AssetCenter:

```
osql -S (local)\ASSETCENTER -U sa -P saacpassword
```

- b Execute the command:

- c Enter the command to detach a database.

Syntax:

Example:

- d Execute the command:

- e Delete the **.mbf** and **.log** files related to the database.
- 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.





# 4 Creating a database description file

## CHAPTER

AssetCenter Database Administrator enables you to extract information from the database. You can control the nature of the extracted information and the extraction format.

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 **doc\infos** sub-folder of the AssetCenter installation folder.

---

 Note:

These files describe the default database structure. They do not include any customizations you may have made.

In order for these description files to reflect customizations made to 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:
    - An AssetCenter database description file (**gbbase\*. \* files**) or a connection to an AssetCenter database.

- A template (.**tpl** file extension) that describes which information to generate. We provide you with the standard templates, but you can create your own if you wish. Sophisticated templates enable you to create RTF or HTML formatted files.
- 

 **Note:**

Among the standard templates 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 itself 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 going through 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 in 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 <instance name>

Instance FIELD
Property P1-n
Collection DEFVALDEPENDENCIES as DEFVALSCRIPT
Collection SYSENUMVALUES as SYSENUMVALUE
Object O1-n as <instance name>

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

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

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

Instance FEATDESC
Collection PARAMS as FEATPARAMDESC
Object O1-n as <instance name>

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

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

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

Instance VIEW
Property P1-n
Collection PAGES as PAGE
Collection FIELDSINLISTCONFIG as PAGEITEM
Object O1-n as <instance name>

Instance PAGE
Property P1-n
Collection FIELDS as PAGEITEM
Object O1-n as <instance name>

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

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

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

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

```
Global Values
Property P1-n
```

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

- Database: the database itself.
- Table: database tables.
- Field: fields in the tables.
- Link: links in the tables.
- Index: indexes in the tables.
- Script: scripts for calculating the values of 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 object>.] <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 variables used in the declaration of a function are not verified. Thus, it is possible to use a variable without declaring it, which would cause errors when the script is executed. 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 = "Integer (8 bit)"
case 2: StrType = "Integer (16 bit)"
case 3: StrType = "Integer (32 bit)"
case 4: StrType = "Floating point number"
case 5: StrType = "Double-precision number"
case 6: StrType = "Text"
case 7: StrType = "Date and time"
case 9: StrType = "Binary field"
case 10: StrType = "Date"
```

```

case 11: StrType = "Time"
case 12: StrType = "Long text field"
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"> <li>■ 1: AssetCenter Database Administrator was executed in reference to a connection.</li> <li>■ 0: AssetCenter Database Administrator was executed in reference to a database description file.</li> </ul> | No                  |
| Connection    | Name of the AssetCenter connection used to access the database.                                                                                                                                                                                                                 | Yes                 |
| AppInfo       | Information about AssetCenter.                                                                                                                                                                                                                                                  | No                  |
| AppVersion    | AssetCenter version number.                                                                                                                                                                                                                                                     | No                  |
| AppBuild      | AssetCenter build 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"> <li>■ 1: The login has create access rights for the table.</li> <li>■ 0: The login does not have create access rights for the table.</li> </ul> | Yes                 |

| Property name        | Description                                                                                                                                                                                             | Connection required |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|
| Delete               | This property may have two values: <ul style="list-style-type: none"> <li>■ 1: The login has delete rights for the table.</li> <li>■ 0: The login does not have delete rights for the table.</li> </ul> | Yes                 |
| ComputeString        | Table description string.                                                                                                                                                                               | No                  |
| InternalName         | Internal name.<br>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, taking into account the filter and the sort order: <ul style="list-style-type: none"> <li>■ 0: no</li> <li>■ 1: yes</li> </ul>    | No                  |
| IsLast               | Indicates whether the item is located at the end of the collection, taking into account the filter and the sort order: <ul style="list-style-type: none"> <li>■ 0: no</li> <li>■ 1: yes</li> </ul>      | No                  |
| Count                | Counts the number of items browsed in the collection, taking the filter into account.                                                                                                                   | No                  |
| CurrentIndex         | Position of the item in the collection, taking into account the filter and the sort order.                                                                                                              | No                  |
| System               | Indicates whether this object is a system object (cannot be modified): <ul style="list-style-type: none"> <li>■ 0: no</li> <li>■ 1: yes</li> </ul>                                                      | No                  |
| HasFeatureValueTable | Indicates whether this table has a table of feature values associated with it: <ul style="list-style-type: none"> <li>■ 0: no</li> <li>■ 1: yes</li> </ul>                                              | No                  |
| IsFeatureValueTable  | Indicates whether this table is a table of feature values: <ul style="list-style-type: none"> <li>■ 0: no</li> <li>■ 1: yes</li> </ul>                                                                  | No                  |



| Property name | Description                                                                                                                                  | Connection required |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------|---------------------|
| HasMemoField  | Indicates whether this table has a <b>Long text</b> -type field. <ul style="list-style-type: none"> <li>■ 0: no</li> <li>■ 1: yes</li> </ul> | 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"> <li>■ 1: The login has update rights for the field.</li> <li>■ 0: The login does not have update rights for the field.</li> </ul>               | Yes                 |
| Write         | This property may have two values: <ul style="list-style-type: none"> <li>■ 1: The login has create access rights for the field.</li> <li>■ 0: The login does not have create access rights for the field.</li> </ul> | Yes                 |

| Property name   | Description                                                                                                                                                                                                                                                                                                                                         | Connection required |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|
| Read            | This property may have two values: <ul style="list-style-type: none"> <li>1: The login has read rights for the field.</li> <li>0: The login does not have read rights for the field.</li> </ul>                                                                                                                                                     | Yes                 |
| UserType        | By default, the data entry and display format is the format for the "Type" property.<br>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, as a number of characters.                                                                                                                                                                                                                                                                                 | No                  |
| ReadOnly        | Defines whether the field may be modified, regardless of the access rights for the person connected to the database.<br>This field may have two values: <ul style="list-style-type: none"> <li>1: The field can never be modified by users.</li> <li>0: The field may be modified by a user, if the user has the required access rights.</li> </ul> | No                  |
| Historized      | This field may have two values: <ul style="list-style-type: none"> <li>1: History is recorded for the field.</li> <li>0: History is not recorded for the field.</li> </ul>                                                                                                                                                                          | No                  |
| ForeignKey      | This field may have two values: <ul style="list-style-type: none"> <li>1: The field is a foreign key.</li> <li>0: The field is not a foreign key.</li> </ul>                                                                                                                                                                                        | No                  |
| PrimaryKey      | This field may have two values: <ul style="list-style-type: none"> <li>1: The field is a primary key.</li> <li>0: The field is not a primary key.</li> </ul>                                                                                                                                                                                        | No                  |
| InternalName    | Internal name<br>This information has no particular use.                                                                                                                                                                                                                                                                                            | 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                  |

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

| Object name              | Description                                 |
|--------------------------|---------------------------------------------|
| 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            | Integer (8 bit)         | Integer from -128 to +127.                                                                                                     |
| 2            | Integer (16 bit)        | Integer from -32,768 to +32,767.                                                                                               |
| 3            | Integer (32 bit)        | Integer from -2,147,483,647 to +2,147,483,646.                                                                                 |
| 4            | Floating point number   | 4-byte floating point number.                                                                                                  |
| 5            | Double-precision number | 8-byte floating-point number.                                                                                                  |
| 6            | Text                    | Text in which all characters are allowed.                                                                                      |
| 7            | Date and time           | Date and time                                                                                                                  |
| 9            | Binary field            | 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).                                                                                                           |
| 11           | Time                    | Just time (no date)                                                                                                            |
| 12           | Long text field         | 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            | Boolean (Yes/No)        |
| 3            | Monetary                |
| 4            | Date                    |
| 5            | Date and time           |
| 6            | Time                    |
| 7            | System itemized list    |
| 8            | Custom itemized list    |
| 9            | Feature value           |
| 10           | Percentage              |
| 11           | Duration                |
| 12           | Table or field SQL name |

| Stored value | Literal value |
|--------------|---------------|
| 13           | Basic script  |

## 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.
- "Duration": 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"> <li>▪ 1: The login has update rights for the link.</li> <li>▪ 0: The login does not have update rights for the link.</li> </ul>               | Yes                 |
| Write         | This property may have two values: <ul style="list-style-type: none"> <li>▪ 1: The login has create access rights for the link.</li> <li>▪ 0: The login does not have create access rights for the link.</li> </ul> | Yes                 |
| Read          | This property may have two values: <ul style="list-style-type: none"> <li>▪ 1: The login has read rights for the link.</li> <li>▪ 0: The login does not have read rights for the link.</li> </ul>                   | 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"> <li>1: The target table is pre-defined.</li> <li>0: The target table is pre-defined.</li> </ul> | No                  |
| Historized               | This field may have two values: <ul style="list-style-type: none"> <li>1: History is recorded for the field.</li> <li>0: History is not recorded for the field.</li> </ul>                                                                                                                        | No                  |
| Cardinality              | Cardinality of the link.                                                                                                                                                                                                                                                                          | No                  |
| InternalName             | Internal name<br><br>This information has no particular use.                                                                                                                                                                                                                                      | No                  |
| Label                    | Label.                                                                                                                                                                                                                                                                                            | No                  |
| Desc                     | Description                                                                                                                                                                                                                                                                                       | No                  |
| SqlName                  | SQL name.                                                                                                                                                                                                                                                                                         | No                  |
| System                   | Indicates whether this object is a system object (cannot be modified): <ul style="list-style-type: none"> <li>0: no</li> <li>1: yes</li> </ul>                                                                                                                                                    | 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 |

| Object name              | Description                                                                                                                   |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------|
| 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  | Equivalent displayed by Asset-Center Server |
|--------------|----------------|---------------------------------------------|
| 0            | Normal         | Normal                                      |
| 1            | Comment        | Comment                                     |
| 2            | Image          | Image                                       |
| 3            | History        | History                                     |
| 4            | Feature values | 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"> <li>■ 1: You can create several records where the group of fields in the index can have exactly the same value.</li> <li>■ 0: You cannot create more than one record where the group of fields in the index has a given value.</li> </ul>                                                                                                   |
| NullValues    | <p>This property is only significant if the "Duplicates" property is set to "No".</p> <p>It indicates whether or not the index can have more than one "NULL" value. (The index has a value of "NULL" if all the fields constituting the index have a value of "NULL".)</p> <ul style="list-style-type: none"> <li>■ 1: You can create several records having an index of NULL.</li> <li>■ 0: You can only create one record having an index of NULL.</li> </ul> |
| InternalName  | <p>Internal name</p> <p>This information has no particular use.</p>                                                                                                                                                                                                                                                                                                                                                                                             |
| Label         | Label as it appears in the detail screens.                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Desc          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| SqlName       | SQL name                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| System        | <p>Indicates whether this object is a system object (cannot be modified):</p> <ul style="list-style-type: none"> <li>■ 0: no</li> <li>■ 1: yes</li> </ul>                                                                                                                                                                                                                                                                                                       |

## 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"> <li>■ 0: no</li> <li>■ 1: yes</li> <li>■ 2: script</li> </ul>                                                                                                                                                                                                                                  |
| ScriptType                  | Type of information managed by the script. This property can have one of the following values: <ul style="list-style-type: none"> <li>■ 1: Data entry for a field is mandatory.</li> <li>■ 2: By default, displays a feature in a table.</li> <li>■ 3: Available character for a feature in a table.</li> <li>■ 4: Records history for a field.</li> <li>■ 5: Default value of a field.</li> <li>■ 6: <b>For inheritance purposes</b> (bForInheritance) field of a feature in a table.</li> </ul> |
| Source                      | Script that calculates the value of a field as it is displayed in the interface.                                                                                                                                                                                                                                                                                                                                                                                                                  |
|                             | 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"> <li>■ Integer: Integer from -32,768 to +32,767</li> <li>■ Long: Integer from -2,147,483,647 to +2,147,483,646</li> <li>■ Double: 8-byte floating-point number</li> <li>■ String: Text for which all characters are accepted</li> <li>■ Date: Date (not time).</li> </ul>                                                                                                                                              |
| 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 whether this object is a system object (cannot be modified): <ul style="list-style-type: none"> <li>■ 0: no</li> <li>■ 1: yes</li> </ul>                         |
| Label         | Label of the feature.                                                                                                                                                      |
| Desc          | Description                                                                                                                                                                |
| SQLName       | SQL name                                                                                                                                                                   |
| Unit          | Unit of the feature.                                                                                                                                                       |
| Type          | Entry type of the feature.<br>Itemized list used to populate the value of the feature. This property is only useful if the feature's <b>Type</b> is <b>Itemized list</b> . |
| MinValue      | The minimum value accepted by the feature (numeric <b>type</b> only).                                                                                                      |
| MaxValue      | The maximum value accepted by the feature (numeric <b>type</b> only).                                                                                                      |

| Property name  | Description                                                                                                                               |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| IsConsolidated | Indicates whether the value is calculated by consolidation. <ul style="list-style-type: none"> <li>■ 0: no</li> <li>■ 1: yes</li> </ul>   |
| HasClass       | Indicates whether the feature is attached to a feature class: <ul style="list-style-type: none"> <li>■ 0: no</li> <li>■ 1: yes</li> </ul> |

## 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). |

| Property name            | Description                                                                                                                                                                                 |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 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 whether the feature is inherited: <ul style="list-style-type: none"> <li>■ 0: no</li> <li>■ 1: yes</li> </ul>                                                                     |
| CreationHistorized       | History line created when the main record is created.                                                                                                                                       |
| Write                    | This property can take the values: <ul style="list-style-type: none"> <li>■ 0: The login as write rights over the feature.</li> <li>■ 1: The login does not have write rights.</li> </ul>   |
| Read                     | This property can take the values: <ul style="list-style-type: none"> <li>■ 0: The login as read rights over the feature.</li> <li>■ 1: The login does not have read rights.</li> </ul>     |
| Update                   | This property can take the values: <ul style="list-style-type: none"> <li>■ 0: The login as update rights over the feature.</li> <li>■ 1: The login does not have update rights.</li> </ul> |

## 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 whether the feature class has a parent class: <ul style="list-style-type: none"> <li>■ 0: no</li> <li>■ 1: yes</li> </ul> |

---

### 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 of the calculated field.                                                                                                          |
| UserType      | Result type returned by the calculated field.                                                                                             |
| Type          | Type of calculated field.                                                                                                                 |
| IsVisible     | Indicates whether the calculated field is visible by default: <ul style="list-style-type: none"> <li>■ 0: no</li> <li>■ 1: yes</li> </ul> |

## 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<br>This information has no particular use. |

| Property name | Description                                                                                                                                                           |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| System        | Indicates whether this object is a system object (cannot be modified): <ul style="list-style-type: none"> <li>■ 0: no</li> <li>■ 1: yes</li> </ul>                    |
| Label         | Label of the screen.                                                                                                                                                  |
| Desc          | Description                                                                                                                                                           |
| SQLName       | SQL name                                                                                                                                                              |
| FuncDomain    | Functional domain associated with the screen.                                                                                                                         |
| HasSystemPage | Indicates whether this screen has information displayed constantly (a banner above tabs): <ul style="list-style-type: none"> <li>■ 0: no</li> <li>■ 1: yes</li> </ul> |
| HasNotebook   | Indicates whether the screen has tabs: <ul style="list-style-type: none"> <li>■ 0: no</li> <li>■ 1: yes</li> </ul>                                                    |
| 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<br>This information has no particular use.                                                                                           |
| System        | Indicates whether this object is a system object (cannot be modified): <ul style="list-style-type: none"> <li>■ 0: no</li> <li>■ 1: yes</li> </ul> |
| Label         | The label of the page.                                                                                                                             |
| SQLName       | SQL name                                                                                                                                           |

# Instance PageItem

## Properties

Table 4.27. Properties of the PageItem instance

| Property name     | Description                                                                                                                                       |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| IsVerticalSplit   | Indicates whether this object is a vertical separator for the page: <ul style="list-style-type: none"> <li>■ 0: no</li> <li>■ 1: yes</li> </ul>   |
| IsHorizontalSplit | Indicates whether this object is a horizontal separator for the page: <ul style="list-style-type: none"> <li>■ 0: no</li> <li>■ 1: yes</li> </ul> |
| FrameLabel        | The label of the frame.                                                                                                                           |
| SQLName           | SQL name.                                                                                                                                         |



| Property name | Description                                                                                                                       |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------|
| IsField       | Indicates whether this object is a field on the page: <ul style="list-style-type: none"> <li>■ 0: no</li> <li>■ 1: yes</li> </ul> |
| IsLink        | Indicates whether this object is a link: <ul style="list-style-type: none"> <li>■ 0: no</li> <li>■ 1: yes</li> </ul>              |
| IsFeature     | Indicates whether this object is a feature: <ul style="list-style-type: none"> <li>■ 0: no</li> <li>■ 1: yes</li> </ul>           |
| IsCalcField   | Indicates whether this object is a calculated field: <ul style="list-style-type: none"> <li>■ 0: no</li> <li>■ 1: yes</li> </ul>  |
| 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. |

| Property name | Description                                    |
|---------------|------------------------------------------------|
| 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

| Variable 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. |

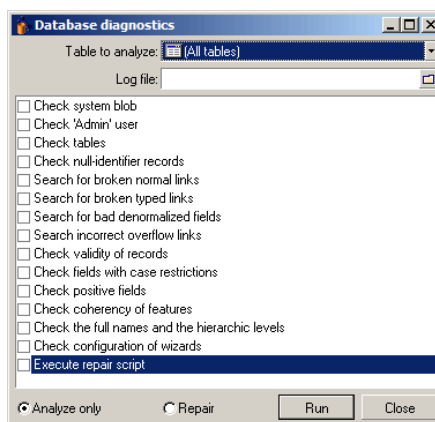
| Variable name      | Description                                                                 |
|--------------------|-----------------------------------------------------------------------------|
| Dbb.Fullname       | Full pathname of the database description file used.                        |
| Dbb.Shortname      | Name (without extension) of the database description file used.             |
| Dbb.Path           | Pathname for the database description file used.                            |
| Dbb.Name           | Name (with extension) of the database description file used.                |
| Dbb.Ext            | Extension of the database description file used.                            |
| Template.Fullname  | Full pathname of the database description template used.                    |
| Template.Shortname | Pathname of the database description template used (without the extension). |
| Template.Path      | Path of the database description template used.                             |
| Template.Name      | Name of the database description template used (with the extension).        |
| Template.Ext       | Extension of the database description template used.                        |



# 5 Diagnostics and repairing 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 **(All tables)** to parse or repair the entire database.

