HP Asset Manager

Software version: 9.30

Administration

Document Release Date: 31 March 2011
Software Release Date: March 2011

Legal Notices

Copyright Notices

Restricted Rights Legend

Warranty

For information about third-party and/or open source license agreements, or to view open source code, use one of the following ways:

- In the `ThirdParty` directory of the software installation CD-ROM
- In the directories in which the binary files of the third-party and/or open source tools are located after installation of the software.
- Through the component's url indicated in the **Open Source and Third-Party Software License Agreements** guide

Trademark Notices

Acknowledgements

This product includes software developed by the Apache Software Foundation (http://www.apache.org/ [http://www.apache.org/]), which is Copyright © The Apache Software Foundation. All rights reserved.

This product includes software developed by The OpenLDAP Foundation, which is Copyright ©, The OpenLDAP Foundation, Redwood City, California, USA. All Rights Reserved. OpenLDAP® is a registered trademark of the OpenLDAP Foundation.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/), which is Copyright © The OpenSSL Project. All rights reserved.

This product includes software developed by the OpenSymphony Group (http://www.opensymphony.com/), which is Copyright © The OpenSymphony Group. All rights reserved.

This product includes code licensed from RSA Data Security.

This product includes software developed by the JDOM Project (http://www.jdom.org/), which is Copyright © Jason Hunter & Brett McLaughlin. All rights reserved.

Build number: 690

Table of Contents

List of Figures

List of Tables

Introduction

Who is the Administration module intended for?

The Administration module is intended for all organizations using Asset Manager.

It is generally implemented by the following persons:

- Network administrator
- Database administrator
- Asset Manager administrator

What does the Administration module do?

The Administration module enables you to perform the following tasks:

- Manage an Asset Manager database
- Manage connections to an ODBC database
- Manage Asset Manager
- Manage Asset Manager Automated Process Manager
- 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 Asset Manager for use over a network
- Implement automatic monitoring procedures
- Create, update, delete user profiles

1 Installing License Keys

Obtaining HP AutoPass License Keys

To connect to the Asset Manager database, you will require HP AutoPass License Keys. The License Keys provided to you by HP contain authorizations to use the list of Asset Manager modules, functions and related services that you have acquired with your contract.

They also determine:

- The number of authorized users (Named and Floating).
- The number of guest users.

License keys are obtained and installed during the initial installation and can be updated periodically.

Every time you get new License Keys, they need to be added to the existing set recorded in the HP AutoPass License Management Tool.

Tip:

HP AutoPass License Management Tool stores the set of installed License Keys in the `LicFile.txt` located in the `data` folder of the HP AutoPass License Management Tool installation folder (`C:\Program Files\HP\Asset Manager 9.30 xx\autopass\data` by default).

There are several ways to obtain and install License Keys from HP. These are explained in the section Installing HP AutoPass License Keys [page 14] step.

The online help supplied with the HP AutoPass License Management Tool also provides further information on these methods (see **Help** menu), as well as for related maintenance procedures. Alternatively, please contact HP technical support.

Installing HP AutoPass License Keys

❗ Warning:

Before attempting this installation, ensure that Java (JRE 1.5 minimum) is installed on the machine on which you wish to launch the HP AutoPass License Management Tool, and add or modify the **JAVA_HOME** system variable to have it point to the JRE or JDK installation folder.

For example:

```
C:\Program Files\Java\jre6
```

The full set of License Keys need to be installed:

1. On the machine from which the database will be activated using Asset Manager Application Designer.

   ▸ Changing the HP AutoPass License Key file [page 17].

2. On the machine acting as server for Asset Manager Automated Process Manager (from where the database is periodically signed).

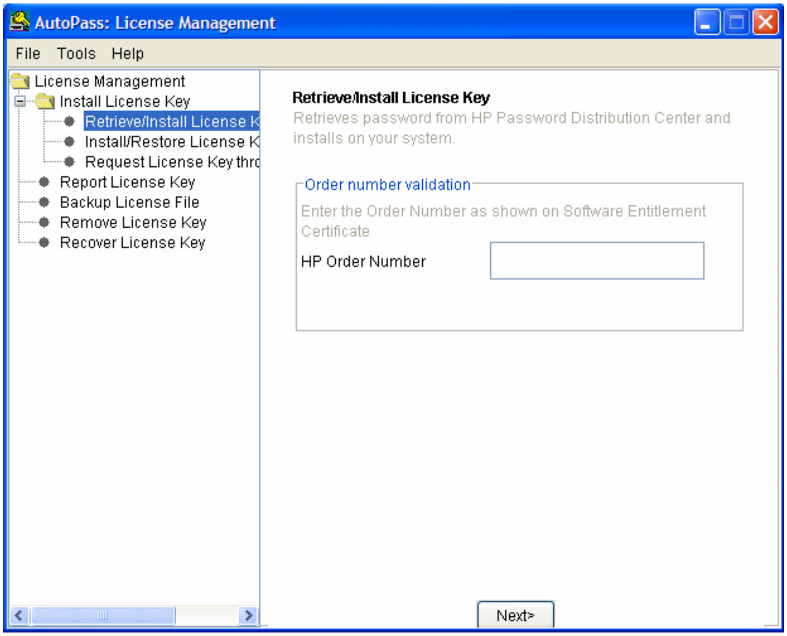
Note:

This can be the same machine.

There are several ways to obtain and install License Keys from HP.

For example, if you receive the License Keys through a file:

1. Launch the HP AutoPass License Management Tool from the Asset Manager program group, for instance **Windows Start/ All programs/ HP/ Asset Manager 9.30 en** menu. The following screen displays:



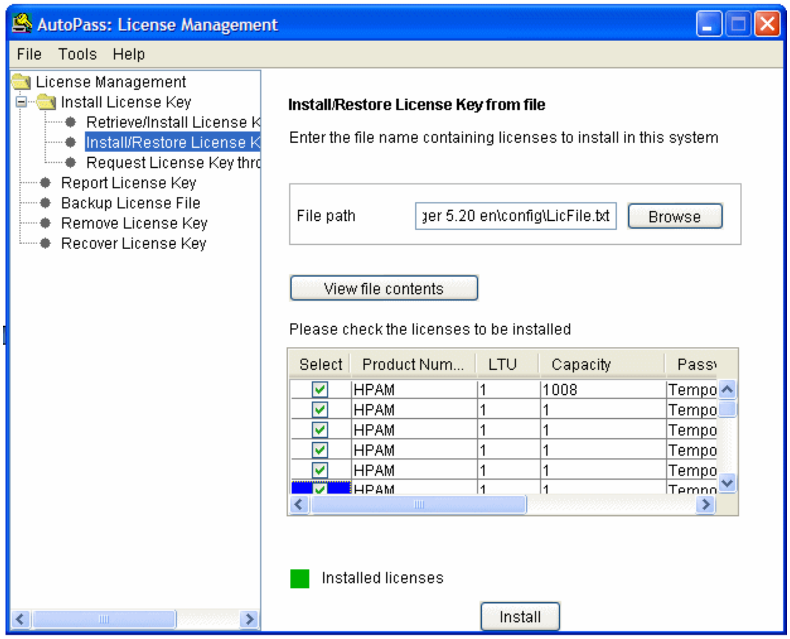
Retrieve/Install License Key

Retrieves password from HP Password Distribution Center and installs on your system.

Enter the Order Number as shown on Software Entitlement Certificate

2. Select **Install/Restore License Key from file**.
3. Click **Browse** to select the file you have received.

4 Click **View file contents**. The list of license rights is displayed, as in the following example:



Install/Restore License Key from file

Enter the file name containing licenses to install in this system

File path: ɟer 5.20 en\config\LicFile.txt

Please check the licenses to be installed

Select	Product Num...	LTU	Capacity	Passw
☑	HPAM	1	1008	Tempo
☑	HPAM	1	1	Tempo
☑	HPAM	1	1	Tempo
☑	HPAM	1	1	Tempo
☑	HPAM	1	1	Tempo
☑	HPAM	1	1	Tempo

5 Check the box in the **Select** column for all the rights you wish to install (generally all available), and click **Install**.

The rights change to green to confirm that they have beeen installed.

6 You can check which License Keys are recorded in the HP AutoPass License Management Tool by selecting **Report License Key**.

7 If you'd like to backup and/or replicate those rights to another machine, select **Backup License File**.

This creates a `.dat` file that you can install or restore using the **Install/Restore License Key from file**.

8 Select **File/ Close**.

To complete the installation, you need to carry out two more procedures:

1 Activate the database using Asset Manager Application Designer after installing a new or modified License Key file. See Changing the HP AutoPass License Key file [page 17]

2 Run Asset Manager Automated Process Manager and configure the **Signal presence of database server** (UpdateToken) module (**Tools/ Configure modules** menu), so that the database is signed on a regular basis. If possible,

this instance of the server should not be moved to another computer, since this will require changing the license file.

This signing process compares the license rights granted by HP AutoPass (local License Key file) with those activated in the database to which Asset Manager Automated Process Manager is connected. If the former rights are less than the latter, the database is locked.

Note:

We recommend that you run Asset Manager Automated Process Manager on a computer that has a short response time with the DBMS.

Changing the HP AutoPass License Key file

You must reinstall HP AutoPass License Keys in the following cases:

- If the **ITAM** License Key expires (**AM Asset Portfolio** module).
- If you receive additional License Keys.

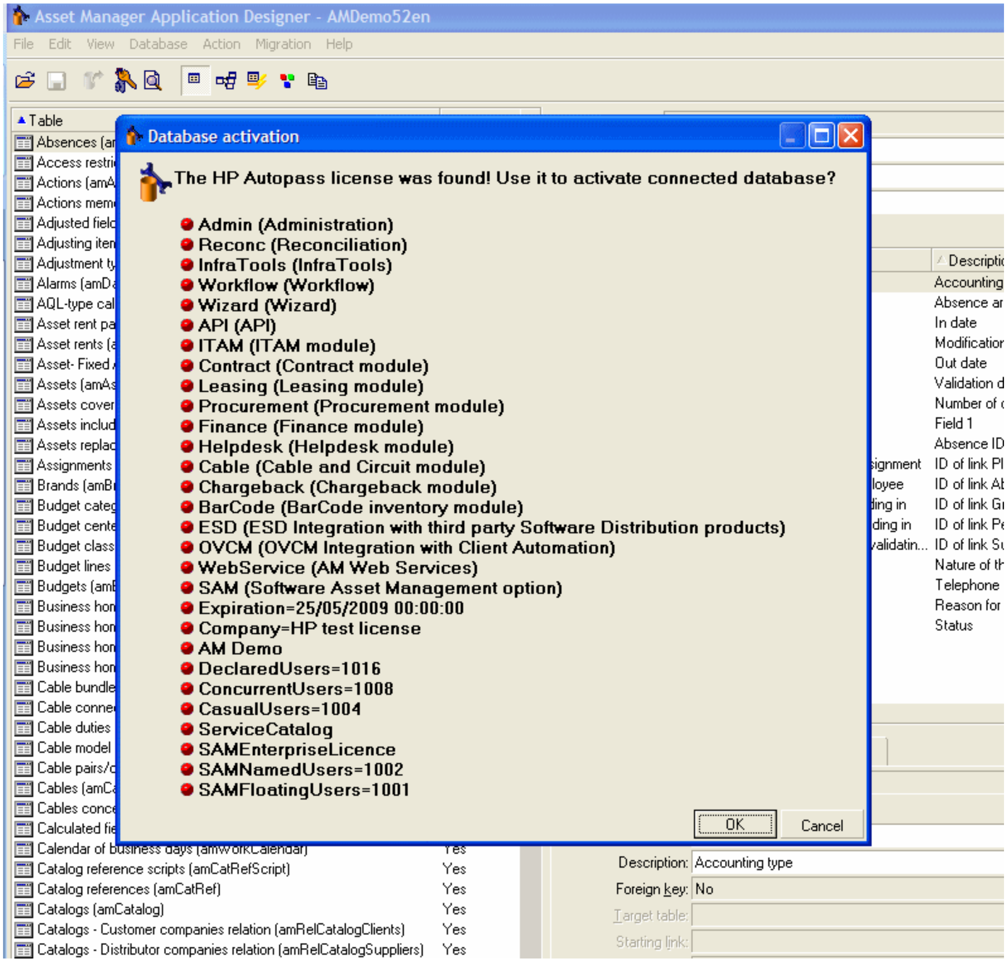
If any of these cases apply:

1. Obtain the new or addtional HP AutoPass License Keys.

   ▶ Obtaining HP AutoPass License Keys [page 13]
2. Install the new or modified License Keys.