The Log file field enables you to save a log file containing the diagnostics of your database. We recommend you read this file (whose extension is **.log**) with the AssetCenter Log viewer. You can access the Log viewer via the Windows **Start/Programs/Peregrine/AssetCenter Log viewer** menu.

You have two options for this:

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

 **Warning:**

The repairs are conditioned by whether or not you select **Execute repair script**, in the same window. If this option is not activated, only the automatic reparation will take place. If you select 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 the 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 the AssetCenter database (read-only) 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 to 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 required 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                                                                                                                         |
|--------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Security when accessing the AssetCenter database</b>                  | When using reports, an AssetCenter login and password are required to access the database.<br><br>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. |
| <b>DBMS connection parameters</b>                                        | 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.                                                |
| <b>Selecting the connection used to access the AssetCenter database.</b> | The user of the report selects the appropriate connection.                                                                                               | The user directly accesses the database without using the AssetCenter connections.                                                              |
| <b>Link between the DBMS engine and the report</b>                       | The report is independent of the database engine used.<br><br>When you change DBMS, you don't have to modify your reports.                               | The report depends on the database engine.<br><br>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
- Features

All these objects are identified by their SQL names.

---

 **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 connections 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 ODBC Administrator. You select the AssetCenter connection when you create and use the report. The standard connection dialog box in AssetCenter is used for this.

### How to create your own ODBC connections

- 1 Launch 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 in 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 | Archival

## CHAPTER

Archival, as its name suggests, allows you to archive data that you feel is no longer necessary to be kept in the database in full but for which you would like to maintain a trace and keep the associated records in the current database. For example, you might want to archive a record associated with a retired computer that is however still referenced at the accounting level.

---

## Enabling archival

---

### Warning:

In the case of DB2, you must create 8K tablespaces.

► [DB2 UDB \[page 50\]](#)

---

Archival can be enabled both before and after creating the database. In both cases, the procedure is the same. However, when archival is enabled prior to the creating the database, you can specify a particular **tablespace** (logical data storage space) for the archived data in the **Archive** field of the database creation screen. This option is not available for existing databases.

Archival is activated in AssetCenter Database Administrator using the **Database/Enable archival** menu. Upon activation, the following operations are carried out on the database:

- An archival table is created for each table of the database. This table duplicates the structure of the standard table and bears the following SQL name:

```
ar<name of the standard table>
```

 **Warning:**

No archival tables are created for the tables listed below. Therefore, no archives can be created for the records of these tables:

```
SysBlob, amAccessRestr, amAction, amCFAql, amCFScript, amCalcField, amFeatMemo, amFeatParam, amFeatScript, amFeature, amFeatureClass, amFuncRight, amHelp, amHelpMemo, amImage, amLoginSlot, amOption, amProfile, amSysConfig, amUserRight, amViewDef, amFuncDomain
```

In AssetCenter Database Administrator, archival tables (i.e. duplicated and prefixed tables) are not accessible. Their structure being identical to the one of the standard table, they should not be edited or modified. Any modification to a table automatically triggers a symmetrical modification at the level of the associated archival table.

 **Note:**

If archival was enabled prior to the creation of the database, the archival tables are stored in the **tablespace** specified in the **Archive** field of the database creation screen.

- For each foreign key of the archived tables, the following fields are created in the standard table (and in the corresponding archival table):

- ```
ar<SQL name of the foreign key>
```

This field stores the identifier of the archived record.

- ```
arstr<SQL name of the foreign key without the type identifier>
```

This field stores the description string of the archived record.





Note:

Depending on the fields, the type identifier can be one of the following:

- **l: Integer (32 bit)** type field
  - **i: Integer** type field
  - **d: Double-precision number** type field
  - **str: Text** type field
  - **dt: Date and time** type field
  - **b: Boolean (Yes/No)** type field
- 
- An additional connection (<Name of the original connection> - **With archival**) is created for the database.
- Once the enablement is complete, the database is ready for archival.

## Archival principles

Four primary archival operations can be performed on records:

- **Archival:** moving multiple records to the corresponding archival tables. Archived records are deleted from the standard table. The data to be archived erases any previously archived data.
- **Backup:** copying multiple records to the corresponding archival tables. The records are not deleted from the standard table.
- **Purge:** destruction of multiple records from archival or standard tables.
- **Restore:** restoring of multiple records from the archival tables to the standard tables.



Warning:

The records deleted from the archival tables during a purge cannot be restored.

The operations previously described are implemented using extensions of the AssetCenter query language (AQL). The following table details the syntax of the operations and the tables on which they are available.

| Operations | AQL syntax                                         | Available for    |
|------------|----------------------------------------------------|------------------|
| Archival   | ARCHIVE FROM <Clause FROM><br>WHERE <Clause WHERE> | Standard tables. |

| Operations | AQL syntax                                          | Available for                 |
|------------|-----------------------------------------------------|-------------------------------|
| Backup     | BACKUP FROM <Clause FROM><br>WHERE <Clause WHERE>)  | Standard tables.              |
| Purge      | PURGE FROM <Clause FROM> W<br>HERE <Clause WHERE>)  | Standard and archival tables. |
| Restore    | RESTORE FROM <Clause FROM><br>WHERE <Clause WHERE>) | Archival tables.              |



Note:

For more information on AQL and on the syntax of the FROM and WHERE clauses, refer to the **Advanced use guide**, chapter **AQL queries**.

## Viewing archived data

Simple links (1-n cardinality) pertaining to archived records appear grayed out when referenced within a detail screen.

## Example

### Example of data archival

The following example presents the archival of closed purchase requests that have not been modified in the last 90 days. In this case, the solution is to archive those records from the **amRequest** table for which:

- the date referenced in the **dtLastModif** field is more than 90 days old,
- the value of the **seStatus** field is **41**.



Note:

Values in itemized lists are associated with an integer. The value that interests us is the one corresponding to the value **Closed**, which is 41. To access the values in a system itemized list, place the focus on the field populated by the list and simultaneously press Shift and F1. The context-sensitive help for the field is displayed and contains the list of values for the system itemized list and their associated integers.

To solve the issue raised by this example, we will therefore create an action for the **amRequest** table. This action will execute an AQL archival query on the

records matching the previously defined criteria (specified in the WHERE clause of the query).

- 1 Select **Tools/ Actions/ Edit** and click **New** to create a new action.
- 2 Choose a **Name** for the action and select the **amRequest** table as the **Context** of the action. The type of the action is **Script**.

 **Note:**

The definition of a **Context** is not mandatory. It is even forbidden if you choose to use a workflow to execute this action periodically.

- 3 Enter the following script in the **Script** tab of the action:

```
RetVal = amDbExecAql("ARCHIVE FROM amRequest WHERE dtLastModif < AddDays(getdate(), -90) AND seStatus = 41")
```

 **Note:**


The **amDbExecAql** function queries the database. For more information on this function, refer to the **Programmer's Reference**.

- 4 Validate the creation of the action.
- 5 You can call this action from a workflow or execute it directly from the purchase requests screen.

To control the archival:

- 1 Open the purchase requests screen using the **Procurement/ Purchase requests** menu.
- 2 Right-click the list of purchase requests and select the previously created action from the shortcut menu.

Records corresponding to a closed purchase request and that have not modified for more than 90 days are archived and disappear from the list.

However, if they are referenced by other records, they will appear grayed-out in the detail of those records. If you have administrative rights, you can then display the archived record by clicking .

 **Note:**

This example can be found in the datakit.

## Example of data restoration

If you want to restore a purchase request archived by mistake:

- 1 Connect to the database using the archival connection.
- 2 Select **Tools/ Actions/ Edit** and click **New** to create a new action.
- 3 Choose a **Name** for the action and select the **arRequest** table (archival table of the **amRequest** table) as the **Context** of the action. The type of the action is **Script**.

---

 **Note:**

The definition of a **Context** is not mandatory. It is even forbidden if you choose to use a workflow to execute this action periodically.

- 4 Enter the following script in the **Script** tab of the action:

```
RetVal = amDbExecAql ("RESTORE FROM arRequest WHERE lReqId="&[lReqId])
```

---

 **Note:**

The **amDbExecAql** function queries the database. For more information on this function, refer to the **Programmer's Reference**.

- 5 Validate the creation of the action.
- 6 Open the archived purchase requests (**arRequest**) screen using the **Administration/ List of screens** menu, and then select the archived request.
- 7 Right-click and select the previously created action from the shortcut menu. The record is retrieved and deleted from the archive and is transferred back to the production database.

## Processing of linked records

During archival operations, the way the linked records are handled depends on the type of the link. The following table details the behavior of the archival by link type.

| Link type        | Processing                                                                                                                                                                                                                                                                                                                                                                           |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| OWN              | <p>Archival operations handle linked records as archived records. If you archive a record, all its linked records will also be archived. If you purge a record, all its linked records will also be purged.</p> <p>For example, an asset and all its sub-assets are archived, copied, restored or purged simultaneously.</p>                                                         |
| DEFINE or NORMAL | <p>Foreign keys in linked records are reset to zero. Archival fields (arl&lt;Name&gt;Id and arstr&lt;Name&gt;) are respectively populated with the identifier of the archived record and with its description string.</p> <p>For example, if you archive an expense line, the corresponding foreign key in the record of the asset linked to this expense line is set back to 0.</p> |

## Case of features

It is useful to specify the behavior of the archival function when features are involved.

If we consider a record **A** and its associated feature **C** (with a **Type** other than **Link**):

- If **A** is archived, **C** is also archived.
- If **C** is archived, the link between the record **A** and its feature **C** is broken and the record no longer appears in the detail of the record. This link is restored in the two following cases:
  - 1 **A** is subsequently archived. In this case, the link between **A** and **C** is re-established in the archived data.
  - 2 **C** is restored. In this case, the link between **A** and **C** is re-established in the database.

If we consider a record **A** and its **Link Type** associated feature **C**, and **D**, another record to which the feature **C** points:

- If **A** is archived, **C** is archived. **D** is not automatically archived.
- If **C** is archived, the link between **A** and **C**, as well as the link between **C** and **D** are broken. The feature no longer appears in the detail of the record **A**. The integrity of the link between **A**, **C** and **D** is restored in the following two cases:

- 1 **A** and **D** are subsequently archived. In this case, the relationship between **A**, **C** and **D** is re-established in the archived data.
  - 2 **C** is restored. In this case, the relationship between **A**, **C** and **D** is re-established in the database.
- If **D** is archived, the link between **C** and **D** is broken. The integrity of the links between **A**, **C** and **D** is restored in the two following cases:
    - 1 **A** is subsequently archived. In this case, the relation between **A**, **C** and **D** is re-established in the archived data.
    - 2 **D** is restored. In this case, the link between, **A**, **C** and **D** is re-established in the database.

---

## Disabling archival

Archival can be disabled at any given time. Disabling archival implies that it has been previously enabled and that the database has been modified to contain the additional tables dedicated to archival.

### Disabling archival

To disable archival, clear the **Database/ Enable archival** option.

---

 **Warning:**

Disabling archival destroys the archival tables and erases all archived data.

---



# 9 Denormalization

## CHAPTER

The AssetCenter database is highly normalized. This database property consists of eliminating unnecessary redundancies and dependencies between tables in order to obtain a usable and rational relational database.

Denormalization can be considered the opposite process. A value is stored in several places, thus creating voluntary redundancies in order to accelerate data processing and retrieval, and to improve performance in general.

---

## When is it advisable to denormalize a value?

There is no absolute rule for this matter. Some factors, though, may help you determine whether it is worth considering denormalizing your database:

- The metrics of your database. If the database is small to medium in size, denormalization has no significant advantages. In some instances, denormalization may even have a negative impact on performance. For instance, denormalization could significantly increase the size of tables; thus, even though data retrieval becomes faster, record updates may be slowed down.
- The most critical queries used and the expected response time. When these queries use data that is distributed over multiple tables, and/or when calculations are performed on one or more columns before the queries return

the response, it may prove useful to denormalize the values returned by these queries.

- Performance. If you consider that performance of the application is sufficient, it is not necessary to apply denormalization.

---

## Denormalization principles in AssetCenter

In theory, there are several denormalization techniques available. AssetCenter only accepts one of them: column denormalization. A column, i.e. a field, is repeated in several tables in order to prevent the creation of joins between tables in queries.

As an example, let's take the case of a report on employee absences. This report is produced very often. Most of the data required to populate the report is stored in the Absences table. One data item used in the report, the email address of the employee supervisor, is stored in another table, however: the Employees table. It might be worth considering denormalizing this data in order to avoid a join between the two tables when you create the report.

After denormalization of this value has been carried out, the field is duplicated in the Absences table. This operation emphasizes one of the potential risks of classic denormalization, i.e. maintaining redundant data. AssetCenter eliminates this risk since it automatically maintains normalized data integrity.

Generally speaking, denormalizing a value is an operation that is fully transparent for the user. The duplicated value never appears in a details screen and the rules used to write an AQL query remain unchanged. AssetCenter detects automatically the presence of a denormalized value and simplifies queries on the fly.

---

 Note:

Fields created through denormalization are visible in AssetCenter Database Administrator though. Their SQL names use the **df** prefix. Links use the **ln** prefix.

---



## Denormalizing a value

Once the appropriateness of denormalization has been validated, you may use AssetCenter Database Administrator to carry out the operation for real.

- 1 In AssetCenter Database Administrator, connect to your database
- 2 Select the denormalized field's or link's target table and the select the **Database/ Add a denormalization** menu. The following screen prompts you to select the field to denormalize:

---

 **Note:**

If you check the **Create an index on the denormalization field** box, the sort operations on this value will be accelerated.

---

Click **Create**.

- 3 Save your changes and follow the steps in the database update wizard (select a log file, select an optional configuration file, etc.) and then click **Finish**.
- 4 The denormalized field or link is now active. It will be automatically used by queries when the context allows.

---

## Example

Let's consider the previous example in which a report on employee absences is created. Because this report and the data it contains are retrieved frequently, it is appropriate to denormalize the field that stores the email address of the department supervisor.

The **Manager.Email** remote field needs to be denormalized. To do so:

- 1 In AssetCenter Database Administrator, select the Absences table (**amAbsence**) and the select the **Database/ Add a denormalization** menu.
- 2 In the screen that appears, select the **Manager.Email** remote field from the drop-down list. This list displays the fields and links of the Absences table and enables you to select remote fields that are part of a link. In our example, you should unfold the **Supervisor** link, and then choose the associated **Email** field.
- 3 Click **Create**. You should see a new field appear in the Absences table with the **df** prefix. Select the **File/ Update database structure** menu to validate

your changes. AssetCenter Database Administrator displays a database structure update wizard.

- 4 Click **Next** and then select a folder for the log file.
- 5 Click **Next**. The following page of the wizard prompts you for an optional conversion file. This page is not used for denormalization.
- 6 Click **Finish**. The wizard displays the operations being processed and informs you when the database has been successfully updated. The queries used in the report will use the denormalized field, thus avoiding the usage of a join between the Absences and the Employees tables.

# 10 | Import

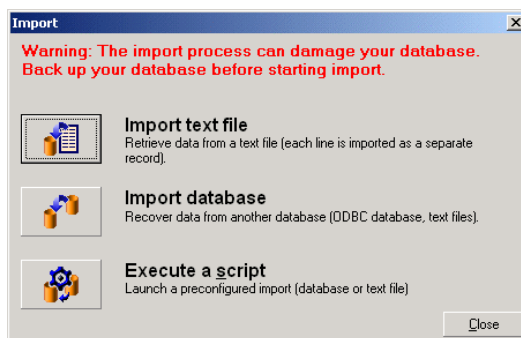
CHAPTER

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 multiple 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.
- Multiple 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 multiple 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.

---

## 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" fields as reconciliation keys

We advise against using the "ID" fields of tables as reconciliation keys if you want to re-import 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 concurrent access to the AssetCenter database

We advise against performing imports concurrently 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 Monetary | <p>The source field must be a number.</p> <p>If the source field originates from a structured database and the field type is "Number" or "Monetary" then there is no constraint.</p> <p>If the source field is a "Text" type field, you must present a number, use the period "." as a decimal separator and do not use a separator for thousands.</p>                                                                                                                                                                                                                                                                               |
| Boolean (Yes/No)   | The source field must display "1" (for "Yes"), or "0" (for "No").                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Date               | <p>If the source field originates from a structured database and the field type is "Date" or "Date and 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"> <li>■ The date format (order of year, month and day) must be the same for all records. Specify this format when importing.</li> <li>■ You must systematically enter a day, month and year.</li> <li>■ Use the same separator (of your choice) to separate days, months and years for all records. Specify this separator when importing.</li> </ul> |

| If the value is:        | Then:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 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><b>Example</b></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>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Duration                | <p>You must respect the constraints defined by the "UserType" and "UserTypeFormat" properties when the UserType is "Duration".</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Table or field SQL name | <p>You can import any alphanumeric 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 and time                  | <p>If the source field originates from a structured database and the field type is "Date and time" or "Date", 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 and 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 "Number", "Text", 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 and 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 **Open**.
- 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 **Open**.

# 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 **OK**, 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.

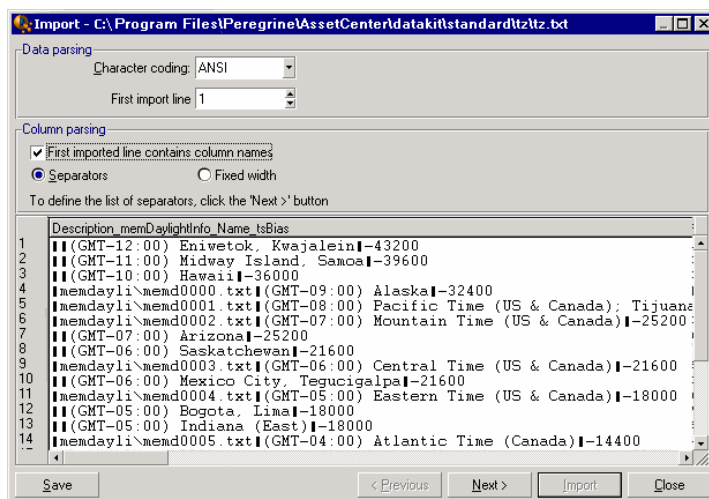
Specify this character in the following screen.

## 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 **Next**.

## Column separators

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

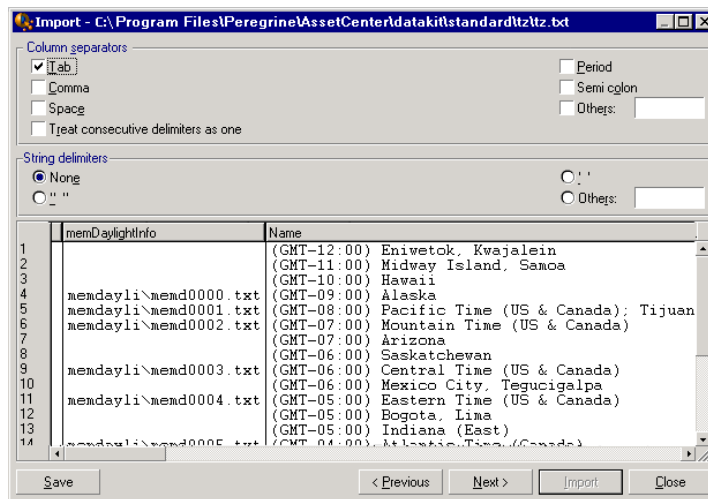
If you check the **Treat consecutive delimiters as one** option, AssetCenter will treat identical consecutive delimiters as one and will not create empty columns. If you want to create an empty column, uncheck this box use two consecutive delimiters in your text file.

## 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 to you 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 **Next** 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 **Next** 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 **Next** 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

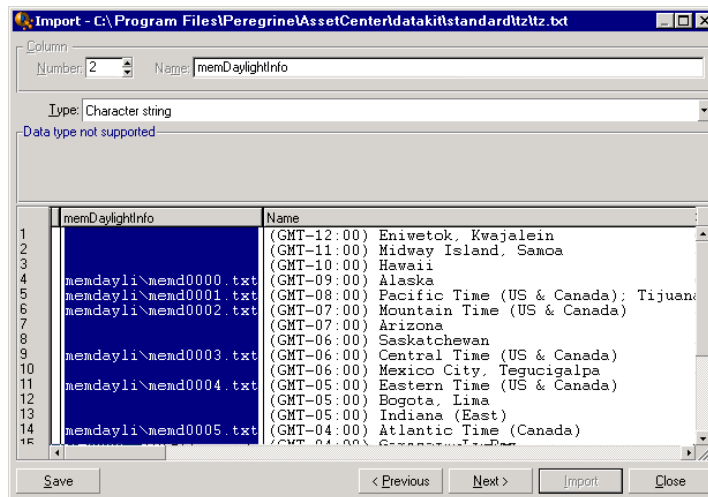
Enter the field type to be imported. AssetCenter lets you choose from the following possibilities:

- **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 **Next** 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 **OK** 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 **OK** 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 re-import 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.

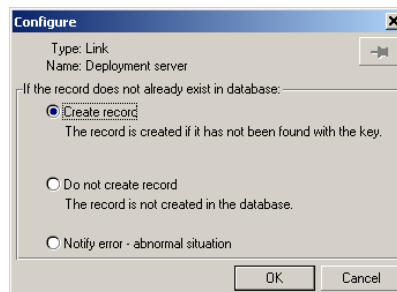
## Configuring the creation of linked records

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.

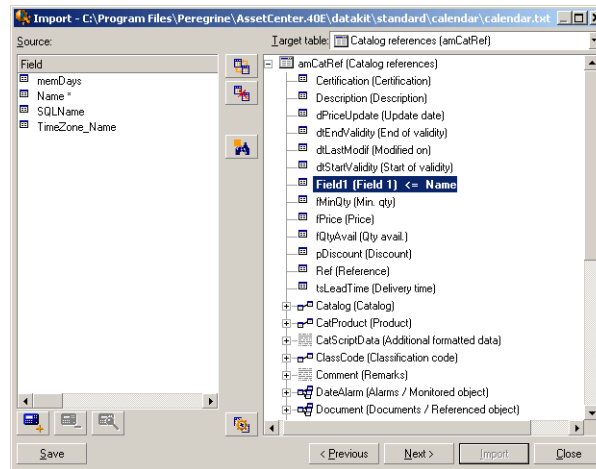
The effect of the pushpin depends on the types of links:

- 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:**

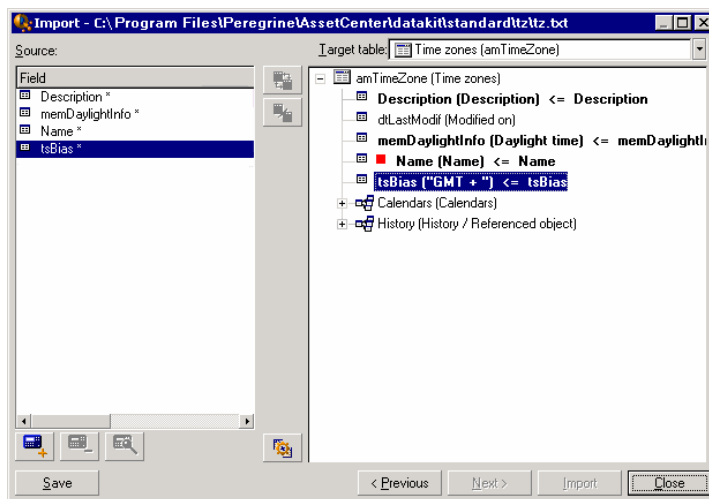
AssetCenter uses three kinds of links between records: n links: For example, a portfolio item can only be linked to one location, but a location can be linked to several portfolio items. 1 links: For example, a portfolio item can only be linked to one comment and a comment can only be linked to one portfolio 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:

- Long text field
- Binary field


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.

---

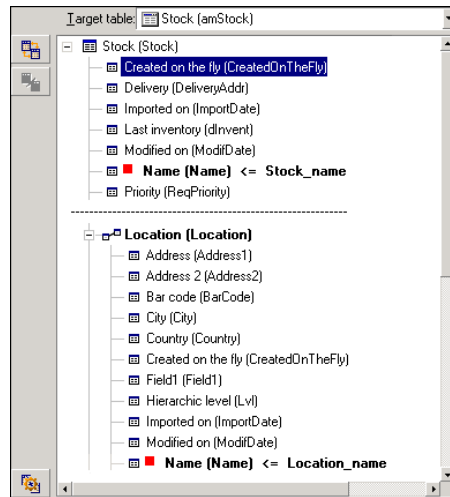
### Importing documents

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



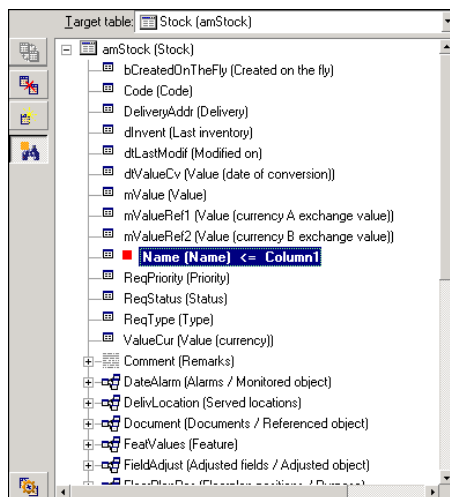
In this example, a stock is identified by two main keys:

- 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.

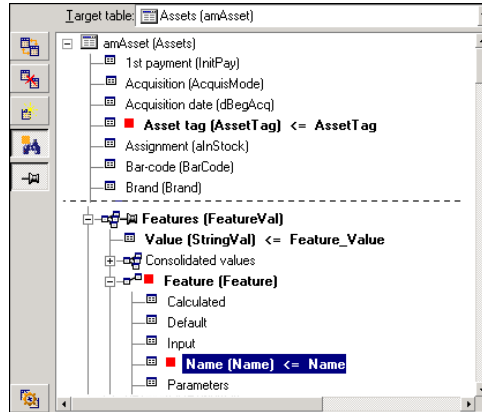
## Example 3: Keys that update feature values in database records







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.

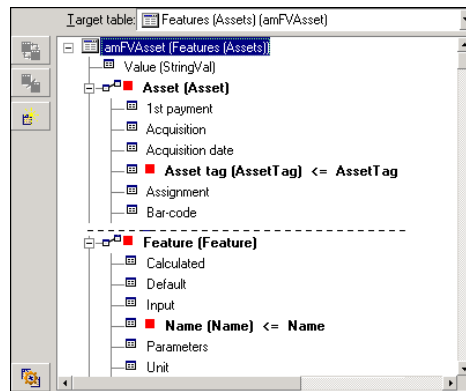
You can use either of these two methods:

### 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):

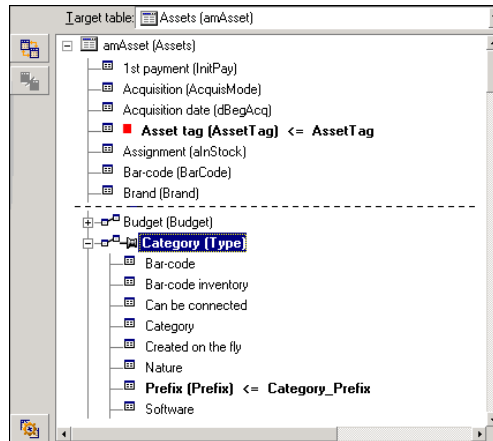






- 1 The ( ■ Feature (Feature) \*, ■ Asset (Asset) \*) pair identifies the feature associated with the asset.
- 2 The key on ■ Name (Name) indicates which is the key field of the ■ Feature (Feature) \* link.
- 3 The key on ■ Asset tag (AssetTag) indicates which is the key field of the ■ Asset (Asset) \* 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  Model (Model) 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 **Next** 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 **Import** 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 **Save** 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 **Save** 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 **Close** 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 Enter the full name of the script file.
- 4 Enter the password associated with the ODBC source.
- 5 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

```
amimpl [-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 (including MSDE) | Sybase SQL Server        |
|-------------------------|------------------|--------------------------------|--------------------------|
| <b>EngineName</b>       | Oracle           | ODBC                           | Sybase                   |
| <b>DatabaseLocation</b> | Server name      | Data source name               | ServerName:Database-Name |
| <b>User</b>             | Account name     | MS SQL Server user name        | Account name             |
| <b>Password</b>         | Account password | MS SQL Server 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 | Controlling access to the database

## CHAPTER

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

---

 Note:

Access rules can only be managed by database administrators.

---

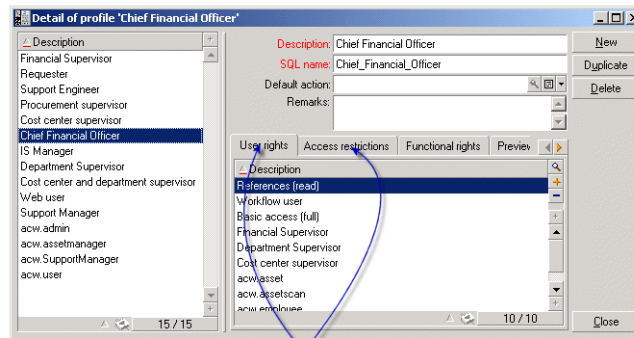
Use the **Administration/ User profiles** menu to display the list of user profiles.

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



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

## Importance and overview of database access rights

AssetCenter is a program that can be used concurrently 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 concurrent 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 in access management

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 rights screen, used to regulate access to the AssetCenter database, is organized as follows:

- Rights are displayed in a column.

| Object                                                       | Rights (RIU/CD) |
|--------------------------------------------------------------|-----------------|
| + Problem types (amProblemClass)                             | R-/..           |
| + Problem types - Employee groups relation (amRelGrpPbClass) | R-/..           |
| + Product options (amProdOption)                             | r-/..           |
| - Products (amCatProduct)                                    | rii/..          |
| + (Fields)                                                   | R..             |
| + (Links)                                                    | R..             |
| - (Features)                                                 | Riu             |
| + Multiplier factor (fv_OverheadFactor)                      | RIU             |
| + Unit period of time (fv_UnitDuration)                      | R..             |
| + (Calculated fields)                                        | r..             |
| + Progress statuses (amDaDepProgress)                        | .../..          |
| + Projects (amProject)                                       | r-/..           |
| + Purchase order lines (amPOrdLine)                          | .../..          |
| + Queries (amQuery)                                          | .../..          |
| + Receipt lines (amReceiptLine)                              | .../..          |
| + Receiving slips (amReceipt)                                | .../..          |

- 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.

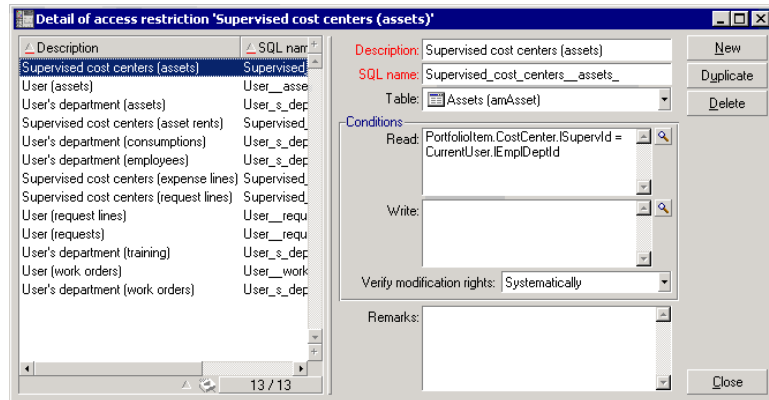
## Defining access restrictions

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.

---

 **Tip:**

If you use an LDAP server to manage the passwords, the password entered into the AssetCenter database is not accounted for, with the exception of the **Admin** user.

---

- 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 using 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 AssetCenter API.

---

 Note:

The login types do not take into account connections to the database made via AssetCenter Export, AssetCenter Server, or 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**.

## Length of validity of the connection

It is possible to define a length of validity for a given connection. This is defined in the **Time-based access control** frame.

The length of validity corresponds to the dates given in the **Start of validity** and **End of validity** fields. You can associated a local calendar with a the specified dates (**Calendar** field).

## How connection slots work

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

For as long as the user is connected to the database, the connection slot is regularly updated by AssetCenter. The update frequency is defined in the option **Connection slot update delay (minutes)** in the "Access control" section of 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 **Delete** 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 of AssetCenter. To do this:

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

---

 Note:

Only the administrator, using the "Admin" login, can delete connection slots. This function is not available to other AssetCenter administrators.

---

### 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 **Enable 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.

The password that controls users' access to the AssetCenter database can be stored in several different ways:

- Directly in the AssetCenter database.
- In an external LDAP directory.
- Using Windows NT security.

Password management in AssetCenter is task reserved for the administrator.

## Definition of user passwords

You must be connected to the AssetCenter database as the administrator in order to create passwords for anyone wishing to connect to the database.

Each person in the Departments and Employees table (amEmplDept) can have a password assigned. This password is defined in the **Profile** tab.

A password is created in the **Password administration** frame. This password is associated with the user profile (**Profile** frame) and a period of validity (**Time-based access control** frame).

### Define a user profile

You must define a profile for the user connecting to the database before assigning a password.

The user profile is defined by the following fields:

- Access type: Refer to section [Managing user connections](#) [page 174] in this chapter.
- User account status: Active, Locked, or Disabled.
  - Active: The user can connect to the database.

- **Locked:** The user is temporarily denied access to the database, for example, after giving an incorrect password three times in a row. Locking is for 15 minutes.
- **Disabled:** The user may not connect to the database.
- **Action on login:** Action executed on connecting to the database

## Password and validity of connection

The validity period has precedence over the status of the user account (active, locked or disabled). If the validity period expires and the user account is active, the named user will not be able to connect.

For further information on defining connection periods, refer to section [Length of validity of the connection](#) [page 175] of this chapter.

## Define a password

Use the following fields to define a user password:

- **User name:** Enter the user name, which is not necessarily the name of the person.
- **Password**

The password may be modified by the administrator or by the current user, depending on the selected options.

The following options are associated with the password:

- **Force change:** This option forces the user to change their password at the next connection
- **Non-modifiable:** This option stops the current user from changing their password; Only the administrator is authorized to do this
- **Never expires:** This option stops the password from expiring
- **Expiration date:** This option enable you to define an end of validity date for the password. The default validity is defined at the level of the database options (**Administration/ Database options** command).

For more information on the database options concerning password management, refer to chapter [Managing passwords](#) [page 178].

## Administrative options associated with password management

### Password administration rules

You must respect a specific order for defining passwords:

- 1 Creating or modifying the script library associated with the password (if you do not wish to use the default library)
- 2 Populating the password options using the password configuration wizard
- 3 Defining the database options

## Password configuration wizard

To access the password configuration wizard, select **Tools/ Actions**.

This wizard enables you to define basic options concerning a password (validity, notification before expiration) as well as the options called by the script library (minimum password length, minimum number of different characters).

The wizard has a set of default values. These values are saved in the Options table and may be modified.

The wizard automatically loads the script library defined in the database options and saves the values populated in the Options table.

## Database options

The administrator must define the password management options. Use the **Administration/ Database options, Passwords** sections.

- Password validity (expressed in days).
- Password historization: User passwords are stored in the database and users may not use the same password several times.
- Warning before password expiration (delay in days).
- Library associated with passwords.

These options are linked to the values defined with the password configuration wizard.

## Library associated with the password

The rules for entering a password are defined in a script library. The default library is called **PasswordLib**; it can be modified (**Tools/ Script library**).

---

### Important:

The Basic function used in the script in the library must have the same name as the library itself.

- 
- The password must contain a minimum number of characters
  - The password may not contain several times the same letter
  - The password may not contain the user name



The default script imposes the following rules on users registering passwords:

## Particular case of the Admin login

The **Admin** login is always stored in the AssetCenter database, whether the other passwords are stored in the AssetCenter database or in the LDAP directory.

---

### Warning:

Do not confuse the **Admin** login with a user with **Administration rights**.

---

## How the Admin login works

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 cannot connect using 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:

Warning: Anyone who knows the **Admin** 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 with this login name and using the **Tools/ Change password** menu.

## Admin login password

If despite the precautions specified above you forget this password, there is no way of recovering it using AssetCenter. Call Peregrine Systems support.

# Solution 1: Storing passwords in the AssetCenter database

## Creating the password

The password is stored in the **Password** field (LoginPassword).

## Modifying the password

### Modifications performed by the user

Every AssetCenter user can modify their password by opening the database with 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

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.

# Solution 2: Storing passwords in an LDAP directory

## Reminders about LDAP

LDAP (Lightweight Directory Access Protocol) is a directory-interrogation protocol.

Certain messaging servers, such as Microsoft Exchange, use this protocol to expose their directory.

## General rules about stocking passwords in an LDAP directory

If your company manages its employees' passwords centrally in an LDAP directory, you might find it useful to use the directory to control access to the AssetCenter database.

You will also be able to take advantage of other password management features offered by the LDAP directory, such as:

- Mandatory periodical modification.
- Verification of format and number of characters.
- Etc.

## Implementing an LDAP directory

- 1 Make sure that all AssetCenter users can access the LDAP directory.
- 2 Create the AssetCenter users in the LDAP directory and assign them a login and password.
- 3 Deactivate the **Anonymous** access to the LDAP directory.
- 4 Launch AssetCenter.
- 5 Connect to the working database via the **File/Connect to database** menu.
- 6 Display the database options via the **Administration/Database options** menu.
- 7 Populate the options having the section name **LDAP authentication**.
- 8 Create or modify the users of the **Employees** table (amEmplDept) by populating the fields in the **Profile** tab.

In particular, populate the following fields:

- **Login** (UserLogin): login that connects the user to the AssetCenter database.
- Possibly the field that stores the corresponding LDAP login, if it's different from the **UserLogin** field.

Examples of LDAP logins:

- Microsoft Exchange: **CN=<identifier>, CN=<DOMAIN NAME> or DC=<DOMAIN NAME>, CN=<identifier>**
- Lotus Domino: **<FirstName> <Name>/<DOMAIN NAME>**



Tip:

You can import the LDAP directory using Connect-It.

- 9 Make sure that the LDAP DLLs are installed on the AssetCenter user computers. This is the case if you selected the **Full installation** option when you installed the AssetCenter client.  
This is also the case if you selected the **Customized installation** option, and chose **LDAP authentication**.  
This is not the case, however, if you selected the **Standard installation** option.
- 10 Have the users reconnect to the database in order for the new options to be taken into account at the level of their AssetCenter client.

## LDAP connection to the AssetCenter database

After you implement the LDAP authentication in the AssetCenter database, the users will connect by:

- 1 Launching AssetCenter.
- 2 Connecting to the working database via the **File/Connect to database** menu.
  - Login: This is the login that exists in the **Employees** table (amEmpIDept) in the **Login** (UserLogin) field.  
AssetCenter finds the login in the LDAP directory. It can do this because of its settings, configured via the **Administration/Database options** menu.
  - Password:
    - Microsoft Exchange: Windows password.
    - Lotus Domino: Internet password defined in Domino.
 The validity of this password is verified through the LDAP server.

## Secure connection (SSL) to the LDAP server

To secure access to the LDAP server, two options are available in AssetCenter.

- 1 **LDAPCertFile**: This option requires a single OpenLDAP server and uses a single authentication certificate.

To enable this option, the **LDAPCertFile** file must point to a text file containing a certificate in the **PEM** with the following syntax:

```
-----BEGIN CERTIFICATE-----
... (CA certificate in base64 encoding) ...
-----END CERTIFICATE-----
```

- 2 **LDAPCertDir**: This option, which is more complex, requires multiple OpenLDAP servers and uses multiple authentication certificates.
- 



This option is used more often for Connect-It.

---

To enable this option, the **LDAPCertDir** file must point to a directory containing certificates whose names are based on the hash code. Each file in the folder must be in the **PEM** format and contain the subject of each certificate:

Use the **c\_rehash** command to enable the hash function in the current directory and create links in Unix.

---



Enabling a secure connection to the LDAP server requires all computers to be configured before connection.

---





# 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.

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 in 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.

## From the DOS command prompt

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

```
amsrv -cnx:<connection name> -login:<login> -password:<password for login>
```

### 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 in 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 an NT service).

To connect AssetCenter Server to the database, use one of the following:

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

### Disconnecting AssetCenter Server from a database

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

- The **File/ Disconnect from database** 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 **amsrv1** 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.

Location of this file: ► **AssetCenter - Installation** guide, chapter **.ini and .cfg files**.

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

## Using the integrated security of Windows 2000, XP, or Server 2003

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 **NT** 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 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 User Manager

The fields to populate are:

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



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

- ▶ Add NT users to the database module (AddUser) [page 204]

## 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.

---

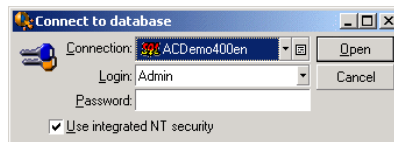
## Step 5: Activate the automatic triggering of the AssetCenter Server modules

- 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 using 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 domains: Windows 2000 and Windows Server 2003 Active Directory are supported.
- Windows clients: Windows 2000, XP and Server 2003 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.

---



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.

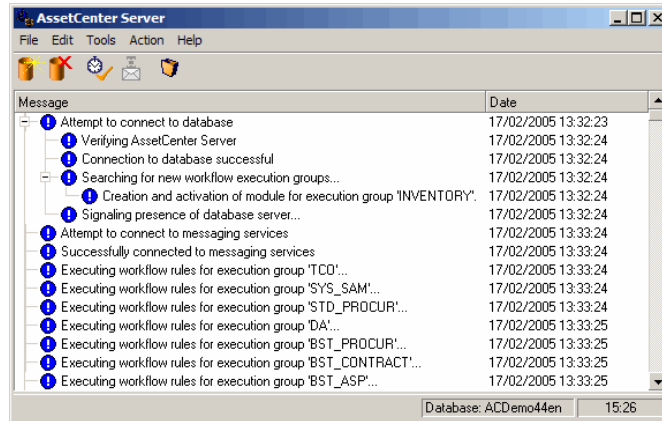
## If you exit AssetCenter Server

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                                                                                                                                                                                        |
|-------------------------------------------------------------------------------------|---------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | <b>File/ Connect to database</b>      | Connect to a database.                                                                                                                                                                          |
|  | <b>File/ Disconnect from database</b> | Disconnect from the database.                                                                                                                                                                   |
|  | <b>Action/ Activate</b>               | Selectively activate deadline monitoring agents.                                                                                                                                                |
|  | <b>Action/ Empty list</b>             | Clear messages from main screen.                                                                                                                                                                |
|  | <b>Action/ Connect to messaging</b>   | Try to connect to external messaging systems.<br>This button is grayed out if the connection is successful.<br>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 **Tools/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 **Tools/ 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 <b>Days</b> field.<br><br><b>Example</b><br>Every Monday                                                                                                                                                                                                                                                    |
| "Day of the year:"      | A day or selection of days. These are defined via the <b>Day</b> , <b>Month</b> and <b>Year</b> check boxes.<br><br><b>Example</b><br>July 20                                                                                                                                                                                                                                           |
| "The first"             | The days of the week defined using the <b>Day</b> check box, and for the month(s) and year(s) defined using the <b>Month</b> and <b>Year</b> check boxes.<br><br><b>Examples</b><br>"The first" Friday of each month.<br>"The second" Monday of the month of September.<br>"The next to last" Wednesday of the month of November.<br>"The last" Tuesday of every month throughout 2000. |
| "The second"            |                                                                                                                                                                                                                                                                                                                                                                                         |
| "The third"             |                                                                                                                                                                                                                                                                                                                                                                                         |
| "The fourth"            |                                                                                                                                                                                                                                                                                                                                                                                         |
| "The next to last"      |                                                                                                                                                                                                                                                                                                                                                                                         |
| "The last"              |                                                                                                                                                                                                                                                                                                                                                                                         |

## 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 of time, 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:

## 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.

## 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**.

## Calculate rents and loans module (Rent)

The **Calculate rents and loans** module generates or recalculates the expense lines:

- Contract-level rent payments
- Loan payments

Configuring the **Calculate rents and loans** module defines:

- the number of days for which expense lines are generated for rent payments.
- the frequency with which the module is executed

### Overview

#### Rent

AssetCenter Server verifies at the defined intervals whether it needs to generate or recalculate expense lines. If this is the case, it does so.

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 (prorated or allocated) 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 (prorated or allocated) 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.

## Loan payments

AssetCenter Server regularly checks whether expense lines require generating or recalculating for all loan amortization schedules.

## Parameters

---

 **Note:**

This section only applies to rent payments; it does not apply to loans.

---

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 sets 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.

Location of this file: ► **AssetCenter - Installation** guide, chapter **.ini and .cfg files**.

---

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, 2004 through July 1, 2007.
- 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, 2005, AssetCenter Server is launched for the first time: it generates:

- Past rents from July 1, 2004 through July 1, 2005.
- The present rent on July 1, 2005.
- The projected rents from August 1, 2005 through July 1, 2006.

Following these calculations, the **Recalculation effective from** field indicates the date of the last non-projected expense line, i.e. July 1, 2005.

AssetCenter Server runs in the background: 2 months later on September 1, 2005, it generates:

- The projected rents from October 1, 2005 through September 1, 2006.
- 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, 2005 through October 1, 2005.

## 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 the **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.

# Create assets, consumables, 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/ Let AssetCenter Server create the items received in the portfolio** 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.

# Execute workflow rules for execution group modules

Once a workflow execution group (Example: **ADMIN**) 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.

## 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 Desktop Inventory (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 Inventory** connector in the **Scenario diagram** window by clicking on the title bar of the **Desktop Inventory** 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 Desktop Inventory 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 Inventory** connector, refer to the Connect-It **User's guide**, chapter **Peregrine Systems connectors**, section **Desktop Inventory 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 documentation for the **iddac.scn** scenario, refer to the Connect-It **User's guide**, chapter **Scenario documentations**, section **Creating a scenario documentation**.

## Update the database with the results of the scanners module (PdiAc)

This module enables you to program the recovery of **.xml.gz** files by Desktop Inventory. These files store the results of each computer scan.

The folder that contains the **.xml.gz** files is specified at the level of the Connect-It scenario **addcpu.scn**.

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 (PdiAc)**.

- Parameters of the Connect-It **addcpu . scn** scenario, which is located in the **scenario\ntsec\ntacxx** (where xx is the version number of AssetCenter) sub-folder of the Connect-It installation folder.

## Parameters of the module **Update the database with the results of the scanners (PdiAc) (User data field)**.

Here is the parameter line by default:

```
"$connectit_exedir$/conitsvc.exe" -once -wpplog '$connectit_exedir$/../scenario/pdi/PdiAc41/PdiAc.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/pdi/PdiAc41/PdiAc.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 **PdiAc.scn** scenario.

You do not need to modify these parameters.

## Parameters of the **PdiAc . scn** Connect-It scenario

To modify the **PdiAc . scn** scenario:

- Execute the Connect-It scenario editor.
- Open the **PdiAc . scn** scenario, which is located in the **scenario\pdi\pdia41\PdiAc . scn** sub-folder of the Connect-It installation folder.

- Select the **PDI** connector in the **Scenario diagram** window by clicking the **PDI** title bar (but not anywhere in the box).
- Select the **Tools/ Configure** menu.
- Click **Next**.
- Select the processing mode (read or write).
- Click **Next**.
- Select your connection protocol. The protocol used by default is **Local or network file(s)**.
- Click **Next**.
- Specify the location of the scan files or folder in the **Folder name** field.

To find out which information is populated in the AssetCenter database, examine in detail the **PdiAc.scn** scenario.

## 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.

## 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 (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 (hours)**, accessible via the **Administration/ Database options** menu in the AssetCenter application.

## 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.

## Signal presence of 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 in 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.

## 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**).

# 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** (SQL name: 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.

## Trigger the periodical chargeback rules module (CbkTimer)

This module is used to trigger the chargeback-system rules.



---

 **Note:**

The chargeback rules are used to automate the creation of chargeback expense lines.

---

AssetCenter Server checks at regular intervals, as defined in the **Trigger the periodical chargeback rules** module, whether to verify the chargeback system scheduler.

This is determined by comparing the value of the **Last execution of the rule** field (dtLastExec) in the chargeback systems with the execution frequency of the scheduler.

After executing a chargeback rule, AssetCenter Server stores the date and time of execution in the **Last execution of the rule** field (dtLastExec).

If an expense line cannot be generated, AssetCenter Server creates a chargeback event. You can consult the list of chargeback events in the **Events** tab of a chargeback system and identify the cause.

To configure the **Trigger the periodical chargeback rules** module, select the **Enabled** option, and populate the triggering calendar of the module; The **User data** field does not need to be populated.

## 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 exists, 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.

## 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.

---

## 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 in AssetCenter Server.

Indicate which verifications to carry out by checking the appropriate boxes.

# Administering AssetCenter Server via the Web

The installation program of AssetCenter in Windows 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:

- Starting AssetCenter Server as an NT service.
- Accessing the AssetCenter Server service via the Web.
- Driving the AssetCenter Server service via the Web.

## Starting 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:**

To correctly install the AssetCenter Server service, we recommend you proceed in the following manner:

- 1** Create a user account in Windows (on the computer where this service is installed).

This account must have the right necessary to start the AssetCenter Server service.

The environment in which this account is set up must allow you to use client layers of the DBMS installed on the computer where AssetCenter Server service is installed.

This is a reminder that the local system account only accesses by default the system environment variables.

- 2** Install the AssetCenter Server in this account.
-

## 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 the **amsrvcf.ini** file.

Location of this file: ► **AssetCenter - Installation** guide, chapter **.ini and .cfg files**.

- 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 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 \$\$SoftName; 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.

Location of this file: ► **AssetCenter - Installation** guide, chapter **.ini and .cfg files**.

---

## 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.

Location of this file: ► **AssetCenter - Installation** guide, chapter **.ini and .cfg files**.

---







# 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.

---

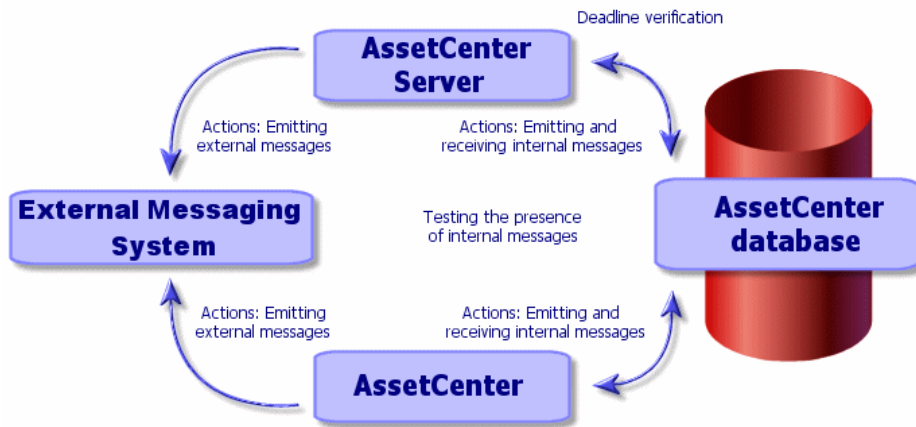
## Overview of messaging

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@address.domain], SMTP: [name2@address.domain]
```

For example:

```
SMTP: jdoe@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 a user profile form with the following details:

- Name:** Hartke
- First:** Richard
- Title:** Building Maintenance
- Mr/Mrs:** Mr.
- Department:** Administration
- Messaging tab:**
  - Account:** SMTP:hatke
  - Password:** [masked]

- Populate the **Email** field of the **General** tab with the following syntax:

SMTP: [name@address.domain]

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.:

EMail: SMTP:rhartke@taltek.com

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 **Bcc** fields, if you wish) with the appropriate address, in the following manner:

SMTP: [name@address.domain]

Or using a calculated string.