   ▶ Installing HP AutoPass License Keys [page 14]
3. Reactivate the database with Asset Manager Application Designer:
   1. Select the **Action/ Activate database...** option. The **Database activation** screen displays as in the example below, with a list of the

authorizations granted by your contract (which you should verify at this point - resize the window if necessary to display the full list):


Asset Manager Application Designer - AMDemo52en
File Edit View Database Action Migration Help
Database activation
OK
Cancel

The HP Autopass license was found! Use it to activate connected database?

- Admin (Administration)
- Reconc (Reconciliation)
- InfraTools (InfraTools)
- Workflow (Workflow)
- Wizard (Wizard)
- API (API)
- ITAM (ITAM module)
- Contract (Contract module)
- Leasing (Leasing module)
- Procurement (Procurement module)
- Finance (Finance module)
- Helpdesk (Helpdesk module)
- Cable (Cable and Circuit module)
- Chargeback (Chargeback module)
- BarCode (BarCode inventory module)
- ESD (ESD Integration with third party Software Distribution products)
- OVCM (OVCM Integration with Client Automation)
- WebService (AM Web Services)
- SAM (Software Asset Management option)
- Expiration=25/05/2009 00:00:00
- Company=HP test license
- AM Demo
- DeclaredUsers=1016
- ConcurrentUsers=1008
- CasualUsers=1004
- ServiceCatalog
- SAMEnterpriseLicence
- SAMNamedUsers=1002
- SAMFloatingUsers=1001

Description: Accounting type
Foreign key: No
Target table:
Starting link:

b Click **OK** to activate the connected database using the displayed license.

Warning:

If you try to connect to a database that has not been activated, you will get the following error message:

- Windows client:

```
Database is not activated. You must activate database with a valid HP A
utoPass license.
```

- Web client:

```
com.peregrine.ac.AmException: Error (12 001): There is no license file
in this database. It cannot be opened
```

2 Asset Manager Application Designer - GUI

Introduction

Asset Manager Application Designer is an administration tool for Asset Manager 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 Asset Manager database, we recommend limiting its use to users with the appropriate skills.
- Asset Manager 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 Asset Manager and Asset Manager Application Designer.
- Asset Manager Application Designer 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

Asset Manager Application Designer is executed from the same program group as Asset Manager. Click its icon to launch it.

Note:

At startup, the main Asset Manager Application Designer 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

Asset Manager Application Designer's user interface is composed of three panes:

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

Note:

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

The Asset Manager Application Designer GUI shares the same look and feel as Asset Manager, such as for consultation and creation. For more information, refer to the **User interface** guide.

Basic Operations on the graphical interface

Selecting a language

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

▶ Display language according to the nature of the objects [page 67]

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 Asset Manager Application Designer used to create the database,
- Languages inserted after creating the database (**Action/ Insert a language in the database**).

▶ Modifying Asset Manager client languages [page 65]

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 **View** menu. The available objects are:

- Fields
- Links
- Indexes
- Screens
- 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 **View** 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.

File manipulation functions

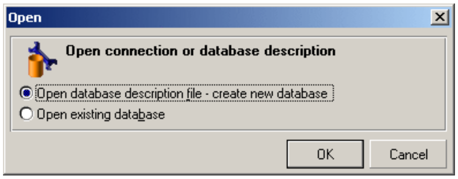
The **File** menu groups the functions relating to loading and saving a file.

Opening a file

Select **File/Open**.

The following dialog box displays:

Figure 2.1. Database opening screen


Open
Open connection or database description
● Open database description file - create new database
○ Open existing database
OK
Cancel

This dialog box enables you to choose one of two Asset Manager Application Designer 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 Asset Manager Application Designer session by selecting one of these two options.

Note:

In the last part of the **File** menu, Asset Manager Application Designer lists by default the last four opened documents. You can thus quickly open one of these documents by directly clicking it in the menu. The number of documents that appear in the recent files list can be set using the **Maximum number of recently opened documents to store in File menu** option (click **Edit/Options**, then expand the **Documents** branch).

Open database description file - create new database

To create a new database Asset Manager Application Designer needs a description of that database. This description comes in the form of a file that contains the structural information relating to an Asset Manager database. It plays the role of a data template during the creation of the database.

By validating this choice, you must provide Asset Manager Application Designer the description file `gbbase.xml` located in the `Config` sub-folder of the Asset Manager 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 Asset Manager database before being able to customize it. Selecting this option will display the connection that is also used for Asset Manager.

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 Asset Manager Application Designer session. If you have made any modifications, Asset Manager Application Designer 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 - Asset Manager Application Designer 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/ Update 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 Asset Manager Application Designer. If you have made any modifications during the session, Asset Manager Application Designer will ask you if you want to save them.

Edit functions

Asset Manager Application Designer proposes all the classic edition functions:

Cut/Copy/Paste

The **Edit** menu groups 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.

Search

Asset Manager Application Designer 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 Asset Manager Application Designer 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 Asset Manager Application Designer. Each option is described in the **Description** zone of the dialog box.

3 Standard database description files

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 Asset Manager API.

Definition of an Asset Manager database

An Asset Manager 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.

Asset Manager comes with a demonstration database.

With Asset Manager, you can create several databases and open them one-at-a-time per 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.

Introduction to the database description files

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.

Ways to obtain

There are several ways to obtain a description of the Asset Manager 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 Asset Manager 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 Asset Manager Application Designer with a connection to your database.

- Asset Manager Application Designer program: It enables you to freely create description files of the Asset Manager database (tables, fields, links and indexes).

  It makes use of:

  - The Asset Manager database description files (**gbbase*.*** files) or a connection to the Asset Manager 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 Asset Manager, 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 Asset Manager program

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 Asset Manager installation folder.

Note:

These files were created using Asset Manager Application Designer and the **dbase.tpl** and **tables.tpl** templates. These templates are located in the **doc\infos** sub-folder of the Asset Manager 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.

Structure of the `database.txt` file

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.

Structure of the `tables.txt` file

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 Asset Manager language version to the next; they are always the same.

Description of the tables

This section describes the tables in the Asset Manager database:

- Identification of Asset Manager tables
- String describing the Asset Manager tables

Identifying Asset Manager tables

Each table in the Asset Manager database is described by:

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

  It is used when Asset Manager 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 Asset Manager has sufficient space available to display the entire description (status bar, for example).

Table description strings

You use Asset Manager Application Designer to define the description strings used for Asset Manager tables.

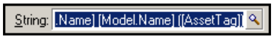
In an Asset Manager table, the description string describes those records that appear in drop-down lists and that enable you to select linked records.

It also defines the name that appears in the title bar of the Windows client when 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:



String: .Name] [Model.Name] ([AssetTag]

In this table, the title of the detail window for an asset whose brand (**Brand.Name**) is "Asus", model (**Model.Name)** is "AsusLX512" model and asset tag (**AssetTag**) is "1" will be:

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

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

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 Asset Manager database:

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

Identifying Asset Manager fields

Each field in an Asset Manager table is described by:

- An "**SQL name**". SQL names are in English. SQL names do not change from one Asset Manager 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**". Asset Manager 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 Asset Manager status bar.

Field types ("Type") and data-entry formats ("User type")

This section lists:

- The type of Asset Manager fields.
- The data-entry formats and data types for Asset Manager 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 Asset Manager Application Designer.
- Read the **database.txt** file, located in the `doc\infos` folder of the Asset Manager installation folder. The **Field Type** column details the field types used.
- Display the context-sensitive help for that field (Windows client only).

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

Provides the field's storage format.

Table 3.1. Fields - type

Value given in **Field Type** column in `database.txt` file	Value in **Type** field in Asset Manager Application Designer	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.
Date	Date	**Date** format field only (no time).
Time	Time	**Time** format field only (no date).
Date and time	Date and time	Date and time.
Variable length binary field	Variable-length binary field	Used to store images, or other binary files, 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.

Data-entry format and data type ("User type")

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

- The value of the **User type** in the **General** tab of Asset Manager Application Designer: 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 3.2. Data types - Entry format

Value given in **Field data display and entry type** column in `database.txt` file	Value in **User type** field in Asset Manager Application Designer	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**.
Boolean (Yes/No)	Boolean (Yes/No)	Boolean.
Number	Number	Number.

Value given in **Field data display and entry type** column in `database.txt` file	Value in **User type** field in Asset Manager Application Designer	Description
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
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 if the **Type** (seType) is **open**.
Value of feature	Value of feature	Reserved. Do not use.
Percentage	Percentage	A field with this format contains percentages. This **Percentage** format displays values with two decimal places.
Duration	Duration	**Duration** type values. The display format and default unit are defined by the **UserTypeFormat** property.
Table or field name	Table or field name	SQL name of a table or a field.
BASIC script	BASIC script	Reserved. Do not use.
N/A	Unknown	Any other type.

Note:

On list and detail screens, the format of fields whose data type is set to **Date**, **Date and time**, **Time** or **Number** cannot be modified by clients.

You can display additional field information if you set the **User type** field in Asset Manager Application Designer to one of the following values:

Table 3.3. Value of the User types field

Value of **User type** field in Asset Manager Application Designer	Additional information
Duration	This field enables you to define the display format and the default unit of a duration: Syntax: `%[.D1]U1[l][d][%[.D2]U2[l][d]]...[%[.Dn]Un[l][d]]` - **Ux** is one of the following units: - 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**. - **l** is used to specify that the duration should be displayed its long format. If **l** is omitted, the unit will be displayed in its shortened form. Example: **%Yl** 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**. Example if the duration format is **%Ml%D%Hd%.1N**: - If you enter **50**, Asset Manager takes this to mean **50 hours** and stores **180 000** seconds. - If this stored value is **5 461 220** seconds, Asset Manager displays **2 months 3d 5h 2.3min**.
System itemized list	Values taken by the itemized list.
Custom itemized list	Name of the itemized list.

dtLastModif fields

The field with the SQL name "dtLastModif" exists in all tables in the Asset Manager database.