Name: Email

Context: Employees (amEmpIDept)

Type: Messaging Preview...

Description Messaging Features History

Referenced object:

Priority: Normal

Acknowledgment

To: [E.Mail]

Cc:

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. The administrator is put in copy.

## win.ini file (Windows) or mail.ini (Unix)

To use the messaging system in AssetCenter, you must add the following command lines:

- in Windows, in the **win.ini** file
- in UNIX, by creating a **mail.ini** file

```
[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
```

Location of these files: ► **AssetCenter - Installation guide**, chapter **.ini and .cfg files**.

## 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 the 'Messaging' tab of a user profile form. The 'Name' field contains 'Hartke' and 'First' is 'Richard'. The 'Title' is 'Building Maintenance' and 'Mr/Mrs' is 'Mr.'. The 'Department' is 'Administration'. The 'Account' field is populated with 'MAPI:MS Exchange Settings' and the 'Password' field contains a masked password '\*\*\*\*\*'.

- Populate the **Email** field of the **General** tab with the following syntax:

```
MAPI: [mailbox name]
```

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 populated with 'MAPI:gcolombo'. The 'ID #' is 'DEMO-M031', 'Cost center' is 'Common Line', '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 **Bcc** fields, if you wish) with the appropriate address, in the following manner:

```
MAPI: [mailbox name]
```

Or using a calculated string.

The screenshot shows a user detail form with the 'Messaging' tab selected. The 'Name' field is 'Email', the 'Context' is 'Employees (amEmplDept)', and the 'Type' is 'Messaging'. The 'To' field contains the value '[EMail]'. Other fields include 'Referenced object', 'Priority: Normal', 'Acknowledgment', 'Cc: MAPI:administrator', and 'Subject'. The 'Message' field is 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.

## win.ini file (Windows) or mail.ini (Unix)

Add the following lines to the [mail] section:

- in Windows, in the **win.ini** file
- in UNIX, by creating a **mail.ini** file

```
MAPI=1
MAPIX=1
```

Location of these files: ► **AssetCenter - Installation** guide, chapter **.ini and .cfg files**.

## 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 profile form with the 'Messaging' tab selected. The 'Account' field is populated with 'VIM:slistel/TALTEK' and the 'Password' field is masked with asterisks. Other visible fields include Name (Listel), First (Susan), Title (Switchboard Manager), and Department (Administration).

- **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]

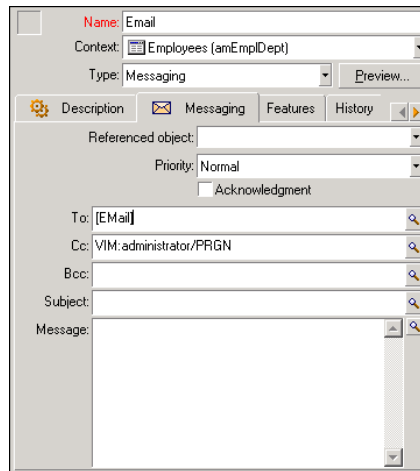
The screenshot shows the 'General' tab of a user profile form. The 'E-Mail' field is populated with 'VIM:slistel/TALTEK'. Other fields include Location (/San Mateo site/Building 01/Hall/001 - Reception/), Telephone (650) 572-9000, Fax (650) 572-9099, ID # (DEMO-M041), Cost center (Common Line), Hire date (21/12/1995), and Bar code (DEMO-U005).

- 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 **Bcc** fields, if you wish) according to the following syntax:

VIM: [name/domain]



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.d11**.
- 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.d11** 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.

### win.ini file (Windows) or mail.ini (Unix)

Add the following line to the **[mail]** section:

- in Windows, in the **win.ini** file
- in UNIX, by creating a **mail.ini** file

```
SMT=1
```

Location of these files: ► **AssetCenter - Installation** guide, chapter **.ini and .cfg files**.

## 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 shows a user profile configuration window. At the top, the user's name is 'Hartke, Richard' with 'Name: Hartke' and 'First: Richard'. Below this, there are fields for 'Title: Building Maintenance', 'Mr/Mrs: Mr.', and 'Department: Administration'. A tabbed interface is visible with tabs for 'General', 'Training', 'Portfolio', 'Costs', 'Projects', 'Profile', and 'Messag'. The 'Profile' tab is active, showing 'Profile:' (with a dropdown arrow), 'Login type: Named', 'Login:' (text field), and 'Password:' (text field). Below these is 'Default currency:' (with a dropdown arrow). The 'Identification' section contains 'Domain name:', 'User:', and 'Full name:' (all text fields). The 'Rights' section has three checkboxes: 'Administration rights', 'InfraTools Remote Control Manager', and '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 **Bcc** fields, if you wish) in the following manner:

```
AM: [login of recipient]
```

Or using a calculated string.


The screenshot shows a user detail form for 'Email'. The 'Context' is set to 'Employees (anEmplDept)'. The 'Type' is 'Messaging'. The 'Referenced object' is empty. The 'Priority' is 'Normal'. The 'Acknowledgment' checkbox is unchecked. The 'To' field is populated with '[EMail]'. The 'Cc' field is populated with 'AM:Brodin'. The 'Bcc' field is empty. The 'Subject' field is empty. The 'Message' field is 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 in 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 Customizing planner viewer screens

## CHAPTER

---

## Overview

A planner viewer page can be added to:

- The detail of a table using AssetCenter Database Administrator.
- To a wizard, using AssetCenter.

This is only true if you have at least the following information:

- A field storing a beginning (either the date or the date and time).
- A field storing an end (either the date or the date and time).

---

## Customizing how the planner viewer pages work

The options governing how all the planner viewer pages work are defined via the **Edit/Options** menu. Then go to **Display/Planner tools**.

 **Note:**

The fact that Saturday and Sunday appear as unworked days cannot be changed.

---

# Adding a planner viewer page to a table

## Adding a planner viewer page to a table

You can add (or delete) a planner viewer page to a detail screen using AssetCenter Database Administrator:

- 1 Launch AssetCenter Database Administrator.
- 2 Connect to the database you want to modify by selecting the **File/Open** menu and clicking **Open existing database**.
- 3 Select the table to modify.
- 4 Display the pages of the table (**View /Pages** menu).
- 5 Add a page by clicking **New**.
- 6 Populate the fields in the **General** tab.
- 7 Display the **Contents** tab.
- 8 Define a new field in the editing zone by typing a string in the following format:

```
PlannerViewer- [A] | StartDate= [B] | EndDate= [C] | StartBefore= [D] | EndBefore= [E] | StartAfter= [F] | EndAfter= [G] | ItemColor= [H] | GenerateColors= [I] | CompleteDates= [J] | LineLabel= [K] | ItemLabel= [L] | Progress= [M]
```

Syntax: ▶ [References: parameter syntax of the planner viewer pages \[page 251\]](#).

- 9 Add the field thus defined (▶ button).
- 10 Save the page by clicking **Create**.
- 11 Display the details of the table (**View/Details** menu).
- 12 Select the detail to which you want to add the previously created page.
- 13 Display the **Pages** tab.
- 14 Transfer the page previously created from the list of available pages to the list of detail pages (▶ button). Then place the page in the desired order among the other pages.
- 15 Save the detail thus modified by clicking **Modify**.
- 16 Save all these modifications in the database by selecting the **File/Update the database structure**.



Tip:

Do not populate the **Conversion file** field when it appears.



## Practical case

- 1 Launch AssetCenter Database Administrator.
- 2 Connect to the demonstration database by selecting the **File/Open** menu and clicking **Open existing database**.
- 3 Select the **Work orders** table (amWorkOrder).
- 4 Display the pages of the table (**View/Pages** menu).
- 5 Add a page by clicking **New**.
- 6 Populate the following fields in the **General** tab:



| Field    | Value          |
|----------|----------------|
| SQL name | pg.In_progress |
| Title    | In progress    |



- 7 Display the **Contents** tab.
- 8 Define a new field in the editing zone by typing the following string:

```
PlannerViewer-!WorkOrderId|StartDate=dtActualFixStart|EndDate=dtActualFixed|EndBefore=dtSchedFixed|StartBefore=dtSchedFixStart|ItemColor=!TechId|GenerateColors=true|CompleteDates=true|LineLabel=self|ItemLabel=Title
```

The parameter values thus defined are interpreted as follows:

| Parameter      | Value            | Effect                                                                                                                                                         |
|----------------|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PlannerViewer  | !WorkOrderId     | The events are the work orders themselves.                                                                                                                     |
| StartDate      | dtActualFixStart | The start of the events is defined by the <b>Actual start</b> field.                                                                                           |
| EndDate        | dtActualFixed    | The end of the events is defined by the <b>Actual end</b> date.                                                                                                |
| EndBefore      | dtSchedFixed     | If an event is not finished at the moment defined by the <b>Planned end</b> field, the corresponding circle ● and square ■ will appear in red.                 |
| StartBefore    | dtSchedFixStart  | If an event does not start at the moment defined by the <b>Planned start</b> field, the corresponding circle ● and square ■ will appear in red.                |
| GenerateColors | true             | A color is automatically assigned to the box <input type="checkbox"/> that represents the events according to the value of the <b>Technician</b> field.        |
| ItemColor      | !TechId          |                                                                                                                                                                |
| CompleteDates  | true             | If the <b>Actual start</b> or <b>Actual end</b> fields (dtActualFixStart and dtActualFixed) are empty, their values are replaced by the current time and date. |

| Parameter | Value | Effect                                                                                                                                                                                                                         |
|-----------|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| LineLabel | self  | The description string of the <b>Work orders</b> table (am-WorkOrder) is used to identify the events in the list to the left of the window  . |
| ItemLabel | Title | The value of the <b>Title</b> field is displayed in the box <input type="text"/> that represents the events in the list  .                    |

- 9 Add the field thus defined ( button).
- 10 Save the page by clicking **Create**.
- 11 Display the details of the table (**View/Details** menu).
- 12 Select the detail whose SQL name is **sysamWorkOrder**.
- 13 Display the **Pages** tab.
- 14 Transfer the **In\_progress** page from the list of available pages to the list of detail pages ( button). Then place the page in the desired order among the other pages.
- 15 Save the detail thus modified by clicking **Modify**.
- 16 Save all these modifications in the database by selecting the **File/Update the database structure**.



**Tip:**

Do not populate the **Conversion file** field when it appears.

- 17 Launch AssetCenter.
- 18 Connect to the demonstration database by selecting the **File/Connect to database** menu.



**Warning:**

If you are already connected to the demonstration database, you must disconnect then reconnect.

- 19 Display the list of work orders (**Portfolio/ Work orders** menu).
- 20 Create two work orders. Only populate the following fields:

| Tab | Name of the field | SQL name of the field | Value for work order 1 | Value for work order 2 |
|-----|-------------------|-----------------------|------------------------|------------------------|
|     | Work order #      | WONo                  | CP3                    | CP4                    |
|     | Notified on       | dtNotif               | 07/19/2002 11:00:00    | 07/19/2002 11:00:00    |

| Tab                     | Name of the field   | SQL name of the field | Value for work order 1                | Value for work order 2                    |
|-------------------------|---------------------|-----------------------|---------------------------------------|-------------------------------------------|
| General                 | Type                | seType                | Internal maintenance                  | Internal maintenance                      |
| General                 | Title               | Title                 | Analysis                              | Repair                                    |
| Tracking/<br>Technician | Technician          | Contact               | Select the technician of your choice. | Select another technician of your choice. |
| Tracking/<br>Scheduling | Planned start       | dtSchedFix-<br>Start  | 07/20/2002 14:00:00                   | 07/21/2002<br>14:00:00                    |
| Tracking/<br>Scheduling | Planned end<br>date | dtSchedFixed          | 07/21/2002 12:00:00                   | 07/24/2002<br>17:00:00                    |
| Tracking/<br>Scheduling | Actual start        | dtActualFix-<br>Start | 07/20/2002 08:00:00                   | 07/23/2002<br>08:00:00                    |
| Tracking/<br>Scheduling | Actual end          | dtActualFixed         | 07/22/2002 12:00:00                   |                                           |



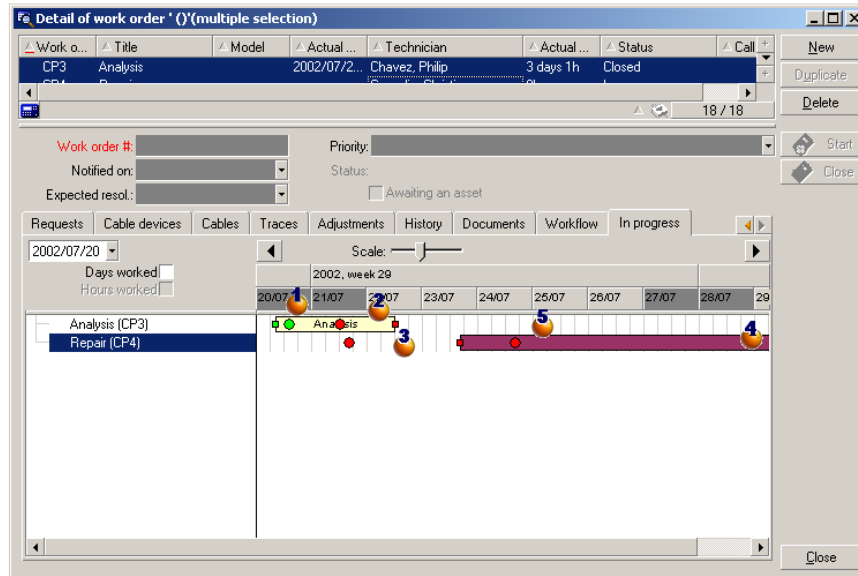
- 21 Select both **CP3** and **CP4**.
- 22 Select the **In progress** tab.
- 23 Use the **Scale** cursor as well as the  and  buttons to view the entire period covered by the **CP3** work order and beginning of the **CP4** work order:

Figure 14.1. Planner viewer - Example with work orders



- 1 Work order CP3: The **Actual start** comes before the **Planned start**, but the **Planned start** is declared by the parameter `StartBefore`. This is why the circle and square are green.
- 2 Work order CP3: The **Actual end** comes after the **Planned end**, but the **Planned end** is declared by the parameter `EndBefore`. That is why the circle and square are red.
- 3 Work order CP4: The **Actual start** comes after the **Planned start**, but the **Planned start** is declared by the parameter `StartBefore`. This is why the circle and square are red.
- 4 Work order CP4: The **Actual end** is not populated. Its value is replaced by the current date, because the parameter `CompleteDates` equals `true`. This is why the box is extended until the current date.
- 5 Work order CP3: The current day comes after the **Planned date**, but the **Planned date** is declared by the parameter `EndBefore`. This is why the circle is red (the square is not visible in this picture).

# Adding a planner viewer page to a wizard

## Adding a planner viewer page to a wizard

You can add a planner viewer page to a wizard using the AssetCenter wizard editor.

- 1 Launch AssetCenter.
- 2 Connect to the database you want to modify by selecting the **File/Connect to database** menu.
- 3 Display the list of actions (**Tools/ Actions/ Edit** menu).
- 4 Either select the wizard you want to modify or create a new one.



Note:

The Type field (seActionType) must have the value **Wizard**.

- 5 Display the **Wizard** tab.
- 6 Select an existing page or add a new one.
- 7 In this page, add a **PLANNER** node.
- 8 Populate the parameters of the **PLANNER** node, especially the **CONTENT** node whose value must reference the **amGeneratePlanningData(strTableSqlName, strProperties, strIds)** API of the AssetCenter API library:
  - **strTableSqlName** indicates the starting table to designate the events to display in the planner viewer page.  
Example: "**amWorkOrder**".
  - **strProperties** provides the list of parameters for the planner viewer page in the form:
 

```
"MainField= [A] | StartDate= [B] | EndDate= [C] | StartBefore= [D] | EndBefore= [E] | StartAfter= [F] | EndAfter= [G] | ItemColor= [H] | GenerateColors= [I] | CompleteDates= [J] | LineLabel= [K] | ItemLabel= [L] | WhereCond= [N] | OrderBy= [O] | Progress= [M] "
```

Syntax: ► [References: parameter syntax of the planner viewer pages \[page 251\]](#).
  - **strIds** identifies the list of records returned by the wizard whose events you want to display.



Note:

The planner viewer page displayed by the wizard does not allow you to modify the events represented therein.


## Practical case

- 1 Launch AssetCenter.
- 2 Connect to the demonstration database by selecting the **File/Connect to database** menu.
- 3 Display the list of actions (**Tools/ Actions/ Edit** menu).
- 4 Create a new wizard.
- 5 Populate the following fields:

| Name of the field | SQL name of the field | Value      |
|-------------------|-----------------------|------------|
| Name              | Name                  | CP         |
| Context           | ContextTable          | (No table) |
| Type              | seActionType          | Wizard     |

- 6 Display the **Wizard** tab.
- 7 Select **PAGE1**.
- 8 Add a **PLANNER** node to this page by right-clicking and selecting **Edit/PLANNER** from the shortcut menu.
- 9 Populate the **CONTENT** parameter of the **PLANNER** node with the following value:

```
RetVal = amGeneratePlanningData("amWorkOrder", "MainField=lWorkOrderId|S
tartDate=dtActualFixStart|EndDate=dtActualFixed|EndBefore=dtSchedFixed|
StartBefore=dtSchedFixStart|ItemColor=lTechId|GenerateColors=true|Compl
eteDates=true|LineLabel=self|ItemLabel=Title|WhereCond=seType\=0|OrderB
y=Technician.Name, WOno", "")
```

- 10 Save the wizard by clicking **Create**.
- 11 Test the wizard by clicking the icon .

# References: parameter syntax of the planner viewer pages

This section explains how to set the parameters of a planner viewer page.