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

This field is updated each time a record is modified in the Asset Manager 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 **Full name** 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/".
  ```

- **Employees and departments** (amEmplDept) 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 Asset Manager.

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 Asset Manager 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 Asset Manager language version to the next; they are always the same.
- Its "**Label**".
- Its "**Description**".

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

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

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

Warning:

The label and description of the link are different from the label and description of the target table. This is because several links can exist between Asset Manager tables. For example, between the Portfolio items table and the **Employees and departments** (amEmplDept) 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 3.4. 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 3.5. 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 Asset Manager:

- "1-->1" links: A record in table A can only be linked to a single record in table B. For example, the **Employees and departments** 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:

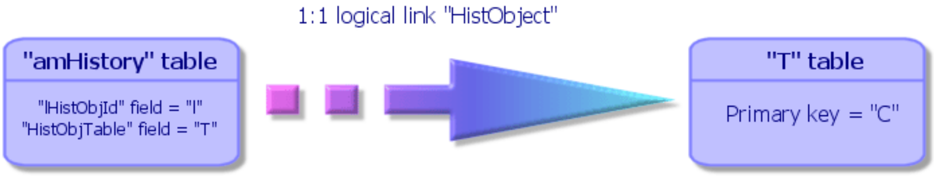
- 1 links: A record in table A can only be linked to a single record in table B, and vice-versa. A 1 link between two tables in Asset Manager is represented by two "1-->1" links.
- N links: A record in table A can be linked to several records in table B, but a record in table B is only linked to a single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

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 3.1. Links between the History table and a target table



Tables derived from links

Intermediary tables

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

They do not appear in the Asset Manager 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 3.2. Intermediary tables - utilization

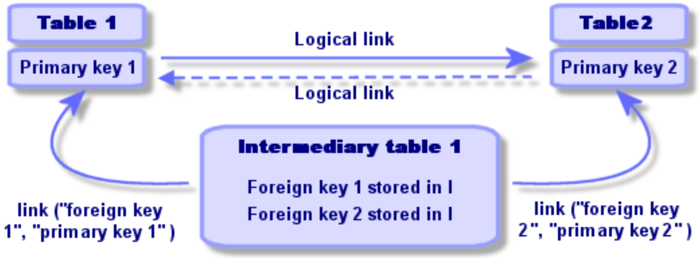
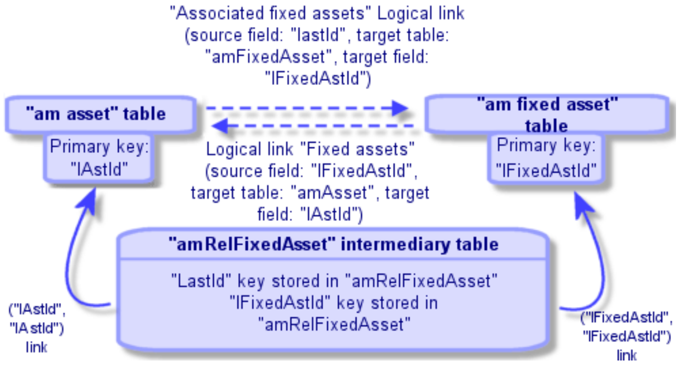


Table 1
Primary key 1
Logical link
Logical link
Table2
Primary key 2
Intermediary table 1
Foreign key 1 stored in I
Foreign key 2 stored in I
link ("foreign key 1", "primary key 1")
link ("foreign key 2", "primary key 2")

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

Figure 3.3. Link between the Assets table and the Fixed assets table


"Associated fixed assets" Logical link (source field: "lastId", target table: "amFixedAsset", target field: "lFixedAstId")
"am asset" table
Primary key: "lAstId"
"am fixed asset" table
Primary key: "lFixedAstId"
Logical link "Fixed assets" (source field: "lFixedAstId", target table: "amAsset", target field: "lAstId")
"amRelFixedAsset" intermediary table
"LastId" key stored in "amRelFixedAsset"
"lFixedAstId" key stored in "amRelFixedAsset"
("lAstId", "lAstId") link
("lFixedAstId", "lFixedAstId") link

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 Asset Manager 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.

Note:

For overflow links, it is highly recommended to add a relevance script corresponding to the case when the overflow link is not populated. **Example:** amPortfolio:Computer is irrelevant when

```
Retval = [amPortfolio:Model.Nature.OverflowTbl]<>"amComputer
```

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 other tables:

- 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 Asset Manager Application Designer,
- Verify its behavior in Asset Manager.

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 Asset Manager 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)

Description of indexes

This section describes the indexes in the Asset Manager 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 Asset Manager 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 Asset Manager Application Designer, 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.

4 Creating, modifying and deleting an Asset Manager database

This chapter explains how to:

- Create an Asset Manager database:

- Modify certain parameters of an Asset Manager database:

- Delete an Asset Manager database:

Prerequisites of creating an Asset Manager database

A successful installation requires the following components installed in one of following two ways:

- Asset Manager Application Designer (and Asset Manager) installed on one machine; with Asset Manager Automated Process Manager and HP AutoPass installed on another.
- Asset Manager Application Designer, Asset Manager Automated Process Manager and HP AutoPass installed on a single machine.

▶ The Asset Manager **Installation and upgrade** guide

You also need to have obtained and installed License Keys, as explained in Installing License Keys [page 13]

Creating an empty shell with the DBMS

Before creation

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 Asset Manager and the scalability of your architecture.

  There is no one single correct way to create a database. It 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 HP 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 and UDB DB2 use the term **database**, Oracle uses the term **schema**.

Process of the creating a database

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 Asset Manager connections. In this case, you must define update rights at the level of each table of the Asset Manager database.

These steps are described below for each supported DBMS.

Information to be taken into account when reserving space for the database

The Asset Manager 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 Asset Manager Application Designer.

Oracle

Table 4.1. Important server parameters when creating the Oracle instance

Parameter	Recommended value
CHARACTER SET	- To store text in non-Asian languages (English, French, German, Italian, Spanish, Russian, etc) you can use a non-Unicode character set (instead of UTF8). - To store text in all languages, including Asian languages (Japanese, simplified Chinese, Korean), use the **UTF8** character set: (▶ Languages that can be used in text fields [page 71]) - To store text in a given Asian language (Japanese, simplified Chinese, Korean), you can use the character set corresponding to your language instead of UTF8. - Disadvantages: The number of characters that you can store in the text fields is divided by two. - Advantages: Performance is better as opposed to when the UTF8 character set is used. Note: 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 **8192**. Note: With certain Oracle versions, this parameter cannot be modified once the database is created.

Parameter	Recommended value
db_block_buffers	Size of the database memory cache. This size is expressed in units of **db_block_size**. For example: to obtain a **200 MB** cache: ■ `db_block_buffers = 25000` if `db_block_size = 8192` ■ `db_block_buffers = 100000` if `db_block_size = 2048` Note: 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).
shared_pool_size	This parameter is expressed in bytes. This size varies from **9,000,000** to **15,000,000** 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 **large_pool_size** on Oracle 8i).
log_buffer	This parameter is expressed in bytes. We recommend using **163840**.
processes	`<Number of concurrent users> + <Number of Oracle system processes>.` Note: This value is defined according to the rights granted by the Oracle license. For example: **50** is acceptable for 10 users at maximum.
dml_locks	**500**
open_cursors	`<Number of concurrent users> * Max [30; <Maximum number of records updated in multiple selection>].`
optimizer_mode	Use the **RULE** mode during initial import, then the **FIRST_ROWS** or **ALL_ROWS** mode as soon as the DB has been imported and statistics concerning tables and indexes calculated. (It is a function of Asset Manager Automated Process Manager.)
sort_area_size **sort_area_retained_size**	■ On non-MTS servers ■ **sort_area_size** corresponds to the number of bytes, dedicated for sorts in memory. ■ **Sort_area_retained_size** 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 `65536*number of concurrent users`, then increase it if necessary according to use.

Verifications before creating the database with Asset Manager

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 Asset Manager database. For example, if you install Asset Manager 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 Asset Manager 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, 2000 or 2005

An SQL Server 7.0 or 2000 is prepared using the Microsoft SQL Enterprise Manager utility. The administration utility to use for SQL Server 2005 is **SQL Server Management Studio Express**.

Table 4.2. 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). The system must not swap.
auto create statistics	As set in the database template at database creation.	Database setting, to be enabled for all Asset Manager databases.
auto update statistics	As set in the database template at database creation.	Database setting, to be enabled for all Asset Manager 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 Asset Manager database (at least **150 MB** of **DATA** for a small Asset Manager 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 statitics', 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 on each of the client workstations:

- If you have installed SQL Server 7.0 or 2000, use the ODBC SQL Server driver.
- If you have installed SQL Server 2005, you can use the SQL Native Client driver or the SQL Server ODBC driver. The SQL Native Client driver provides better performance than the standard SQL Server ODBC driver and lets you access SQL Server 2005 specific features.

Important:

The current version of the SQL Native Client driver has a known error that impacts certain SQL features (i.e. when using queries).

To workaround this in Asset Manager you can set the value of **StmtCache** to **0**. The **StmtCache** parameter is used to increase the performance of queries sent to the SQL server. By setting its value to 0, server performance is decreased.

This operation is done automatically when you select the SQL Native Client driver. However, after creating the connection between the client workstation and the server using this driver, you can add the **StmtCache** parameter manually in the **[Connexion]** section of Asset Manager's **amdb.ini** file. This will improve performance.

For example:

```
StmtCache=30
```

DB2 UDB

The information that follows should be considered after you configure the DBMS, but before you create the Asset Manager database in the DBMS:

- The following operations are required for Asset Manager 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:

1

```
CREATE DATABASE <Database name>
```

Note:

In order to store text in all languages, including Asian languages (Japanese, simplified Chinese, Korean), specify the **UTF-8** character set in the **CODESET** parameter (▶ Languages that can be used in text fields [page 71]). For example:

```
CREATE DATABASE <Database name> USING CODESET UTF-8
```

If you want to store text in only one Asian language (Japanese, simplified Chinese, Korean), you can use the character set that corresponds to your language (CODESET parameter).

- Disadvantages: The number of characters that you can store in the text fields is divided by two.
- Advantages: Performance is better as opposed to when the UTF-8 character set is used.

2
```
Update database configuration for <Database name> using APPLHEAPSZ 2048
```

3
```
Update database configuration for <Database name> using APP_CTL_HEAP_SZ
2048
```

4
```
Update database configuration for <Database name>; using DBHEAP 4096
```

5
```
Update database configuration for <Database name>; using LOGFILSIZ 500
```

6
```
Update database configuration for <Database name>; using DFT_QUERYOPT 2
```

7 If you wish to use the Asset Manager database with the archival function, DB2 will need to be able to handle 8K pages.

Note:

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.

For example:

```
db2 CREATE Bufferpool BP8K IMMEDIATE  SIZE -1 PAGESIZE 8 K ;

db2 CREATE REGULAR TABLESPACE TB8K PAGESIZE 8 K  MANAGED BY AUTOMATIC S
TORAGE BUFFERPOOL BP8K ;

db2 CREATE SYSTEM TEMPORARY TABLESPACE TB8KTMP PAGESIZE 8 K  MANAGED BY
AUTOMATIC STORAGE BUFFERPOOL BP8K ;
```

8 If you need to defragment IDs, your database will need to use a specific, temporary workspace.

The temporary workspace can be activated in DB2 by the following command:

```
db2 CREATE USER TEMPORARY TABLESPACE TEMPTBSPACE MANAGED BY AUTOMATIC S
TORAGE
```

Note:

Defragmenting database IDs limits the number of IDs used as keys for records.

IDs are defragmented using the command line version of Asset Manager Application Designer.

The syntax is as follows:

```
amdbal -iddefrag:<Connection name>;<Login of user with administration r
ights>;<Password associated with the login>
```

To enable DB2 to process Asset Manager queries, you must set a value that is at least **12288** (12Kb) for the **STMTHEAP** parameter (heap size for SQL instructions).

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**):

1.
   ```
   CATALOG TCPIP NODE MyNode REMOTE CALIFORNIUM SERVER 50000 REMOTE_INSTAN
   CE INST_1 SYSTEM CALIFORNIUM OSTYPE NT
   ```

2.
   ```
   CATALOG DATABASE SAMPLE AS SAMPLE AT NODE MyNode
   ```
3. Then from a standard command prompt:

   ```
   db2icrt MyNode
   ```

Creating the DBMS login

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

It is also possible to configure Asset Manager 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 Asset Manager 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 Asset Manager identified by <password> default tablespace <ass
et> temporary tablespace <temporary_data> profile default;
grant connect, resource, create session to Asset Manager;
```

Database password encryption

Asset Manager previously used its own security algorithm to encrypt database passwords stored in the `amdb.ini` file. In version 9.30, sign-in security is enhanced with the stronger, industry-standard **AES 256** algorithm.

- When you create a new connection to a database using the **File/ Manage connections...** Windows client menu, or modify sign-in details of an existing connection, the password stored in the `amdb.ini` file will be encrypted using AES 256.
- When opening an existing database connection from `amdb.ini`, Asset Manager is backward compatible with the previous encryption system and can connect using the old password.

If you wish to enhance security of an existing database connection with AES 256 encryption, you should change the old password. In this way the old password, encrypted with the weaker method, will no longer be usable.

For details of the new method, see http://en.wikipedia.org/wiki/Advanced_Encryption_Standard.

Creating a connection with Asset Manager

Process of declaring a connection to an empty shell

1. Start Asset Manager.
2. Asset Manager 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 a demostration database using MS SQL Server [page 58]
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 Asset Manager.

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

Useful information for creating a connection to a demostration database using MS SQL Server

Field	Value
Name	<Name of your choice>
Engine	Microsoft SQL Server
Datasource	AMDemo93en (SQL Server)
Database	<Name of the empty shell created with the MS SQL Server editor>
User	itam
Password	password

Field	Value
Owner	<Leave this field empty>

Creating the database structure with Asset Manager Application Designer

We are going to use Asset Manager Application Designer to create the Asset Manager database structure inside the empty shell created with the DBMS.

This structure is based on a description file installed with Asset Manager.

Process of creating the database structure

1. Start Asset Manager Application Designer.
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 `gbbase.xml` file, which is located in the `config` sub-folder of the Asset Manager installation folder (in general: `C:\Program Files\HP\Asset Manager 9.30 xx\config`).
   e. Open the file (**Open** button).
3. In order to store text in all languages, including Asian languages (Japanese, simplified Chinese, Korean):
   1. Select the **Database/ Use Unicode text fields** menu item.
   2. Save the gbbase.xml file (**File/ Save** menu).

   Warning:

   For Oracle and DB2, this option can only be enabled if the database uses the UTF8 character set. ▶ Creating a database at the DBMS level [page 49]

Note:

This option is not required if you want to store text in only one Asian language (Japanese, simplified Chinese, Korean).

- Disadvantages: The number of characters that you can store in the text fields is divided by two.
- Advantages: Performance is better as opposed to when the UTF8 character set is used.

4 Start the wizard to create the database structure (**Action/ Create database...** menu).

5 Populate the pages of the wizard (navigate through the wizard pages using the **Next** and **Previous** buttons).

▶ Parameters in the Create database wizard [page 61]

Note:

If you are using Oracle as your DBMS and if an error message appears when you import data, check the value of the **NLS_LANG** parameter in the Windows **Registry**.

NLS_LANG must correspond to the Asset Manager Application Designer installation language. To do this:

1. Open a DOS command prompt (**Start/ Run...** Windows menu) and enter the following command:

   ```
   regedit
   ```

   This starts the **Registry Editor**.
2. If, for example, you are using Oracle version 10g, expand this branch: **My computer\ HKEY_LOCAL_MACHINE\ SOFTWARE\ ORACLE\ KEY_OraClient10g_home2**.

   Please read the information at the following site for the branch names used for other Oracle versions:

   http://www.oracle.com/technology/tech/globalization/htdocs/nls_lang%20faq.htm#_Toc110410552
3. Double-click the **NLS_LANG** parameter to modify it.

   If, for example, Asset Manager Application Designer is installed in English, enter this value: **AMERICAN_AMERICA.WE8MSWIN1252**.
4. Click **OK** to save your changes.

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

Parameters in the Create database wizard

Generate SQL script / Create database page:

Fields	Value
Database	Select the connection to the empty shell created with Asset Manager. ▶ Creating a connection with Asset Manager [page 58]
Creation	- **Create the database**: Select this option if you wish to create a database with the standard SQL script. - **Generate the SQL script**: Select this option if you wish to generate the standard SQL database creation script in order to modify it. In this case: 1 Finish executing the database creation wizard, which will simply create a SQL script file. 2 Modify the SQL script. 3 Execute the database creation wizard once more, this time selecting the option **Create the database using a script**. - **Create the database using a script**: Select this option if you wish to create a database using a SQL script created using the **Generate the SQL script** option that you have subsequently modified yourself. Note: Subsequent pages of the wizard depend on the option that you select here.
Use advanced creation options	Select this option if you need to define the tablespaces for tables, indexes and LOBs. Note: The tablespaces must be created beforehand using the DBMS.

If **Generate the SQL script** is selected in the previous page, the **Select script** page displays:

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

Fields	Value
Separator	Select the separator for SQL statements: ■ **/**: For an Oracle database ■ **GO**: For all other databases ■ **Any other separator**: as long as it is a valid separator (such as **;**). You are free to define your own separator such as **<MySeparator>**, but the database creation script will no longer work.

If **Create the database**, **Create the database using a script** or **Import line-of -business data** is selected in the previous page, the **Creation parameters** page displays:

Fields	Value
Creation script	Enter the full path of the SQL script file to create. (This field does not appear if you select the **Create the database** option on the first page of the wizard).
Password	Password to associate with the administrator login. Note: The Asset Manager database adminstrator is the record in the **Employees and departments** (amEmplDept) table for which the **Name** (Name) field is set to **Admin**. The database connection login is stored in the **User name** (UserLogin) field. The administration name is **Admin**.
Confirmation	Confirm the value entered in the **Password** field.

If **Create the database** or **Create the database using a script** is selected in the previous page, the **Create system data** page displays:

Fields	Value
Use time zones	Select this option in order to be able to manage the time-zone differences between the database server, the date field and the Asset Manager clients. ▶ **Advanced use** guide, chapter **Time zones**.
Server	Select the time zone of the Asset Manager database server. 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.

Fields	Value
Use help on fields	Select this option to insert the information describing the database fields and links.

If **Import line-of -business data** is selected in the previous page, the **Data to import** page displays:

Fields	Value
Available data	Select the sample data or line-of-business data to import into the database. For a production database, we recommend that you: ■ Select the line-of-business data for the modules you wish to use. ■ Do not select the sample data. Note: The data that you do not import when creating the database can be easily imported later.
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.

Updating the database structure

Asset Manager Application Designer 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 Asset Manager database, refer to the **Migration** guide.

Signing the database with Asset Manager Automated Process Manager

Run Asset Manager Automated Process Manager and configure the **Signal presence of database server** (UpdateToken) module (**Tools/ Configure modules** menu), so that the database is signed on a regular basis.

Note:

This must be done on a machine where HP AutoPass License Management Tool is installed with a set of License Keys that matches the one activated for the database (HP AutoPass is automatically installed with Asset Manager Automated Process Manager, but installing the License Keys to this instance of HP AutoPass is a manual step; ▶ Installing HP AutoPass License Keys [page 14]).

The signing process compares the License Keys granted by HP AutoPass with those activated in the database to which Asset Manager Automated Process Manager is connected. If the former keys are less than the latter, the database is not signed.

If a database remains unsigned for more than a week, the database is locked.

To unlock the database:

1. Resolve the the License Key mismatch between HP AutoPass and the database.

   To do this, you can either:

   - add the missing License Keys to the HP AutoPass instance on the Asset Manager Automated Process Manager machine.

     ▶ Installing HP AutoPass License Keys [page 14]
   - or remove the excess License Keys from the database:
     a. On the machine used to activate the database with Asset Manager Application Designer, run HP AutoPass License Management Tool and use **Remove License Key**.
     b. Select the License Keys to remove.
     c. Click **Remove**.
     d. Reactivate the database with the right set of License Keys (▶ Changing the HP AutoPass License Key file [page 17]).
   - or reinstall the License Keys from scratch:
     a. On the machine used to activate the database with Asset Manager Application Designer, delete the `LicFile.txt` located in the `data` folder of the HP AutoPass License Management Tool installation folder (`C:\Program Files\HP\Asset Manager 9.30 xx\autopass\data` by default).
     b. Add the required License Keys to the HP AutoPass instance on the Asset Manager Automated Process Manager machine.
2. Run Asset Manager Automated Process Manager and run the **Signal presence of database server** module again.

   The database should be unlocked.

Modifying Asset Manager client languages

Overview

Certain Asset Manager database objects are multilingual.

▶ Display language according to the nature of the objects [page 67]

In order for multilingual objects from the Asset Manager 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 65]

2. Next:

   - Windows client: install the corresponding language version of Asset Manager client on the user workstation; Multilingual database objects are automatically displayed in the language of the client).

     ▶ Displaying a given language at the Asset Manager client level [page 67]

   - Web clients: install the corresponding language version of Asset Manager Web (Asset Manager Web Tier and `aamapi93.dll` used by Asset Manager Web Service); Multilingual database objects are automatically displayed by the Web client (the URL points to a language specific version of Asset Manager Web).

     ▶ Displaying a given language at the Asset Manager client level [page 67]

     ▶ **Installation and upgrade** guide, chapter **Installing, configuring, removing and updating Asset Manager Web**, section **Installing Asset Manager Web/ Preparing your Asset Manager Web installation / Components to install**.

It is possible to use any language supported by the Asset Manager software suite.