A planner viewer page can be added to:

- The detail of a table (with AssetCenter Database Administrator).  
Syntax and the list of usable parameters: ► [Adding a planner viewer page to a table](#) [page 244].
- A page in a wizard (with AssetCenter).  
Syntax and the list of usable parameters: ► [Adding a planner viewer page to a table](#) [page 244].

## PlannerViewer- [A] (table) or MainField= [A] (wizard)

|               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Use           | Specify which events to display.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Values        | <ul style="list-style-type: none"> <li>■ If the events are part of the table:           <div style="border: 1px solid black; padding: 2px; margin: 5px 0;"><code>&lt;Primary key of the table&gt;</code></div>           The table providing context to the other parameters is either:           <ul style="list-style-type: none"> <li>■ The table to which you add the planner viewer page.</li> <li>■ The <code>strTableSqlName</code> table of the API <code>amGeneratePlanningData(strTableSqlName, strProperties, strIds)</code> that defines the wizard.</li> </ul> </li> <li>■ If the events are part of a linked table:           <div style="border: 1px solid black; padding: 2px; margin: 5px 0;"><code>&lt;SQL name of link1&gt;.&lt;SQL name of link2&gt;...&lt;SQL name of linkN&gt;</code></div>           The table providing context to the other parameters is then the target table of <code>LinkN</code>.<br/>Exception: for the parameter <code>LineLabel= [K]</code>, it's the source table of <code>linkN</code> that provides context.         </li> </ul> |
| Example 1     | If you add a planner viewer page to the <b>Work orders</b> table ( <code>amWorkOrder</code> ) and you want to view the work orders themselves, [A] equals <b>IWorkOrderId</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Example 2     | If you add a planner viewer page to the <b>Work orders</b> table ( <code>amWorkOrder</code> ) and you want to view the internal technician absences, [A] equals <b>Technician.Absences</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Default value | None                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

|           |     |
|-----------|-----|
| Mandatory | Yes |
|-----------|-----|

### StartDate= [B]

|               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Use           | Defines the start of events.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Values        | <p>A <b>Date</b> or <b>Date and time</b> type field specified as follows:</p> <ul style="list-style-type: none"> <li>■ &lt;SQL name of a direct field in the table providing context&gt;</li> <li>■ &lt;SQL name of link1 in the table providing context&gt;.&lt;SQL name of link2&gt;...&lt;SQL name of linkN&gt;.&lt;SQL name of the field&gt;</li> </ul> <p>Note:</p> <p>The table providing context is defined by the parameter <code>PlannerViewer- [A]</code> (table) or <code>MainField= [A]</code> (wizard)</p> |
| Example       | For the <b>Absences</b> table (amAbsence), champ <b>Out date</b> , [B] equals <b>dtOut</b> .                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Default value | None                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Mandatory     | Yes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |

### EndDate= [C]

|               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Use           | Defines the end of events.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| Values        | <p>A <b>Date</b> or <b>Date and time</b> type field specified as follows:</p> <ul style="list-style-type: none"> <li>■ &lt;SQL name of a direct field in the table providing context&gt;</li> <li>■ &lt;SQL name of link1 in the table providing context&gt;.&lt;SQL name of link2&gt;...&lt;SQL name of linkN&gt;.&lt;SQL name of the field&gt;</li> </ul> <p>Note:</p> <p>The table providing context is defined by the parameter <code>PlannerViewer- [A]</code> (table) or <code>MainField= [A]</code> (wizard)</p> |
| Example       | For the <b>Work orders</b> table (amWorkOrder), <b>Planned end</b> field, [C] equals <b>dtSchedFixed</b> .                                                                                                                                                                                                                                                                                                                                                                                                              |
| Default value | None                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Mandatory     | Yes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |

### StartBefore= [D]

|             |                                                                                                                    |
|-------------|--------------------------------------------------------------------------------------------------------------------|
| Use         | Defines the deadline that the start of the event must not go over.                                                 |
| Precautions | You must not use the <code>StartAfter</code> parameter at the same time as the <code>StartBefore</code> parameter. |



|        |                                                                                                                                                                                                                                                                                                                                                             |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Values | <p>A <b>Date</b> or <b>Date and time</b> type field specified as follows:</p> <ul style="list-style-type: none"> <li>■ &lt;SQL name of a direct field in the table providing context&gt;</li> <li>■ &lt;SQL name of link1 in the table providing context&gt;.&lt;SQL name of link2&gt;...&lt;SQL name of linkN&gt;.&lt;SQL name of the field&gt;</li> </ul> |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

**Note:**

The table providing context is defined by the parameter `PlannerViewer- [A]` (table) or `MainField= [A]` (wizard)

|               |                                                                                                                                            |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| Example       | For the <b>Work orders</b> table ( <code>amWorkorder</code> ), <b>Expected resol.</b> field, <code>[D]</code> equals <b>dtResolLimit</b> . |
| Default value | None                                                                                                                                       |
| Mandatory     | No                                                                                                                                         |

### **EndBefore= [E]**

|             |                                                                                                                |
|-------------|----------------------------------------------------------------------------------------------------------------|
| Use         | Defines the deadline that the end of the event must be go over.                                                |
| Precautions | You must not use the <code>EndAfter</code> parameter at the same time as the <code>EndBefore</code> parameter. |

|        |                                                                                                                                                                                                                                                                                                                                                             |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Values | <p>A <b>Date</b> or <b>Date and time</b> type field specified as follows:</p> <ul style="list-style-type: none"> <li>■ &lt;SQL name of a direct field in the table providing context&gt;</li> <li>■ &lt;SQL name of link1 in the table providing context&gt;.&lt;SQL name of link2&gt;...&lt;SQL name of linkN&gt;.&lt;SQL name of the field&gt;</li> </ul> |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

**Note:**

The table providing context is defined by the parameter `PlannerViewer- [A]` (table) or `MainField= [A]` (wizard)

|               |                                                                                                                                        |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------|
| Example       | For the <b>Work orders</b> table ( <code>amWorkorder</code> ), <b>Planned end</b> field, <code>[E]</code> equals <b>dtSchedFixed</b> . |
| Default value | None                                                                                                                                   |
| Mandatory     | No                                                                                                                                     |

### **StartAfter= [F]**

|             |                                                                                                                    |
|-------------|--------------------------------------------------------------------------------------------------------------------|
| Use         | Defines the deadline after which the event must start.                                                             |
| Precautions | You must not use the <code>StartBefore</code> parameter at the same time as the <code>StartAfter</code> parameter. |

|        |                                                                                                                                                                                                                                                                                                                                                             |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Values | <p>A <b>Date</b> or <b>Date and time</b> type field specified as follows:</p> <ul style="list-style-type: none"> <li>■ &lt;SQL name of a direct field in the table providing context&gt;</li> <li>■ &lt;SQL name of link1 in the table providing context&gt;.&lt;SQL name of link2&gt;...&lt;SQL name of linkN&gt;.&lt;SQL name of the field&gt;</li> </ul> |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

**Note:**

The table providing context is defined by the parameter `PlanerViewer- [A]` (table) or `MainField= [A]` (wizard)

|               |                                                                                                                                   |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------|
| Example       | For the <b>Work orders</b> table ( <code>amWorkorder</code> ), <b>Notified on</b> field, <code>[F]</code> equals <b>dtNotif</b> . |
| Default value | None                                                                                                                              |
| Mandatory     | No                                                                                                                                |

### EndAfter= [G]

|             |                                                                                                         |
|-------------|---------------------------------------------------------------------------------------------------------|
| Use         | Defines the deadline after which the event must end.                                                    |
| Precautions | You must not use the <code>EndBefore</code> parameter at the same time <code>EndAfter</code> parameter. |

|        |                                                                                                                                                                                                                                                                                                                                                             |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Values | <p>A <b>Date</b> or <b>Date and time</b> type field specified as follows:</p> <ul style="list-style-type: none"> <li>■ &lt;SQL name of a direct field in the table providing context&gt;</li> <li>■ &lt;SQL name of link1 in the table providing context&gt;.&lt;SQL name of link2&gt;...&lt;SQL name of linkN&gt;.&lt;SQL name of the field&gt;</li> </ul> |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

**Note:**

The table providing context is defined by the parameter `PlanerViewer- [A]` (table) or `MainField= [A]` (wizard)

|               |      |
|---------------|------|
| Default value | None |
| Mandatory     | No   |

### GenerateColors= [I]

|     |                                                                                                                   |
|-----|-------------------------------------------------------------------------------------------------------------------|
| Use | Determines if colors must be used automatically to color the box <input type="checkbox"/> that represents events. |
|-----|-------------------------------------------------------------------------------------------------------------------|

|             |                                                                                                         |
|-------------|---------------------------------------------------------------------------------------------------------|
| Precautions | The color that is the reference for the generation of the other colors is <code>ItemColor= [H]</code> . |
|-------------|---------------------------------------------------------------------------------------------------------|

|        |                                                                                                                                                                                                                                   |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Values | <ul style="list-style-type: none"> <li>■ <b>true</b>: the colors will be automatically modified each time the reference color changes.</li> <li>■ <b>false</b>: the colors are defined in RGB format by a given field.</li> </ul> |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|               |      |
|---------------|------|
| Default value | true |
|---------------|------|

|           |    |
|-----------|----|
| Mandatory | No |
|-----------|----|

### ItemColor= [H]

|     |                                                                                                         |
|-----|---------------------------------------------------------------------------------------------------------|
| Use | Determines the reference color to use to color the box <input type="checkbox"/> that represents events. |
|-----|---------------------------------------------------------------------------------------------------------|

|             |                                                                                                |
|-------------|------------------------------------------------------------------------------------------------|
| Precautions | The value can take [H] depending on the value that you gave the GenerateColors= [I] parameter. |
|-------------|------------------------------------------------------------------------------------------------|

|                                           |                                                                                                                                                            |
|-------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Values if GenerateColors= [I] equals true | [H] indicates which field or link conditions the color of the boxes. Each time the value if this field or link changes, the color of the box changes, too. |
|-------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|

|                                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Values if GenerateColors= [I] equals true | [H] can take one of the following forms: <ul style="list-style-type: none"> <li>■ &lt;SQL name of a direct field in the table providing context&gt;</li> <li>■ &lt;SQL name of link1 in the table providing context&gt;.&lt;SQL name of link2&gt;...&lt;SQL name of linkN&gt;</li> <li>■ &lt;SQL name of link1 in the table providing context&gt;.&lt;SQL name of link2&gt;...&lt;SQL name of linkN&gt;.&lt;SQL name of the field&gt;</li> </ul> |
|-------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

**Note:**

The table providing context is defined by the parameter `PlanerViewer- [A]` (table) or `MainField= [A]` (wizard)

|                                            |                                                                                                |
|--------------------------------------------|------------------------------------------------------------------------------------------------|
| Values if GenerateColors= [I] equals false | [H] indicates which type of <b>Integer (32 bit)</b> stores the color of the box in RGB format. |
|--------------------------------------------|------------------------------------------------------------------------------------------------|

|                                            |                                                                                                                                                                                                                                                                                                                        |
|--------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Values if GenerateColors= [I] equals false | [H] can take one of the following forms: <ul style="list-style-type: none"> <li>■ &lt;SQL name of a direct field in the table providing context&gt;</li> <li>■ &lt;SQL name of link1 in the table providing context&gt;.&lt;SQL name of link2&gt;...&lt;SQL name of linkN&gt;.&lt;SQL name of the field&gt;</li> </ul> |
|--------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

**Note:**

The table providing context is defined by the parameter `PlanerViewer- [A]` (table) or `MainField= [A]` (wizard)

**Tip:**

To learn how to determine the RGB value of a color, refer to this guide's chapter **Record lists**, section **Main list layout/Using the 'Configure list' shortcut menu item**.

|               |      |
|---------------|------|
| Default value | None |
|---------------|------|

|           |     |
|-----------|-----|
| Mandatory | Yes |
|-----------|-----|

### CompleteDates= [J]

|     |                                                                                                                                        |
|-----|----------------------------------------------------------------------------------------------------------------------------------------|
| Use | Defines what must be done when at least one of the fields defined by <code>StartDate</code> and <code>EndDate</code> is not populated. |
|-----|----------------------------------------------------------------------------------------------------------------------------------------|


|                        |                                   |
|------------------------|-----------------------------------|
| Example of utilization | Managing work orders in progress. |
|------------------------|-----------------------------------|

|        |                                                                                                                                                                                                                         |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Values | <ul style="list-style-type: none"> <li>■ <b>true</b>: the dates not populated by these fields are replaced by the current date.</li> <li>■ <b>false</b>: the dates not populated by these fields stay empty.</li> </ul> |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|               |       |
|---------------|-------|
| Default value | false |
|---------------|-------|

|           |    |
|-----------|----|
| Mandatory | No |
|-----------|----|

### ItemLabel= [L]

|     |                                                                                                                                                                      |
|-----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Use | Identifies the title to use in the boxes that represent the events in the list  . |
|-----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Values | <ul style="list-style-type: none"> <li>■ <b>self</b>: the title takes the value of the table's description string.</li> <li>■ <b>&lt;SQL name of a direct field in the table providing context&gt;</b>: the name takes the value of the field.</li> <li>■ <b>&lt;SQL name of link1 in the table providing context&gt;.&lt;SQL name of link2&gt;...&lt;SQL name of linkN&gt;</b>: the name takes the value of the field.</li> <li>■ <b>&lt;SQL name of link1 in the table providing context&gt;.&lt;SQL name of link2&gt;...&lt;SQL name of linkN&gt;.&lt;SQL name of the field&gt;</b>: the name takes the value of the field.</li> </ul> |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|


Note:

The table providing context is defined by the parameter `Plan-  
nerViewer- [A]` (table) or `MainField= [A]` (wizard)

|               |      |
|---------------|------|
| Default value | None |
|---------------|------|

|           |    |
|-----------|----|
| Mandatory | No |
|-----------|----|

### LineLabel= [K]

|     |                                                                                                                                                                                          |
|-----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Use | Identifies the title to use to identify the events in the list of events to the left of the window  . |
|-----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

- Values
- **self**: the title takes the value of the table's description string.
  - **<SQL name of a direct field in the table to which we add a planner view page>**: the label takes the value of the field.
  - **<SQL name of link1 in the table to which we add a planner viewer page>.<SQL name of link2>...<SQL name of linkN>**: the label takes the value of the description string or the LinkN table.
  - **<SQL name of link1 in the table to which we add a planner viewer page>.<SQL name of link2>...<SQL name of linkN>.<SQL name of field>**: the label takes the value of the field.

**Note:**

The table providing context is defined by the parameter `Plan-nerViewer- [A]` (table) or `MainField= [A]` (wizard)

|               |                                                                                                                                                                                                                  |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Default value | <ul style="list-style-type: none"> <li>■ If [A] is the table's primary key: description string of the table.</li> <li>■ If [A] is a succession of links: description string of the last link's table.</li> </ul> |
| Mandatory     | No                                                                                                                                                                                                               |

### Progress= [M]

- Usage
- Identifies a field that stores the progression percentage of events.  
The user type of this field must be **Percentage**.

**Note:**

You can create a new field with AssetCenter Database Administrator.

**Note:**

You cannot use a calculated field.

|               |                                                                                                                                                                                                                                   |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Values        | <ul style="list-style-type: none"> <li>■ <b>&lt;SQL of a direct field in the table&gt;</b></li> <li>■ <b>&lt;SQL name of link1&gt;.&lt;SQL name of link2&gt;...&lt;SQL name of linkN&gt;.&lt;SQL name of field&gt;</b></li> </ul> |
| Default value | None                                                                                                                                                                                                                              |
| Mandatory     | No                                                                                                                                                                                                                                |

### WhereCond= [N]

- Use
- Filters the events to display.

|               |                                                                                                                                                                                 |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Values        | Condition expressed in AQL.<br>The AQL condition is not surrounded by any delimiters.<br>On the other hand, the = and   characters must be preceded by the \ escape character.  |
| Example       | For the <b>Work orders</b> table (amWorkorder), if you only want to keep the records whose <b>Type</b> field equals <b>Internal maintenance</b> , [N] equals <b>seType\=0</b> . |
| Default value | None                                                                                                                                                                            |
| Mandatory     | No                                                                                                                                                                              |

## OrderBy= [O]

|               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Use           | Sorts the events to display.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Values        | List of fields or links separated by the , character.<br>The fields are represented by using one of the following syntaxes: <ul style="list-style-type: none"> <li>■ &lt;SQL name of a direct field in the table providing context&gt;</li> <li>■ &lt;SQL name of link1 in the table providing context&gt;.&lt;SQL name of link2&gt;...&lt;SQL name of linkN&gt;.&lt;SQL name of the field&gt;</li> </ul> The links are represented by using one of the following syntaxes: <ul style="list-style-type: none"> <li>■ &lt;SQL name of link1 in the table providing context&gt;: This returns the value of the primary key in the linked record.</li> <li>■ &lt;SQL name of link1 in the table providing context&gt;.&lt;SQL name of link2&gt;...&lt;SQL name of linkN&gt;.&lt;SQL name of the field&gt;</li> <li>■ &lt;SQL name of link1 in the table providing context&gt;.&lt;SQL name of link2&gt;...&lt;SQL name of linkN&gt;.self</li> </ul> |
| Example       | If you add a planner viewer page to the <b>Work orders</b> table (amWorkOrder), and you want to sort the work orders by technician, then the work order number [O] equals <b>Technician.Name, WONo</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Default value | None                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Mandatory     | No                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |



# 15

## 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 **am**.

## 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".



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 in 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 in AssetCenter (which can be found in the **andb.ini** file).

Location of this file: ► **AssetCenter - Installation** guide, chapter **.ini and .cfg files**.

- 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 in 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 in 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 in 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 in 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 in AssetCenter.

### <ViewName>

This argument contains the SQL name of the view as it is defined in AssetCenter.

### Example

The following command opens the view with the SQL name **CurrentAssets**:

```
Execute:OpenView(CurrentAssets)
```

---

## 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 in 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

The following command puts the table of products (SQL name: amPortfolio) in tree-view:

```
Execute:amPortfolio.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 **New** button.
- "Duplicate": Duplicates the selected record in the table "<Table>". The command corresponds to clicking the **Duplicate** button.
- "Delete": Destroys the record selected in the table "<Table>". The command corresponds to clicking the **Delete** button.
- "Modify": Confirms the modifications made to the selected record in the table "<Table>". The command corresponds to clicking the **Modify** button.

- "Create": Confirms the creation of a new record in the table "<Table>". The command corresponds to clicking the **Create** button.
- "CreateContinue": Combines a creation and a duplication. The command corresponds to clicking the **CreateContinue** button.
- "Cancel": Cancels the creation of a new record or the modifications made to the selected record. The command corresponds to clicking the **Cancel** button.
- "Close": Closes the previously opened table "<Table>". The command corresponds to clicking the **Close** 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 8/28/02 at 15:00:00:

```
Execute:amAsset.SetFilter(dtLastModif<[02/08/28 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-Object.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 **Comment** 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.Highlight()
```

## <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 **Asset tag** field (AssetTag) of the **Assets** table (amAsset):

```
Execute:amAsset:AssetTag.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 and 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 "08/28/02" to the **Acceptance date** field (SQL name: dAccept) in the Assets table (SQL name: amAsset), for the selected record:

```
Execute:amAsset:dAccept.SetValue(02/08/28)
```

## <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:Model.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 in 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 location, called "Headquarters" located in San Diego. Here are the actions carried out:

- 1 Open the Locations table
- 2 Enter the name of the location
- 3 Enter the city of the location
- 4 Enter the postal code of the location.
- 5 Create the location

## Introduction

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: **am**
- Topic: **AssetCenter**

We now just have to execute the command that opens the table of locations:

- Command: **OpenTable()**
- Parameter: The SQL name of the table; in this case **amLocation**

The command is written as follows:

```
OpenTable(amLocation)
```

AssetCenter opens the table of locations. 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. **amLocation**
- Parameter: The date entry mode; in this case **New**

The command is written as follows:

```
amLocation.SetRecordMode(New)
```

## Entering data

Next, we need to give AssetCenter the appropriate commands for the fields that interest us:

- **Name** field (name). 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. **amLocation**
    - \* Argument **<Object>**: The SQL name of the field, i.e. **Name**
    - \* Argument **<Value>**: The value of the field, i.e. **Headquarters**

```
amLocation:TextLabel.SetValue(Headquarters)
```

- **City** field (City). 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. **amLocation**
    - \* Argument **<Object>**: The SQL name of the field, i.e. **City**
    - \* Argument **<Value>**: The value of the field, i.e. **San Diego**

```
amLocation:seDataType.SetValue(San Diego)
```

- **Postal code** field (ZIP). 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. **amLocation**
- \* Argument <Object>: The SQL name of the field, i.e. **ZIP**
- \* Argument <Value>: The value of the field, i.e. **CA 92130**

```
amLocation.Unit.SetValue (CA 92130)
```

## Creating the location

In order to create the location, we just need to create the record in the table of locations:

- ◆ Command: <Table>.SetRecordMode(<Mode>)
  - Argument <Table>: The SQL name of the table, i.e. **amLocation**
  - Argument <Mode>: The mode of creation of a record, i.e. **Create**

```
amLocation.SetRecordMode (Create)
```

## Second scenario - DDE calls from Microsoft Excel

 Note:

This scenario has been tested with Microsoft Excel 2002.

This example is based on an Excel worksheet **TestDDE.xls**, which can be found in the **Samples\DDE\Excel** 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 of the file **TestDDE.xls**. It contains a table with three columns named **Name**, **City**, and **ZIP** respectively, and a button **Create location** to which a macro called **Create location** has been assigned.

You just need to enter the appropriate values in the **Name** (cell B6 in this example), **City** (cell C6), **Zip** (cell D6) fields, and click **Create the location** in order for Excel to give issue the orders to AssetCenter to perform the following actions:

- 1 Open the table of locations
- 2 Enter the name of the location
- 3 Enter the city of the location.
- 4 Enter the postal code of the location
- 5 Create the location

## Source code of the macro

For reference, here is the source code of the macro that executes the DDE calls. For further information about DDE calls in Excel or Word and the syntax of VBA language, please refer to the respective online documentation.

```
Sub CreateLocation()

Dim Name As String
Dim City As String
Dim ZIP As String

Name = Worksheets("Data_entry").Range("B6")
City = Worksheets("Data_entry").Range("C6")
ZIP = Worksheets("Data_entry").Range("D6")

Contexte = Application.DDEInitiate(app:="am", topic:="AssetCenter")
Application.DDEExecute Contexte, "OpenTable(amLocation)"
Application.DDEExecute Contexte, "amLocation.SetRecordMode(New)"
Application.DDEExecute Contexte, "amLocation.Name.SetValue(" + Name + ")"
Application.DDEExecute Contexte, "amLocation.City.SetValue(" + City + ")"
Application.DDEExecute Contexte, "amLocation.ZIP.SetValue(" + ZIP + ")"
Application.DDEExecute Contexte, "amLocation.SetRecordMode(Create)"
Application.DDETerminate Contexte

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 in Visual Basic.

For this example, execute **DDETestCenter.exe**, which can be found by following this path: **Samples\DDE\Program**. This program allows you to execute **Execute** and **Request** type DDE commands.



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:

- Source code of the program
- Before starting
- Executing 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\Visual Basic**.

### 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 (amLocation)
```

## Example n 2:

The following **Execute** commands open the table of budgets, create a new record and populate the **Name** field (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 (amFeature) opened beforehand:

```
amFeature.ListAllFields ()
```



# 16 | 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.
- **Regular test, 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 **Tailoring** guide, chapter **Customizing a client workstation**, paragraph, **General AssetCenter interface options**.

## Setting list parameters at the AssetCenter client level

The **FetchingArraySize** 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 **FetchingArraySize** parameter.

This parameter is defined at the level of each AssetCenter client in the **amdb.ini** file and for each AssetCenter connection.

Location of this file: ► **AssetCenter - Installation** guide, chapter **.ini and .cfg files**.

If the **FetchingArraySize** 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 **FetchingArraySize** 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:

```
FetchingArraySize = Do not load more than + 1
```



Note:

We tested this scenario with a WAN with a 250 ms latency. By optimizing this parameter, we gained **1.5 s** on displaying a list of 200 records.

Example of application:

- **Do not load more than** equals **200**.
- Set the value of **FetchingArraySize** to **210**.
- Thus, AssetCenter fetches just once.

## To modify `amdb.ini`

- 1 Edit the `amdb.ini` file.  
Location of this file: ► **AssetCenter - Installation** guide, chapter **.ini and .cfg files**.
- 2 Search for the selection [`<AssetCenter connection name to optimize>`]
- 3 Find out if a line already exists in the section that starts with `FetchingArraySize=`.  
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:  
`FetchingArraySize=<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.

Location of this file: ► **AssetCenter - Installation** guide, chapter **.ini and .cfg files**.





# 17 | 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 out-of-the-box 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: **Draw Tools** and **AssetCenter**.

# Installation

For information on how to install the AssetCenter AutoCAD module, refer to the AssetCenter **Installation** guide, chapter **Configuration in Windows**, section **Integrating with AutoCAD**.

## Configuration

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 **AssetCenter** connector and then 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 then 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 **ODBC** for the AssetCenter database connection type compatible with AutoCAD then click **Next**.

- 4 Select your AssetCenter database that is compatible with AutoCAD (**Connection** field) and populate the **Login** and **Password** fields before clicking **Next**.
- 5 Click **Finish**.

---

## Using AutoCAD integration

### Graphic interface

The AutoCAD module installed by AssetCenter adds two menus in AutoCAD: **Draw Tools** 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 Save the drawing before continuing (**File/ Save as** menu).
- 2 Use the **AssetCenter/Property Portfolio** menu.
- 3 Select **Property Navigator** from **Drawing Management**
- 4 Select your AssetCenter database from the **Profile** drop-down list.



Tip:

If AutoCAD displays error 37000, this can be ignored.

- 
- 5 Choose your location (building, floor, room, etc.) and click **Attach**.
- 



Tip:

The list of locations is hierarchical, even if this is not directly visible:  
Double-click a location to unfold a level.

You can only select one **floor** level to click **Attach**.

- 
- 6 AutoCAD tells you when it is connected to the database, and you can start 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.

---







# Index

## A

- acadi.scn, 297 , 296
- Access (See Access control)
- Access control, 163-184
  - (See Also Passwords)
  - Constraints, 163
  - Editing, 166
  - Functioning, 164
  - Usefulness of, 164
  - User connections, 174
  - Users, 172
- Access restrictions, 170
  - Definition, 166
  - Links - default value, 172 , 172
  - Slow networks - tuning, 292
- Actions
  - Automatic triggering (See AssetCenter Server)
- adbc.dll, 111
- addcpu.scn (scenario), 205 , 203
- additional netmem (Sybase Adaptive Server), 49
- AddLink (DDE), 274
- Add NT users to the database (module), 204 , 204
- Add the computers listed in the NT domain to the database (module), 202 , 202
- Admin (login), 181 , 13
- Administration rights (option), 173
- alias (keyword), 75
- AM (messaging), 238
- amdb.ini, 293 , 290
- amimpl.exe, 161
- amsrvf.ini, 224
- Anonymous (access), 183
- API, 21
- APP\_CTL\_HEAP\_SZ (UDB DB2), 50
- AppBuild (property), 83
- AppInfo (property), 83
- AppLanguage (property), 83
- APPLHEAPSZ (UDB DB2), 50
- AppVersion (property), 83
- AssetCenter Database Administrator (See Databases)
  - Documents - automatic loading, 14
  - Graphic interface, 13-19
  - Interface, 14
  - Languages, 15
  - License, 59
  - Object types, 15
  - Options, 19
  - Precautions, 13
  - Presentation, 13

- Searches, 19
- Tables, 15
- AssetCenter Databases (ODBC connection), 113
- AssetCenter Server, 187-227
  - Administration - Web, 223
    - Activation, 224
    - AssetCenter Server service, 225
    - Driving, 225
    - Start-up, 224
  - Automatic disconnection, 227
  - Constraints, 188
  - Databases
    - Connection, 190
    - Disconnection, 190
  - Execution
    - Recommendations, 188
    - Unix, 191
    - Windows, 190
  - Exiting - consequences, 195
  - Functioning, 187
  - Icons, 196
  - Installation, 199
  - Launching
    - Automatic launching, 189
    - DOS, 190
    - Manual launching, 189
  - Main screen, 196
  - Menus, 196
  - Messaging, 195
  - Modules - configuration, 199
  - Monitoring - time zones, 199
  - Options, 197
  - Service - executing as a, 225
  - Starting as a service, 223
  - Triggering immediately, 222
  - Web - limitations, 226
  - Windows integrated security (See Windows integrated security)
- AutoCAD connector, 296
- AutoCAD - integration, 295-299
  - Installation, 296
  - Presentation, 295
  - Utilization, 297
- auto create statistics (Microsoft SQL Server), 47
- auto update statistics (Microsoft SQL Server), 47

- AvailableScript (object), 96

## B

- Bandwidth (See Slow networks - tuning)
- Base (object), 92, 90, 87, 85
- BaseType (property), 101
- Basic script (input type), 28, 28
- Basic scripts - import, 150
- Binary field (type), 27
- Boolean (input type), 28
- Boolean (Yes/No) (input type), 28

## C

- cache (Oracle), 47
- cache (Sybase Adaptive Server), 50
- Caches, 292
- CalcField (object), 101
- CalcFieldDesc (instance), 97
- CalcMode (property), 93
- Calculated fields
  - Import, 143
- Calculate rents and loans (module), 206
- Calculate stipulated loss values (module), 208
- CaptionDetail (property), 99
- CaptionList (property), 99
- Cardinality (property), 90
- Casual (access), 175
- Character set (parameter), 45
- Class (object), 95
- CloseTable (DDE), 267
- Collection (instance), 71
- CollectionCount (function), 79
- CollectionCreate (function), 78
- CollectionCurrentIndex (function), 79
- CollectionIsFirst (function), 78
- CollectionIsLast (function), 79
- CollectionName (function), 78
- CollectionNext (function), 78
- Columns
  - Import - separators, 138
  - Slow networks - tuning, 289
- Comment (type of information memorized by a link), 34, 34
- Comments, 73

- CompleteDates (parameter), 256
  - ComputeString (property), 84
  - Concatenation, 148
  - config (folder), 17
  - Configure list (menu), 288
  - Connect (DDE), 262
  - Connected (property), 83
  - Connection (property), 83
  - Connections
    - (See Also Connection slots)
    - (See Also Declaration)
    - (See Also LDAP)
    - Declaration, 53
    - Managing, 174
    - Windows XP, 54
  - Connection slots
    - Accidental termination, 176
    - Destruction, 176
      - Manual destruction, 176
    - Functioning, 175
    - Inactive users, 177
    - Updating - delays, 176
  - Connect-It - AutoCAD integration, 296
  - Copy (link type), 33
  - Count (property), 84
  - Create (property), 83
  - Create assets, consumables, etc. corresponding to items received (module), 209
  - CreationHistorized (property), 96
  - Crystal Reports, 114 (See Crystal Reports)
  - CurrentIndex (property), 84
  - Custom itemized list (input type), 28 , 28
- D**
- DATA (Microsoft SQL Server configuration), 47
  - DATA (Sybase Adaptive Server), 49
  - database.txt, 67 , 26 , 21
    - Structure, 23
  - Database (instance), 83 , 71
  - Database administrators, 173
  - database device (Sybase Adaptive Server), 49
  - Database Owner (Microsoft SQL Server), 47
  - Database reconnection delay (option), 197
  - Databases
    - (See Also Access control)
    - (See Also Connection)
    - (See Also Fields)
    - (See Also Import)
    - (See Also Indexes)
    - (See Also Links)
    - (See Also ODBC)
    - (See Also Tables)
    - Access types, 174
      - Casual, 175
      - Floating, 174
      - Named, 174
    - Creation, 41-64
      - DB2 UDB, 50
      - DBMS, 44
      - Empty databases, 43
      - MSDE, 52
      - New database, 17
      - Preliminary verifications, 46
      - Prerequisites, 41
      - Sybase Adaptive Server, 49
    - Data
      - Export, 63
      - Import, 64
    - Definition, 22
    - Deletion, 64
    - Description, 67
    - Description - file, 67-103
      - Global variables
      - Overview
      - Parameters
    - Description - templates, 71-82
      - Browsed components - count of
      - Browsing components
      - Comments
      - else
      - End-of-paragraph mark - deleting
      - Filtering
      - Fixed text
      - Functions
      - Global variables
      - if
      - Including another template, 73
      - Properties - value
      - Sort

- Diagnostics, 105
  - Microsoft SQL Server, 47
  - Opening, 17
  - Repair, 105-107
  - Structure
    - AssetCenter Database Administrator, 55
    - Export, 63
    - Import, 64
    - Updating, 109
    - Viewing, 21
  - UDB DB2 - clients, 51
  - Database servers, 44
  - Data confidentiality, 165
  - Data Only Locking (Sybase Adaptive Server), 50
  - Date (global variable of an AssetCenter database description instance), 102
  - Date (input type), 28, 28
  - Date (type), 27
  - Date and time (input type), 28, 28
  - Date and time (type), 27
  - db\_block\_buffers (parameter), 45
  - db\_block\_size (parameter), 45
  - DB2 UDB, 50, 50
  - dbase.tpl, 23
  - Dbb.Ext (global variable of an AssetCenter database description instance), 103
  - Dbb.Fullname (global variable of an AssetCenter database description instance), 103
  - Dbb.Name (global variable of an AssetCenter database description instance), 103
  - Dbb.Path (global variable of an AssetCenter database description instance), 103
  - Dbb.Shortname (global variable of an AssetCenter database description instance), 103
  - DbbVersion (property), 83
  - dbdict.tpl, 68, 22
  - DBMS (See Databases)
    - Changing, 62
  - DBMS login, 53
  - DDE, 259-286
    - Calls, 259
    - Calls - examples
      - Excel, 282
      - Internal calls, 280
      - Introduction, 280
      - Visual Basic, 284
  - Commands, 260
    - Introduction, 261
  - Commands associated with a table, 266
    - CloseTable, 267
    - GetRecordCount, 268
    - GetSelectionId, 273
    - ListAllFields, 270
    - ListAllLinks, 271
    - OpenTable, 267
    - SetFilter, 272
    - SetRecordMode, 269
    - SetSelection, 272
    - SetViewMode, 268
  - Commands associated with a table and a field or a link, 273
    - AddLink, 274
    - GetValue, 275
    - Highlight, 275
    - RemoveLink, 276
    - SetFocus, 277
    - SetValue, 277
    - SetValueWhere, 278
    - Show, 279
  - Context, 261
  - Global commands, 261
    - Connect, 262
    - Disconnect, 263
    - ExecuteAction, 263
    - ListAllTables, 264
    - ListOpenTables, 265
    - OpenView, 266
  - Prerequisites, 284
  - Services, 260
  - Topics, 260
- DDE commands (See DDE)
  - DDE command servers, 259
  - DDE services, 260
  - DDE TestCenter.exe, 284
  - DDE topics, 260
  - Deadlines - monitoring (See AssetCenter Server)
  - Default (input type), 28, 28
  - default network packet size (Sybase Adaptive Server), 49
  - DefaultScript (object), 96, 87