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

Adding language parameters

Databases are created with Asset Manager Application Designer using the parameters corresponding to the language version of Asset Manager Application Designer.

To add other language parameters:

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

Tip:

It is possible to just install Asset Manager Application Designer.

3. Start Asset Manager Application Designer 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 Asset Manager is installed.

       For example: `C:\Program Files\HP\Asset Manager 9.30 xx-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.

Substituting fonts to display languages that do not use ANSI code pages in the Windows clients

In order to display all user interface items correctly (data, menus, wizards, scripts, navigation bar, etc) in languages that do not use ANSI code pages (non-Western languages), you must:

- Substitute the fonts that are used to display user interface items:
    1. Open the `am.ini` file in a text editor.

       Location of this file: ▶ **Installation and upgrade** guide, chapter **.ini and .cfg files**.
    2. Add the following lines to the corresponding sections:

```
[FontSubstitutes]
Microsoft Sans Serif=Arial Unicode MS
MS Sans Serif=Arial Unicode MS
Times New Roman=Arial Unicode MS
Verdana=Arial Unicode MS
Arial=Arial Unicode MS

[Option]
/ShortcutFavorite/Extended/SHORTCUT_FAVORITE_FONTNAME=Microsoft Sans
Serif
/Advanced/UseHardCodedFontMapper=1
/Advanced/ScriptFont=Arial Unicode MS, 12
```

    3. Save your changes.

- Make sure that the Windows system fonts are Unicode-compatible:
    1. In the Windows display properties (**Control panel/ Display**), **Appearance** tab, click **Advanced**.
    2. Check that the fonts used for the different interface items support Unicode (for example, select the **Arial Unicode MS** font).

Displaying a given language at the Asset Manager client level

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

1. - Windows client: Install an Asset Manager Windows client in the required language.
   - Web client: Install the corresponding language version of Asset Manager Web (Asset Manager Web Tier and Asset Manager Web Service).
2. - Windows client: Start the Asset Manager Windows client.
   - Web client: Start Internet Explorer and enter a URL that points to the instance of Asset Manager Web Tier of the required language.
3. Connect to the database.

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

Note:

At the level of an Asset Manager client, multilingual objects can only be displayed in one language: that of the Windows client or Asset Manager Web Tier itself.

Display language according to the nature of the objects

The display language of an object in Asset Manager depends on the nature of the object:

Table 4.3. Asset Manager Windows clients - display language for labels, screens, pages, etc.

Nature of the object	- Labels and descriptions of tables, fields, links and indexes - Screen label - Page label - System itemized lists value - Extended help on a field or a link
Storage of text of the object	Database
Display language by the Windows Asset Manager client	Language of the Asset Manager client connected to the database
Display language by the Web Asset Manager client	Language of the Asset Manager Web instance to which the URL of the Web client points

Table 4.4. Asset Manager Windows clients - display language for data

Nature of the object	Data stored in the database
Storage of text of the object	Database
Display language by the Windows Asset Manager client	Language in which the user enters the data

Nature of the object	Data stored in the database
Display language by the Web Asset Manager client	Language in which the user enters the data, except for: ■ **Monetary** and **Time** fields in details: In read and edit mode, they are displayed according to the **<default-locale>%locale%</default-locale>** parameter in the `webclient-faces-config.xml` file of the deployed Asset Manager Web Tier `AssetManager.war` file. ■ **Date and time** fields in details: **Edit** mode: the way the fields are displayed cannot be changed by users and is the same for all language versions of Asset Manager. **Read** mode: they are displayed according to the **<default-locale>%locale%</default-locale>** parameter in the `webclient-faces-config.xml` file of the deployed Asset Manager Web Tier `AssetManager.war` file. ■ **Duration** fields in details: In read and edit mode, the way the fields are displayed cannot be changed by users; it depends on the language version of Asset Manager Web Tier. ■ **Date** and **Date and time** fields in lists (main, page and wizard lists): the display format cannot be changed by users; it depends on the regional settings associated to the user account session running Asset Manager Web Service. Warning: The regional settings asociated to the user account session running Asset Manager Web Service should match the language of Asset Manager Web Service. Otherwise some strings may be displayed using an unwanted format.

Table 4.5. Asset Manager Windows clients - display language for tip of the day

Nature of the object	Tip of the day
Storage of text of the object	Database
Display language by the Windows Asset Manager client	Language of Asset Manager Application Designer, used to create the database
Display language by the Web Asset Manager client	Not available

Table 4.6. Asset Manager Windows clients - display language for program messages

Nature of the object	Program message
Storage of text of the object	Client program
Display language by the Windows Asset Manager client	Language of the Asset Manager client connected to the database, except for some messages generated at the database level.
Display language by the Web Asset Manager client	Language of the Asset Manager Web instance to which the URL of the Web client points, except for some messages generated at the database level.

Certain database records require additional information:

Table 4.7. Asset Manager Windows clients - display languages according to the record type

Record type	Comment
SAP Crystal Reports	You can display a report regardless of the language used to create it. However, labels are displayed in the language that was used for the creation of the report. If you want the labels to be available in other languages, you must duplicate the report and translate it. Some strings may depend on the language of the `aamapi93.dll` file used.
Actions and wizards	Same principle as the reports.
Queries	They are language independent
Demonstration databases	They are monolingual (installation language of Asset Manager). You can add other language parameters in the same way you would for your production database.
Workflow schemes	You can execute a workflow scheme, no matter what language was used to create it. However, labels are displayed in the language that was used for the creation of the scheme. You cannot duplicate a workflow scheme because the workflow instances would then also be duplicated.
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 Asset Manager on a client desktop, only `pdf` files matching the installation language are installed. If you need `pdf` files in other languages, simply copy them from a desktop on which they are installed.

Languages that can be used in text fields

You can store and view any character from any language (Western, Asian) that can be entered via the Windows or Web client, given that you perform the following first:

- To adapt an existing database, the format of the database must be migrated to a 9.3 format database. Then:
  - If your DBMS is SQL Server:
    1. Start Asset Manager Application Designer.
    2. Connect to the database.
    3. Select the **Database/ Use Unicode text fields** menu item.
    4. Save (**File/ Save** menu).
- To create a new database:
  1. When you create the empty shell via the DBMS, perform the actions required by the DBMS you are using.

     ▶ Creating a database at the DBMS level [page 49].
  2. When you create the database with Asset Manager Application Designer, enable the **Use Unicode text fields** option on the **Database** menu.

     ▶ Creating the database structure with Asset Manager Application Designer [page 59]

Note:

This feature has the inconvenience of doubling the size of the database, which can hinder performance.

Note:

If a database is not set to store Unicode text fields, you can still enter and save characters to the database but they will not be saved correctly. Consequently, when you display the record, the information in the text field will not be displayed correctly.

Tip:

This feature also provides the Japanese, Korean and simplified Chinese versions with full-size fields. If the **Use Unicode text fields** option is not enabled, only half of the field will be stored.

Also, to ensure that all user interface items are displayed correctly in languages that do not use ANSI code pages (non-Western languages), please refer to the Substituting fonts to display languages that do not use ANSI code pages in the Windows clients [page 66] paragraph.

Adding Unicode support to the Asset Manager DBMS using Oracle

To add Unicode support to the Asset Manager database when this is based on Oracle:

1. Create a new Oracle instance using the Oracle tools Database Configuration Assistant, and the following values:
2. Export the current database with the Oracle export tool (expdp).
3. Import the existing Asset Manager database with the Oracle import tool (impdp), to the Unicode instance created in step 1. Verify that data import has been successful.

   Warning:

   Export and Import can take several hours.

   The export and import process is done with Oracle tools, therefore, HP Software cannot provide tools support if there is any error during the process, especially during the import where the conversion is performed.

   ---

4. Open the file `C:\Documents and Settings\All Users\Application Data\HP\AssetManager\conf\amdb.ini`
5. At the section that describes the connection to the database, add the lines:

   ```
   TableSpace=MY_DATA_TAB_SPC
   TableSpaceIndex=MY_INDEX_TAB_SPC
   TableSpaceLong= MY_DATA_TAB_SPC
   TableSpaceArchive=MY_ARCHIVE_TAB_SPC
   ```

   where

   ```
   MY_DATA_TAB_SPC
   MY_INDEX_TAB_SPC
   MY_ARCHIVE_TAB_SPC
   ```

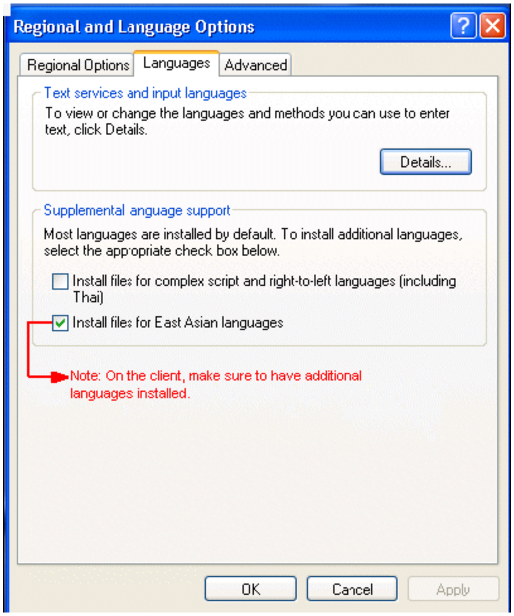
   are the table spaces that you have defined for your Data, Indexes and Archive if there are any.

   Warning:

   This part can take several hours.

   ---

6 Install additional languages on Asset Manager client computers using Oracle tools, as in the screenshot below:


Regional and Language Options
Regional Options
Languages
Advanced
Text services and input languages
Details...
Supplemental anguage support
OK
Cancel
Apply

To view or change the languages and methods you can use to enter text, click Details.

Most languages are installed by default. To install additional languages, select the app opriate check box below.

☐ Install files for complex script and right-to-left languages (including Thai)

☑ Install files for East Asian languages

Note: On the client, make sure to have additional languages installed.