- Define (link type), 33
  - Definitions
    - Access restriction, 166
    - Databases, 22
    - DDE command servers, 259
    - Functional right, 166
    - User profile, 166
    - User right, 166
  - Delete (property), 84
  - Departments - importing, 151
  - Desc (property), 100 , 99 , 98 , 94 , 92 , 90 , 86 , 84
  - Description strings, 24
  - Diagnostics/ Repair database (menu), 105
  - Disconnect (DDE), 263
  - Display languages, 59
    - Adding, 60
    - Object types, 61
    - Replacing, 60
    - Selecting, 60
  - dml\_locks (parameter), 46
  - Documents
    - AssetCenter Database Administrator - automatic loading, 14
    - AssetCenter Database Administrator - maximum number in memory, 17
    - Import, 151
  - Do not load for longer than (option), 288
  - Do not load more than (option), 288
  - Double-precision number (type), 27
  - DstField (object), 91
  - DstTable (object), 91
  - dtLastModif (field), 31
    - Import, 133
  - Duplicates (property), 92
  - Duration (input type), 30 , 28 , 28
- E**
- else (keyword), 76
  - elseif (keyword), 76
  - Employees - importing, 151
  - EmptyOnDup (property), 87
  - Enable automatic disconnection (option), 177
  - EndAfter (parameter), 254
  - EndBefore (parameter), 253
  - EndDate (parameter), 252
  - endfor (keyword), 73
  - endif (keyword), 76
  - endscript (function), 80
  - Enum (property), 94
  - ExecuteAction (DDE), 263
  - Execute workflow rules for execution group (module), 209
  - Exist (function), 77
  - Export database structure and data (menu), 63
  - Exporting structure and data, 63
  - extent i/o buffers (Sybase Adaptive Server), 49
- F**
- FeatClassDesc (instance), 97
  - FeatDesc (instance), 94
  - FeatParam (object), 101
  - FeatParamDesc (instance), 95
  - Feature (object), 96
  - Feature value, 34
  - Feature value (input type), 28 , 28
  - Feature values (information type memorized by a link), 34
  - FeatureValueTable (object), 85
  - FetchingArraySize (parameter), 290
  - Field (instance), 85 , 71
  - Field (object), 101 , 94
  - FieldCase (property), 87
  - FieldCount (property), 84
  - Fields
    - Default value
      - Access restrictions, 172 , 172
      - Description, 26 , 25
      - Label, 26
      - SQL names, 26
      - Types and formats, 26
      - Uniqueness of values - index, 40
  - File (option), 197
  - filter (keyword), 75
  - Filters
    - Database description, 73
    - Slow networks - tuning, 289
  - Fixed text, 72

Floating - access, 174  
 Floating point number (type), 27  
 for (keyword), 73  
 ForceDisplayScript (object), 96  
 Foreign key, 37  
 ForeignKey (property), 86  
 Formula (property), 98  
 FrameLabel (property), 100  
 FullName (field), 31  
   Precautions, 32  
 Full name (field), 31 , 25  
 FullName (property), 97  
 FuncDomain (property), 99  
 function (function), 79  
 Functional rights, 169  
   Definition, 166  
 FVSourceTable (object), 85

## G

gbbase.xml, 17  
 gbbase\*.\*, 67 , 22  
 General (tab), 197  
 GenerateColors (parameter), 254  
 GetRecordCount (DDE), 268  
 GetSelectionId (DDE), 273  
 GetValue (DDE), 275  
 Graphical representation of planning data (See  
 Planner viewer)

## H

HasClass (property), 95  
 HasFeatureValueTable (property), 84  
 HasMemoField (property), 85  
 HasNotebook (property), 99  
 HasParent (property), 97  
 HasSystemPage (property), 99  
 Highlight (DDE), 275  
 Historized (property), 90 , 86  
 History (type of information memorized by a link),  
 34  
 HistoryScript (object), 96 , 91 , 87

## I

Icons, 289

iddac.scn (scenario), 212  
 if (keyword), 76  
 Image (type of information memorized by a link),  
 34  
 Import, 127-162  
   Calculated fields, 147  
   Characters - ISO code, 128  
   Constraints, 130  
   Databases, 128  
   Database structure and data, 64  
   Departments and employees, 151  
   Documents, 151  
   Fields to import  
     Description, 139  
     Mapping, 142 , 141  
   Files, 135  
   Folders - creation, 157  
   Functioning, 128  
   Keys, 144  
   Links - creation parameters, 144  
   Multiple text files, 128  
   Numeric data, 129  
   ODBC - selecting the database, 134  
   Performance, 156  
   Process, 133  
   Recommendations, 129  
   Scripts, 161 , 159  
   Single text file, 128  
   Source tables - mapping, 141  
   Tables - description of parsing, 135  
   Text files  
     Mapping, 141  
     Structure, 134  
     Text files - selection, 134  
     Transfer - configuring, 156  
 Import database structure and data (menu), 64  
 Import scripts  
   Executing from the command line, 161  
   Saving and running, 159  
 Index (instance), 92 , 71  
 IndexCount (property), 84  
 Indexes  
   Description, 40 , 39  
   Labels, 40  
   Number, 45

- SQL names, 40
  - Insert a language in the database (menu), 60
  - Integer (16 bit) (type), 27
  - Integer (32 bit) (type), 27
  - Integer (8 bit) (type), 27
  - InternalName (property), 100, 98, 92, 90, 86, 84
  - IsCalcField (property), 101
  - IsConsolidated (property), 95
  - IsFeature (property), 101
  - IsFeatureValueTable (property), 84
  - IsField (property), 101
  - IsFirst (property), 84
  - IsHorizontalSplit (property), 100
  - IsInherited (property), 96
  - IsLast (property), 84
  - IsLink (property), 101
  - IsValidScript (object), 85
  - IsVerticalSplit (property), 100
  - IsVisible (property), 98
  - ItemColor (parameter), 255
  - Itemized list (input type), 31
  - ItemLabel (parameter), 256
- K**
- Keys
    - Import, 143
    - Examples, 151
- L**
- Label (property), 100, 99, 98, 94, 92, 90, 86, 84
  - Labels
    - Fields, 26
    - Tables, 24
  - Languages
    - AssetCenter Database Administrator, 15
  - LDAP
    - Activation, 184
    - Connections, 184
    - Implementation, 183
    - Passwords, 183, 182
    - Reminders, 182
  - LDAP authentication (See LDAP)
  - LDAP directory (See LDAP)
  - Let AssetCenter Server create the items received in the portfolio (option), 209
  - License
    - Modification, 58
  - LineLabel (parameter), 256
  - Link (instance), 89, 71
  - Link (object), 101
  - LinkCount (property), 84
  - LinkFilter (property), 96
  - Links
    - Cardinality, 34
    - Contextual links, 39
    - Description, 32, 32
    - Import, 144
    - Label, 32
    - SQL names, 32
    - Types, 33
  - Link type
    - Foreign key, 37
  - ListAllFields (DDE), 270
  - ListAllLinks (DDE), 271
  - ListAllTables (DDE), 264
  - ListOpenTables (DDE), 265
  - locking (Sybase Adaptive Server), 50
  - log\_buffer (parameter), 46
  - LOG (Microsoft SQL Server), 47
  - LOG (Sybase Adaptive Server), 49
  - LogError (function), 77
  - LoginId (property), 83
  - LoginName (property), 83
  - LongHelpComment (property), 95, 90, 86
  - LongHelpCommentNoHTMLTag (property), 95, 90, 87
  - LongHelpDesc (property), 95, 90, 87
  - LongHelpDescNoHTMLTag (property), 96, 90, 87
  - LongHelpSample (property), 95, 90, 86
  - LongHelpSampleNoHTMLTag (property), 95, 90, 87
  - LongHelpWarning (property), 95, 90, 87
  - LongHelpWarningNoHTMLTag (property), 96, 90, 87
  - Long text field (type), 27

**M**

mail.ini, 238 , 235 , 233  
 MainField (parameter), 251  
 MainIndex (object), 85  
 MandatoryScript (object), 96 , 87  
 MandatoryType (property), 133  
 MAPI (See Messaging)  
 Max. size (option), 198  
 maximum network packet size (Sybase Adaptive Server), 49  
 max server memory (Microsoft SQL Server), 47  
 MaxValue (property), 94  
 memory (Sybase Adaptive Server), 49  
 Messaging  
   Configuration, 230  
   Connection - problems, 240  
     Errors, 241  
     Test to perform if a problem is encountered, 240  
   Functioning, 229  
   Interfacing, 229-242  
   MAPI  
     mail.ini, 235  
     win.ini, 235  
   MAPI - Configuring employees, 233  
   Multiple recipients, 231  
   SMTP  
     Configuration, 231  
     Employees - configuration, 231  
     mail.ini, 233  
     win.ini, 233  
   Tests, 240  
   Using multiple protocols, 230  
   VIM  
     Configuration, 238 , 237  
     Employees - configuration, 235  
     mail.ini, 238  
     win.ini, 238  
 Messaging reconnection delay (option), 197  
 Microsoft SQL Server  
   Clients, 48  
   Databases - creation, 47  
   Server, 47  
 Migration, 109

MinValue (property), 94  
 Modified on (field), 31  
 Monetary (input type), 28 , 28  
 MSDE, 52

**N**

Name (property), 102 , 97  
 Named - access, 174  
 Neutral (link type), 33  
 Next (menu), 19  
 Normal (link type), 33  
 Normal (type of information memorized by a link), 34  
 NT services, 223  
 NullValues (property), 92  
 Number (input type), 28 , 28  
 Number of open objects (parameter), 45 , 45  
 NumValue (property), 102

**O**

Object (input type), 31  
 Object (instance), 71  
 ODBC, 111-114  
   Connections - creation, 113  
   Connection to use, 113  
   Crystal Reports - creation, 114  
   DBMS, 112  
   Driver  
     Installation, 111  
     Usefulness of, 112  
   Exposed data, 113  
   Functioning, 111  
   Languages, 112  
   Restrictions, 111  
 ODBC access (See ODBC)  
 ODBC driver (See ODBC)  
 open\_cursors (parameter), 46  
 Open an existing database (option), 17  
 Open database description file - create new database (option), 17  
 OpenTable (DDE), 267  
 OpenView (DDE), 266  
 optimizer\_mode (parameter), 46  
 Oracle, 45



- OrderBy (parameter), 258
- Overflow, 36
  - Behavior of a link, 37
  - Tables, 36
- Overflow tables, 36
  - Implementation, 37
- Own (link type), 33
- Owncopy (link type), 33

**P**

- Page (instance), 100
- PageItem (instance), 100
- ParentClass (object), 97
- Password (field), 195
- Passwords
  - (See Also LDAP)
  - Access control, 178
  - Admin login, 181
  - Creation, 182
  - Lost passwords, 182
  - Modification, 182
  - Storage, 182
- pdiac.scn (scenario), 214
- Percentage (input type), 28, 28
- Performance (See Slow networks - tuning)
- Planner viewer, 243-258
  - Options, 243
  - Overview, 243
  - Planner viewer pages
    - Adding to a table, 244
    - Adding to a wizard, 249
    - Configuration, 251
    - Practical case, 250, 245
    - Unworked days, 243
- PlannerViewer (parameter), 251
- Positive (property), 87
- Practical cases
  - Planner viewer, 250, 245
- Previous (menu), 19
- PrimaryKey (object), 85
- PrimaryKey (property), 86
- procedure cache (Sybase Adaptive Server), 49
- processes (parameter), 46
- Progress (parameter), 257

- Property (instance), 71
- Purge the input-events table (module), 215
- Purge the outgoing-events table (module), 216

**R**

- RAID, 44
- RawSource (property), 93
- Read (property), 96, 89, 86
- ReadOnly (property), 86
  - Constraints, 133
- ReadOnlyScript (object), 87
- ReferencedStorageFieldCount (property), 93
- Refresh list (menu), 68
- Regular test (option), 288
- RelDstField (object), 91
- RelevantScript (object), 91, 88, 85
- RelSrcField (object), 91
- RelTable (object), 91
- RemoveLink (DDE), 276
- ReverseLink (object), 91
- rollback segments (Oracle), 46
- Row-Level Locking (Sybase Adaptive Server), 50

**S**

- ScreenFilter (property), 99
- Screens, 292
- Script
  - Overflow link, 37
- script (function), 80
- Script (instance), 93, 71
- Script (object of an AssetCenter database description instance), 98
- ScriptField (instance), 102
- ScriptType (property), 93
- Search for new workflow execution groups (module), 216
- Security, 165
  - (See Also Access control)
- Select folder (menu), 68
- SetFilter (DDE), 272
- SetFocus (DDE), 277
- SetOutput (function), 77
- SetProperty (function), 77
- SetRecordMode (DDE), 269

SetSelection (DDE), 272  
 SetValue (DDE), 277  
 SetValueWhere (DDE), 278  
 SetViewMode (DDE), 268  
 shared\_pool\_size (parameter), 46  
 Show (DDE), 279  
 Signal presence of database server (module), 216  
 , 42  
 Size (property), 86  
     Constraints, 133  
 Slot timeout (option), 177  
 Slow networks - tuning, 287-293  
     Access restrictions, 292  
     Caches, 292  
     Lists, 288  
     Options, 287  
     Screens, 292  
 SMTP (See Messaging)  
 sort\_area\_retained\_size (parameter), 46  
 sort\_area\_size (parameter), 46  
 sort (keyword), 74  
 Sorts  
     Databases - description  
         Filtering, 73  
     Hierarchic lists, 25  
     Slow networks - tuning, 289  
 Source (property), 93  
 Split expense lines in cost centers (module), 217  
 SQLName (property), 100 , 100 , 99 , 98 , 94  
 SqlName (property), 92 , 90 , 86 , 84  
 SQL names  
     Fields, 26  
     Indexes, 40  
     Links, 32  
     Tables, 24  
     Usage, 24  
 SrcField (object), 91  
 SrcTable (object), 91  
 StartAfter (parameter), 253  
 StartBefore (parameter), 252  
 StartDate (parameter), 252  
 String (instance), 102  
 Strings - separators, 138  
 StringValue (property), 101  
 Sybase Adaptive Server, 49 , 49

SysEnumValue (instance), 101  
 SysPage (object), 99  
 System (property), 100 , 99 , 94 , 92 , 90 , 87 , 84  
 system data source (UDB DB2), 51  
 System itemized list (input type), 30 , 28 , 28

## T

Table (instance), 83 , 71  
 Table (object), 99 , 96 , 94 , 92 , 87  
 Table (object of an AssetCenter database description instance), 98  
 TableCount (property), 83  
 Table or field name (input type), 28  
 Table or field SQL name (input type), 28  
 Tables  
     Description, 24 , 24  
     Description string, 24  
     Intermediary tables, 34  
     Label, 24  
     Number, 45  
     Overflow table, 36  
     Reference tables, 36  
     SQL names, 24  
 tables.txt, 67 , 21  
     Structure, 23  
 Tablespaces (Oracle), 46  
 tempdb (Microsoft SQL Server), 48  
 tempdb (Sybase Adaptive Server), 49 , 49  
 Template.Ext (global variable of an AssetCenter database description instance), 103  
 Template.Fullname (global variable of an AssetCenter database description instance), 103  
 Template.Name (global variable of an AssetCenter database description instance), 103  
 Template.Path (global variable of an AssetCenter database description instance), 103  
 Template.Shortname (global variable of an AssetCenter database description instance), 103  
 Templates (menu), 68  
 testdde.xls, 282  
 Text (type), 27  
 Text files - importing, 134  
 Time (global variable of an AssetCenter database description instance), 102

- Time (input type), 28 , 28
- Time (type), 27
- Time zones, 57
- tpl (files), 72 , 67 , 22
- Trees in drop-down lists (option), 288
- Tree structure - import, 149
- Truncate Log on Checkpoint (Microsoft SQL Server), 48
- Truncate Log on Checkpoint (Sybase Adaptive Server), 49
- Type (instance), 91
- Type (property), 98 , 94 , 89 , 88 , 86
  - Constraints to be respected when importing data, 132
- Type-ahead after (option), 287
- Typed (property), 90
- TypeField (object), 91

## U

- UDB DB2
  - Application counter server, 51
  - Clients, 51
- Unique index
  - Precautions for deleting or modifying, 21
- Unit (property), 94
- Update (property), 96 , 89 , 85
- Update statistics for tables (module), 215
- Update the database with the results of the scanners (module), 213 , 211 , 211
- Update the database with the results of the scanners (scenario), 214
- user data source (UDB DB2), 51
- Userlogin (global variable of an AssetCenter database description instance), 102
- User profiles, 167
  - Definition, 166
- User rights, 167
  - Definition, 166
- User type (field), 27
- UserType (instance), 91
- UserType (property), 98 , 89 , 88 , 86
  - Import - constraints, 131
- UserTypeFormat (property), 89 , 86

- Use this connection in service mode (option), 190 , 189
- Use time zones (option), 57

## V

- Value (property), 102
- ValueField (object), 96
- ValueOf (function), 76
- VbReturnType (property), 93
- Verify alarms (module), 219
- Verify history lines (module), 221
- Verify local time compared to that of the server (option), 198
- Verify null-identifier records (module), 221
- Verify stocks (module), 222
- Verify time zone of database server (module), 218
- Verify time zone of database server (option), 198
- View (instance), 98
- VIM (See Messaging)
- vim32.dll, 237

## W

- WAN (See Slow networks - tuning)
- WAN network (See Slow networks - tuning)
- WhereCond (parameter), 257
- win.ini, 238 , 235 , 233
- Windows integrated security
  - AssetCenter Server, 191
  - Employees, 194
  - Functioning, 191
  - Implementation, 192
  - Supported environments, 194
- Write (property), 96 , 89 , 85