7 Verify that the database is now correctly configured to support Unicode, by running the following queries with Oracle SQL Plus:

1.
   ```
   SELECT * FROM NLS_DATABASE_PARAMETERS;
   ```
   You should get the following results:

   ```
   NLS_CHARACTERSET
   AL32UTF8
   NLS_NCHAR_CHARACTERSET
   AL16UTF16
   ```

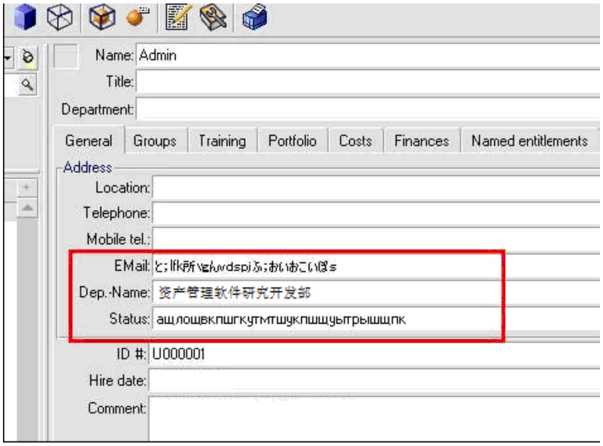
8 Perform a screen text dump on a source containing a Unicode string, for instance:

```
SELECT DUMP (email,1017) FROM amEmplDept;
```

Supposing this contains a Unicode string, you get results similar to:

```
DUMP(EMAIL,1017)
-----------------------------------------------------------------------
---------
NULL
Typ=1 Len=17 CharacterSet=UTF8: 89,e4,b8,d0,d1,8b,d1,89,e4,b8,d1,b8,d1,
89,e4,b8,b9
```

Assuming that Unicode is now correctly supported you should now be able to enter Unicode-based texts in Asset Manager as in the following example:



Name: Admin

Title:

Department:

Location:

Telephone:

Mobile tel.:

EMail: と;lfk所vgんvdspjふ;おいおこいぽs

Dep.-Name: 资产管理软件研究开发部

Status: ащлощвкпшгкутмтшукпшщуытрыщщпк

ID #: U000001

Hire date:

Comment:

Changing your DBMS

Asset Manager enables you to change the DBMS you use to host your Asset Manager 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 Asset Manager 3.x database to a Asset Manager 9.3 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 48],
- Creating a connection with Asset Manager [page 58],
- Creating the database structure with Asset Manager Application Designer [page 59].

Exporting the database structure and its data

To export:

1. Start Asset Manager Application Designer.
2. Open your Asset Manager database (**File/ Open.../ Open existing database** menu in Asset Manager Application Designer).
3. Select the **Action/Export database structure and data...** menu.
4. Indicate the name of your export and click **Save**.

   Asset Manager Application Designer 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 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 Asset Manager Application Designer.

   Asset Manager Application Designer imports the export files into your new database.

Attaching an SQL 2000 database to an SQL 2005 server

To migrate an SQL 2000 system to SQL 2005, you must do the following in order for the database to be 100% usable in Asset Manager:

1. Change the database compatibility level of the previous DBMS (SQL Server 2000). You need to do this because the database is marked as "SQL Server 2000" and is not compatible with the SQL 2005 mode:
    1. Start the SQL Server 2005 administration tool.
    2. Edit the database properties, then set the value of the **Compatibility level** option to **SQL Server 2005**.
2. Convert the SQL syntax of the database triggers:
    1. Start Asset Manager Application Designer.
    2. Select the **Action/ Diagnostics / Repair database...** menu item.
    3. Select the **Check triggers** task, select the **Repair** option then click **Run**.

Deleting an Asset Manager database

For security reasons, Asset Manager does not provide a command to delete a database. You must do this outside of Asset Manager.

Deleting an Asset Manager database requires several steps:

1. Destroying the database itself:

    Use the administration tool of the DBMS.

    - Example for Microsoft SQL Server 7.0: **Microsoft SQL Enterprise Manager**.
    - Example for Oracle: **Oracle User Manager**.

    In the case of Microsoft SQL Server:

    1. Make sure that you have started the Windows service of the SQL Server instance (for example **MSSQL$ASSETMANAGER**) on which the database to be deleted is installed.
    2. Open a DOS command prompt.
    3. Execute the following commands:
        a. Enter the command to start the SQL editor for the SQL Server instance you are using.

        Syntax:

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

        Example:

        ```
        osql -S (local)\ASSETMANAGER -U sa -P saampassword
        ```

b Execute the command:

```
go
```

c Enter the command to detach a database.

Syntax:

```
exec sp_detach_db '<Database name>', 'false'
```

Example:

```
exec sp_detach_db 'ACBase', 'false'
```

d Execute the command:

```
go
```

e Delete the `.mdf` and `.log` files related to the database.

2. Delete connections defined at the level of Asset Manager on each client machine.
3. With Microsoft SQL Server: Delete ODBC connections on each client machine.

5 Creating a database description file

Asset Manager Application Designer 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 Asset Manager database structure:

- The `database.txt` and `tables.txt` files: They contain the complete structure of the database. These files are located in the `doc\infos` sub-folder of the Asset Manager 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 Asset Manager Application Designer with a connection to your database.

- Asset Manager Application Designer program: It enables you to freely create description files of the Asset Manager database (tables, fields, links and indexes). It makes use of:
  - An Asset Manager database description file (`gbbase*.*` files) or a connection to an Asset Manager 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 Asset Manager, 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 Asset Manager 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 Asset Manager Application Designer 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 Asset Manager Application Designer 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 Asset Manager Application Designer 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 Asset Manager Application Designer 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 Asset Manager 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

Asset Manager Application Designer 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 Asset Manager Application Designer.

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 Asset Manager Application Designer, 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:\models\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 (<na
me of the first property> (ASC|DESC) [, <name of the next property> (ASC|D
ESC)])] [<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

Asset Manager Application Designer 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 5.1. 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

Property name	Description	Connection required
Connected	This field may have two values: - 1: Asset Manager Application Designer was executed in reference to a connection. - 0: Asset Manager Application Designer was executed in reference to a database description file.	No
Connection	Name of the Asset Manager connection used to access the database.	Yes
AppInfo	Information about Asset Manager.	No
AppVersion	Asset Manager version number.	No
AppBuild	Asset Manager build number.	No
AppLanguage	Asset Manager language.	No
DbbVersion	The version number of the database structure.	No

Table instance

Properties

Table 5.2. Table instance properties

Property name	Description	Connection required
Create	This property may have two values: - 1: The login has create access rights for the table. - 0: The login does not have create access rights for the table.	Yes
Delete	This property may have two values: - 1: The login has delete rights for the table. - 0: The login does not have delete rights for the table.	Yes
ComputeString	Table description string.	No
InternalName	Internal name. This information has no particular use.	No
Label	Label	No
Desc	Description	No
SqlName	SQL name	No
FieldCount	Total number of fields in the table.	No
LinkCount	Total number of links in the table.	No
IndexCount	Total number of indexes in the table.	No

Property name	Description	Connection required
IsFirst	Indicates whether the item is located at the start of the collection, taking into account the filter and the sort order: - 0: no - 1: yes	No
IsLast	Indicates whether the item is located at the end of the collection, taking into account the filter and the sort order: - 0: no - 1: yes	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): - 0: no - 1: yes	No
HasFeatureValueTable	Indicates whether this table has a table of feature values associated with it: - 0: no - 1: yes	No
IsFeatureValueTable	Indicates whether this table is a table of feature values: - 0: no - 1: yes	No
HasMemoField	Indicates whether this table has a **Long text**-type field. - 0: no - 1: yes	No

Objects

Table 5.3. 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

Object name	Description
Base as Database	Described database

Field instance

Properties

Table 5.4. Field instance properties

Property name	Description	Connection required
Update	This property may have two values: ▪ 1: The login has update rights for the field. ▪ 0: The login does not have update rights for the field.	Yes
Write	This property may have two values: ▪ 1: The login has create access rights for the field. ▪ 0: The login does not have create access rights for the field.	Yes
Read	This property may have two values: ▪ 1: The login has read rights for the field. ▪ 0: The login does not have read rights for the field.	Yes
UserType	By default, the data entry and display format is the format for the "Type" property. The "UserType" property enables you to specify the data entry and display format when it is verified.	No
Type	Storage format	No
UserTypeFormat	Additional information about the "UserType" parameter.	No
Size	Maximum size of the values in the field, 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. This property may have two values: ▪ 1: The field can never be modified by users. ▪ 0: The field may be modified by a user, if the user has the required access rights.	No

Property name	Description	Connection required
Historized	This property may have two values: ■ 1: History is recorded for the field. ■ 0: History is not recorded for the field.	No
ForeignKey	This property may have two values: ■ 1: The field is a foreign key. ■ 0: The field is not a foreign key.	No
PrimaryKey	This property may have two values: ■ 1: The field is a primary key. ■ 0: The field is not a primary key.	No
InternalName	Internal name 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
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): ■ 0: no ■ 1: yes	No
EmptyOnDup	Indicates whether the default value is re-applied in case of duplication. ■ 0: no ■ 1: yes	No
FieldCase	Indicates the relative case behavior for the field. ■ 0: No modification to the field value. ■ 1: Value modified in upper-case characters. ■ 2: Value modified in lower-case characters. ■ 3: Value modified in smartcase.	No

Property name	Description	Connection required
Positive	Indicates whether the numeric-type field is positive: ■ 0: no ■ 1: yes	No

Objects

Table 5.5. Field instance objects

Object name	Description
Base as Database	Described database
Table as Table	Table containing the field.
MandatoryScript as Script	Mandatory calculation script for the field.
DefaultScript as Script	Default script for calculating the field's value.
ReadOnlyScript as Script	Read-only calculation script for the field.
HistoryScript as Script	History calculation script for the field.
RelevantScript as Script	Relevance calculation script for the field.

Possible values for the "Type" property

Table 5.6. 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	Used to store images, or other binary files, 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 5.7. 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
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 5.8. Link instance properties

Property name	Description	Connection required
Update	This property may have two values: - 1: The login has update rights for the link. - 0: The login does not have update rights for the link.	Yes
Write	This property may have two values: - 1: The login has create access rights for the link. - 0: The login does not have create access rights for the link.	Yes
Read	This property may have two values: - 1: The login has read rights for the link. - 0: The login does not have read rights for the link.	Yes
Type	Type of the link.	No
UserType	Type of information managed by the link.	No
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. - 1: The target table is not pre-defined. - 0: The target table is pre-defined.	No
Historized	This property may have two values: - 1: History is recorded for the link. - 0: History is not recorded for the link.	No
Cardinality	Cardinality of the link.	No
InternalName	Internal name This information has no particular use.	No
Label	Label.	No
Desc	Description	No
SqlName	SQL name.	No
System	Indicates whether this object is a system object (cannot be modified): - 0: no - 1: yes	No
LongHelpComment	Comments on the use of the link.	No

Property name	Description	Connection required
LongHelpSample	Examples of values that may be assigned to the link.	No
LongHelpWarning	Important information for the link.	No
LongHelpDesc	Description of the link.	No
LongHelpCommentNoHTMLTag	Remarks about the utility of the link (HTML tags excluded).	No
LongHelpSampleNoHTMLTag	Examples of values that the link can accept (HTML tags excluded).	No
LongHelpWarningNoHTMLTag	Important information about the link (HTML tags excluded).	No
LongHelpDescNoHTMLTag	Description of the link (HTML tags excluded).	No

Objects

Table 5.9. Link instance objects

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

Possible values for the "Type" property

Table 5.10. 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 5.11. Possible values for the "UserType" property

Stored value	Literal value	Equivalent displayed by Asset Manager Automated Process Manager
0	Normal	Normal
1	Comment	Comment
2	Image	Image
3	History	History
4	Feature values	Feature value

Index instance

Properties

Table 5.12. Index instance properties

Property name	Description
Duplicates	Indicates whether or not the index can have more than one non-NULL value. ■ 1: You can create several records where the group of fields in the index can have exactly the same value. ■ 0: You cannot create more than one record where the group of fields in the index has a given value.
NullValues	This property is only significant if the "Duplicates" property is set to "No". 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".) ■ 1: You can create several records having an index of NULL. ■ 0: You can only create one record having an index of NULL.
InternalName	Internal name This information has no particular use.
Label	Label as it appears in the detail screens.
Desc	Description
SqlName	SQL name
System	Indicates whether this object is a system object (cannot be modified): ■ 0: no ■ 1: yes

Objects

Table 5.13. Index instance objects

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

Script instance

Properties

Table 5.14. 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: - 0: no - 1: yes - 2: script
ScriptType	Type of information managed by the script. This property can have one of the following values: - 1: Data entry for a field is mandatory. - 2: By default, displays a feature in a table. - 3: Available character for a feature in a table. - 4: Records history for a field. - 5: Default value of a field. - 6: **For inheritance purposes** (bForInheritance) field of a feature in a table.
Source	Script that calculates the value of a field as it is displayed in the interface.
	Script that calculates the value of a field as it is stored in the database.

Property name	Description
VbReturnType	Type of string calculated by the script: - Integer: Integer from -32,768 to +32,767 - Long: Integer from -2,147,483,647 to +2,147,483,646 - Double: 8-byte floating-point number - String: Text for which all characters are accepted - Date: Date (not time).
ReferencedStorageFieldCount	Number of fields referenced in the script.

Objects

Table 5.15. 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 5.16. Properties of the FeatDesc instance

Property name	Description
System	Indicates whether this object is a system object (cannot be modified): - 0: no - 1: yes
Label	Label of the feature.
Desc	Description
SQLName	SQL name
Unit	Unit of the feature.
Type	Entry type of the feature.

Property name	Description
	Itemized list used to populate the value of the feature. This property is only useful if the feature's **Type** is **Itemized list**.
MinValue	The minimum value accepted by the feature (numeric **type** only).
MaxValue	The maximum value accepted by the feature (numeric **type** only).
IsConsolidated	Indicates whether the value is calculated by consolidation. ■ 0: no ■ 1: yes
HasClass	Indicates whether the feature is attached to a feature class: ■ 0: no ■ 1: yes

Objects

Table 5.17. 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 5.18. 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).

Property name	Description
LongHelpSampleNoHTMLTag	Examples of values that the field can accept (HTML tags excluded).
LongHelpWarningNoHTMLTag	Important information about the field (HTML tags excluded).
LongHelpDescNoHTMLTag	Description of the field (HTML tags excluded).
LinkFilter	The AQL condition of the filter (link type feature).
IsInherited	Indicates whether the feature is inherited: ■ 0: no ■ 1: yes
CreationHistorized	History line created when the main record is created.
Write	This property can take the values: ■ 0: The login has write rights for the feature. ■ 1: The login does not have write rights.
Read	This property can take the values: ■ 0: The login has read rights for the feature. ■ 1: The login does not have read rights.
Update	This property can take the values: ■ 0: The login has update rights for the feature. ■ 1: The login does not have update rights.

Objects

Table 5.19. 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 5.20. 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: - 0: no - 1: yes

Objects

Table 5.21. 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 5.22. Properties of the CalcFieldDesc instance

Property name	Description
Label	Label of the calculated field.
Desc	Description
SQLName	SQL name
	Formula of the calculated field.

Property name	Description
UserType	Result type returned by the calculated field.
Type	Type of calculated field.
IsVisible	Indicates whether the calculated field is visible by default: ■ 0: no ■ 1: yes

Objects

Table 5.23. 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 5.24. Properties of the View instance

Property name	Description
InternalName	Internal name This information has no particular use.
System	Indicates whether this object is a system object (cannot be modified): ■ 0: no ■ 1: yes
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): ■ 0: no ■ 1: yes
HasNotebook	Indicates whether the screen has tabs: ■ 0: no ■ 1: yes

Property name	Description
ScreenFilter	System filter of the screen.
CaptionList	The caption of the screen's list.
CaptionDetail	Title of the screen.

Objects

Table 5.25. 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 5.26. Properties of the Page instance

Property name	Description
InternalName	Internal name This information has no particular use.
System	Indicates whether this object is a system object (cannot be modified): ▪ 0: no ▪ 1: yes
Label	The label of the page.
	Description
SQLName	SQL name

Instance PageItem

Properties

Table 5.27. Properties of the PageItem instance

Property name	Description
IsVerticalSplit	Indicates whether this object is a vertical separator for the page: ■ 0: no ■ 1: yes
IsHorizontalSplit	Indicates whether this object is a horizontal separator for the page: ■ 0: no ■ 1: yes
FrameLabel	The label of the frame.
SQLName	SQL name.
IsField	Indicates whether this object is a field on the page: ■ 0: no ■ 1: yes
IsLink	Indicates whether this object is a link: ■ 0: no ■ 1: yes
IsFeature	Indicates whether this object is a feature: ■ 0: no ■ 1: yes
IsCalcField	Indicates whether this object is a calculated field: ■ 0: no ■ 1: yes
BaseType	If the object is a field, this relates to the type of field.

Objects

Table 5.28. 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.

Object name	Description
CalcField as CalcFieldDesc	The calculated field in this page.

Instance SysEnumValue

Properties

Table 5.29. Properties of the SysEnumValue instance

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

Instance String

Properties

Table 5.30. Properties of the String instance

Property name	Description
Value	The value of the character string.

Instance ScriptField

Properties

Table 5.31. Properties of the ScriptField instance

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

Global variables

Table 5.32. Global variables

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

6 Diagnostics and repairing a database

The **Action/ Diagnostics/ Repair database** menu enables you to test the integrity of an existing Asset Manager 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.

Table to analyze field: Select **(All tables)** to analyze or repair the entire database.

Log file field: This field enables you to save a log file containing the diagnostics of your database.

You have two options for this:

- **Analyze only**: Asset Manager Application Designer performs a basic diagnostic test on the database, but does not perform any repairs if any problems are encountered.
- **Repair**: Asset Manager Application Designer performs a diagnostic of the database and makes any necessary repairs.

When you click **Run**, Asset Manager Application Designer 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.

- ℹ 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.

Warning:

The **Check validity of records** option may take a lot of time (more than 1 day) for tables that have a **Validity** script and contain many records. For such tables, the validity script will be executed for each record in the table.

Asset Manager provides a non-GUI approach for you to check the tables selectively; thereby, skip such kind of tables that you might don't need to diagnose. You can see the **Non-GUI diagnostics approach** section for the usage details.

In some instances, the check may never get through.

Non-GUI diagnostics approach

While **Check validity of records** option forces you to check the validity of the entire database and might take a lot of time, Asset Manager provides a non-GUI diagnostics approach so that you can select only the wanted tables.

You can check the validity of a table using the following command:

```
amdbal -dbcheck:<Database name>;;ValidityScript;Table="<SQL name of the ta
ble>"
```

For example,

```
amdbal -dbcheck:AMDemo93en;;ValidityScript;Table="amComputer"
```

If several tables are to be checked, run the following scripts in batch:

```

amdbal -dbcheck:AMDemo93en;;ValidityScript;Table="amAsset"
amdbal -dbcheck:AMDemo93en;;ValidityScript;Table="amPortfolioItem"
amdbal -dbcheck:AMDemo93en;;ValidityScript;Table="amComputer"
amdbal -dbcheck:AMDemo93en;;ValidityScript;Table="amSoftInstall"
```

7 Accessing the database via ODBC

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

Note:

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

Overview of access to the Asset Manager database

Installing the ODBC driver

The Asset Manager installation program will install the ODBC driver if you:

- Specifically select this package during setup.
- Install other packages which require the driver.

The ODBC driver is called HP Asset Manager 9.30 <language> Driver. **Adbc.dll** is copied to the Windows "system32" folder.

Note:

The ODBC driver is independent of 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 SAP Crystal Reports.

Note:

You are not required to use this driver. You can access the Asset Manager database directly if your reporting program supports the relevant DBMS directly.

Advantages of using the ODBC driver:

Table 7.1. Advantages of using the ODBC driver

	Using the ODBC driver	Without the ODBC driver
Security when accessing the Asset Manager database	When using reports, an Asset Manager login and password are required to access the database. User profiles associated with the login are respected.	When using reports, the database access parameters required by the DBMS are requested. These are not linked with the Asset Manager user profiles.
DBMS connection parameters	You don't need to know the connection parameters required by the DBMS in order to access the database.	You need to know the connection parameters required by the DBMS in order to access the database.
Selecting the connection used to access the Asset Manager database.	The user of the report selects the appropriate connection.	The user directly accesses the database without using the Asset Manager connections.
Link between the DBMS engine and the report	The report is independent of the database engine used. When you change DBMS, you don't have to modify your reports.	The report depends on the database engine. When you change DBMS, you have to modify your reports.

Data items accessible via the ODBC driver

The ODBC driver enables you to view:

- Tables
- Standard fields
- Calculated fields
- 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 Asset Manager Databases. This connection cannot be modified or removed.

You can use two types of ODBC connections to access the Asset Manager database:

- The standard Asset Manager Databases connection.
- A connection you have created yourself.

Usefulness of the standard Asset Manager 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 Asset Manager connection when you create and use the report. The standard connection dialog box in Asset Manager is used for this.

How to create your own ODBC connections

1. Launch ODBC Administrator.
2. Create a new connection by selecting the HP Software Asset Manager Driver.
3. Proceed as usual to create the ODBC connection.

Example: Creating a report in SAP Crystal Reports with the ODBC driver

- Launch SAP Crystal Reports (Designer).
- Open a new report.
- Indicate that the report concerns "SQL/ODBC" data.
- Select the Asset Manager Databases ODBC connection.
- The standard Asset Manager connection dialog box is displayed.
- Select the appropriate Asset Manager connection. Enter the login to be used to create the report and the appropriate password.
- Create the report in the usual fashion.

8 Accessing the database using Citrix XenApp

This chapter outlines the use of Citrix XenApp as a means to provide access to Asset Manager as a hosted application.

Overview

Citrix XenApp is an application virtualization / application delivery product that allows users to connect to corporate applications hosted on a remote server.

XenApp can either host applications on central servers and allow users to interact with them remotely, or stream and deliver them to user devices for local execution.

Citrix XenApp can be used in either of the following configurations:

- **Remote Desktop mode**:
  - Asset Manager is installed on a Citrix XenApp server.
  - An administrator grants users the right to access the Citrix XenApp server.
  - Users access Asset Manager Windows client via a Citrix XenApp remote desktop session using Remote Desktop Protocol (RDP).
- **Shared application mode**:
  - Asset Manager is installed on a Citrix XenApp server.
  - An administrator grants users the right to access Asset Manager on the Citrix XenApp server.

- Users access the Asset Manager Windows client via a Citrix XenApp client or via a URL.

Installing Citrix XenApp

Refer to http://support.citrix.com/productdocs/ for details of how to install and configure Citrix XenApp, in particular the **Getting Started with Citrix XenApp** guide.

Known Limitations

- Asset Manager modal windows cannot be moved or resized.
- When a user creates an Asset Manager connection to the database (via a remote desktop session or a Citrix XenApp client), the connection information is stored in a virtualisation folder on the Citrix XenApp server, rather than in the correct `amdb.ini` file.

  Workaround: administrators should create Asset Manager connections directly on the Citrix XenApp server using the Asset Manager Windows client.
- Users cannot create ODBC connections to the Asset Manager database using the Asset Manager Windows client.

  Workaround: administrators should create ODBC connections to the Asset Manager database directly on the Citrix XenApp server.

9 Archival

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:
>
> For DB2, 8K tablespaces must be used.
>
> ▸ DB2 UDB [page 54]

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 Asset Manager Application Designer 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, amF
eatMemo, amFeatParam, amFeatScript, amFeature, amFeatureClass, amFuncRi
ght, amHelp, amHelpMemo, amImage, amLoginSlot, amOption, amProfile, amS
ysConfig, amUserRight, amViewDef, amFuncDomain
```

In Asset Manager Application Designer, 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.

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 overwrites the last backup.
- 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 Asset Manager 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> WHERE <Clause WHERE>)`	Standard tables.
Backup	`BACKUP FROM <Clause FROM> WHERE <Clause WHERE>)`	Standard tables.
Purge	`PURGE FROM <Clause FROM> WHERE <Clause WHERE>)`	Standard and archival tables.
Restore	`RESTORE FROM <Clause FROM> 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**.

Links pertaining to archived records

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

Examples

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 **Requests** (amRequest) table as the **Context**. The action type 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 < AddDay
s(getdate(), -90) AND seStatus = 41")
```

Note:

The **amDbExecAql** function queries the database. For more information on this function, refere 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 **Asset lifecycle/Procurement lifecycle/ Requests/ 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 🔍.

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	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. For example, an asset and all its sub-assets are archived, copied, restored or purged simultaneously.
DEFINE or NORMAL	Foreign keys in linked records are reset to zero. Archival fields (arl<Name>Id and arstr<Name>) are respectively populated with the identifier of the archived record and with its description string. 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.

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 not 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.

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

Warning:

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

10 Denormalization

The Asset Manager 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 Asset Manager

In theory, there are several denormalization techniques available.
Asset Manager 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 and departments** 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. Asset Manager 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. Asset Manager detects automatically the presence of a denormalized value and simplifies queries on the fly.

Note:

Fields created through denormalization are visible in Asset Manager Application Designer 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 Asset Manager Application Designer to carry out the operation for real.

1. In Asset Manager Application Designer, 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 Asset Manager Application Designer, 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. Asset Manager Application Designer displays a wizard to save the database structure.

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 and departments** tables.

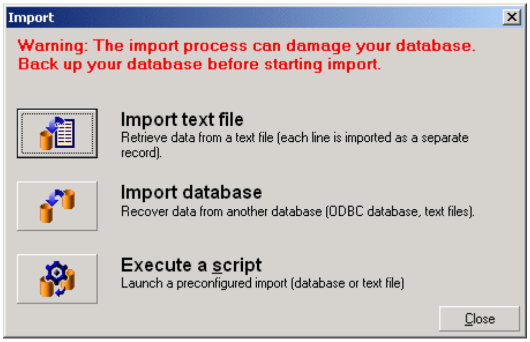
11 Import

This chapter explains how to import data with Asset Manager.

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 11.1. Import - Selecting the type



Import
Warning: The import process can damage your database.
Back up your database before starting import.
Import text file
Retrieve data from a text file (each line is imported as a separate record).
Import database
Recover data from another database (ODBC database, text files).
Execute a script
Launch a preconfigured import (database or text file)
Close

Overview of importing data

An Asset Manager administrator can import data into the Asset Manager 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 Asset Manager 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 Asset Manager.

Importing data from multiple text files

Each text file is mapped to a table in the Asset Manager 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 Asset Manager records. You cannot delete Asset Manager 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 Asset Manager 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 Asset Manager 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 Asset Manager database

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

Avoid concurrent access to the Asset Manager database

We advise against performing imports concurrently on different machines or using Asset Manager on another machine during an import.

Constraints to be respected when importing data into a field

When importing data into the Asset Manager 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	The source field must be a number. If the source field originates from a structured database and the field type is **Number** or **Monetary** then there is no constraint. If the source field is a **Text** type field, you must present a number, use the period "." as a decimal separator and do not use a separator for thousands.
Boolean (Yes/No)	The source field must display "1" (for "Yes"), or "0" (for "No").

If the value is:	Then:
Date	If the source field originates from a structured database and the field type is **Date** or **Date and time**, then there is no constraint. If the source field is a **Text** type field, you must respect the following constraints: ▪ The date format (order of year, month and day) must be the same for all records. Specify this format when importing. ▪ You must systematically enter a day, month and year. ▪ Use the same separator (of your choice) to separate days, months and years for all records. Specify this separator when importing.
System itemized list	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. **Example** If the itemized list is "Yes|1|No|0", importing "Yes" or "1" will give the same results. If you present an empty value, the import module attributes the value "0" to the field. We recommend presenting the stored numeric values, since these are more stable than their text equivalents from one version of Asset Manager to another and independent of the language version used.
Custom itemized list	The value of the field is selected via a "drop-down" itemized list, which can be modified by the user. Users can select one of the existing values of the itemized list. If, in addition, the itemized list is "open", users with adequate rights can enter a new value. This value will be added to the itemized list.
Percentage	Imported values must be the percentage value with or without the percentage sign "%" (for example: "10" or "10%").
Duration	You must respect the constraints defined by the **UserType** and **UserTypeFormat** properties when the **UserType** is **Duration**.
Table or field SQL name	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.

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	If the source field originates from a structured database and the field type is **Date and time** or **Date**, then there is no constraint. 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**.

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 Asset Manager 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. Click the **Import database** option, and then the **Text** tab.

To import an ODBC database: Use the **File/ Import** menu. Click the **Import database** option, and then 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.
- The fields can have the following types:
    - Numeric
    - Character string
    - Date
    - Date and Time
- Imported values must follow the constraints related to the Asset Manager 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 Asset Manager 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. Asset Manager 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 database** 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, Asset Manager 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, Asset Manager 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.

Asset Manager 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.

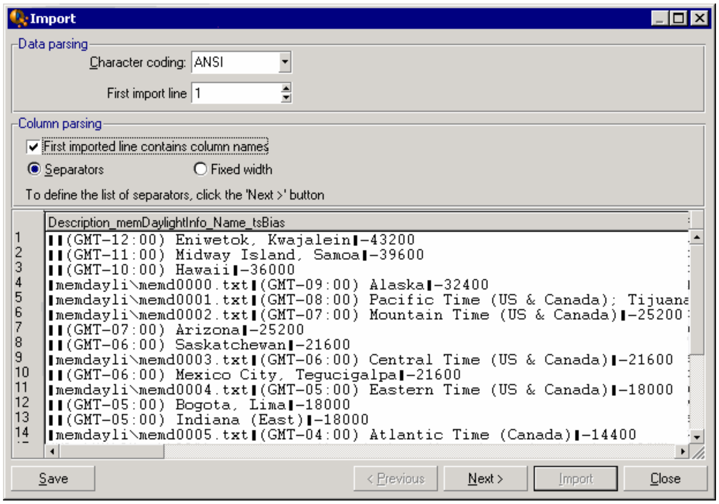
Fixed width

Check this check box if all the values of each field are the same length.

Asset Manager 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.



Data parsing

Character coding: ANSI

First import line: 1

Column parsing

☑ First imported line contains column names

● Separators ○ Fixed width

To define the list of separators, click the 'Next >' button

```
Description_memDaylightInfo_Name_tsBias
▮▮(GMT-12:00) Eniwetok, Kwajalein▮-43200
▮▮(GMT-11:00) Midway Island, Samoa▮-39600
▮▮(GMT-10:00) Hawaii▮-36000
▮memdayli\memd0000.txt▮(GMT-09:00) Alaska▮-32400
▮memdayli\memd0001.txt▮(GMT-08:00) Pacific Time (US & Canada); Tijuana
▮memdayli\memd0002.txt▮(GMT-07:00) Mountain Time (US & Canada)▮-25200
▮▮(GMT-07:00) Arizona▮-25200
▮▮(GMT-06:00) Saskatchewan▮-21600
▮memdayli\memd0003.txt▮(GMT-06:00) Central Time (US & Canada)▮-21600
▮▮(GMT-06:00) Mexico City, Tegucigalpa▮-21600
▮memdayli\memd0004.txt▮(GMT-05:00) Eastern Time (US & Canada)▮-18000
▮▮(GMT-05:00) Bogota, Lima▮-18000
▮▮(GMT-05:00) Indiana (East)▮-18000
▮memdayli\memd0005.txt▮(GMT-04:00) Atlantic Time (Canada)▮-14400
```

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.

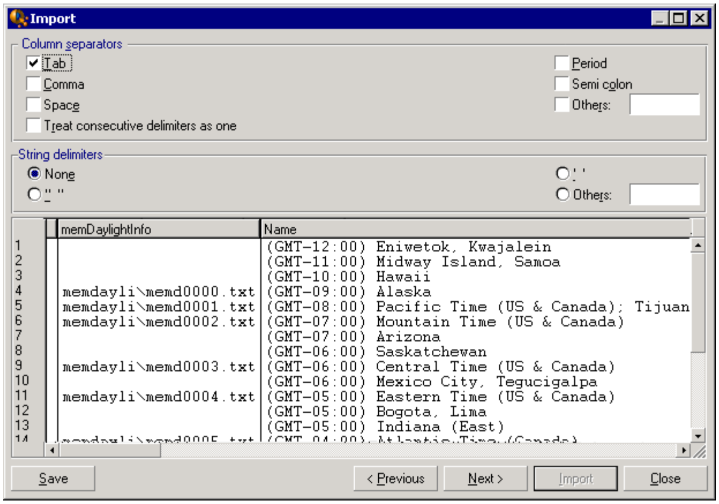
If you check the **Treat consecutive delimiters as one** option, Asset Manager will treat identical consecutive delimiters as one and will not create empty columns. If you want to created an empty column, uncheck this box use two consecutive delimiters in your text file.

String delimiters

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

If, between two delimiters, Asset Manager 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.



Column separators
☑ Tab
☐ Comma
☐ Space
☐ Treat consecutive delimiters as one
☐ Period
☐ Semi colon
☐ Others:

String delimiters
● None
○ " "
○ ' '
○ Others:

	memDaylightInfo	Name
1		(GMT-12:00) Eniwetok, Kwajalein
2		(GMT-11:00) Midway Island, Samoa
3		(GMT-10:00) Hawaii
4	memdayli\memd0000.txt	(GMT-09:00) Alaska
5	memdayli\memd0001.txt	(GMT-08:00) Pacific Time (US & Canada); Tijuan
6	memdayli\memd0002.txt	(GMT-07:00) Mountain Time (US & Canada)
7		(GMT-07:00) Arizona
8		(GMT-06:00) Saskatchewan
9	memdayli\memd0003.txt	(GMT-06:00) Central Time (US & Canada)
10		(GMT-06:00) Mexico City, Tegucigalpa
11	memdayli\memd0004.txt	(GMT-05:00) Eastern Time (US & Canada)
12		(GMT-05:00) Bogota, Lima
13		(GMT-05:00) Indiana (East)

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

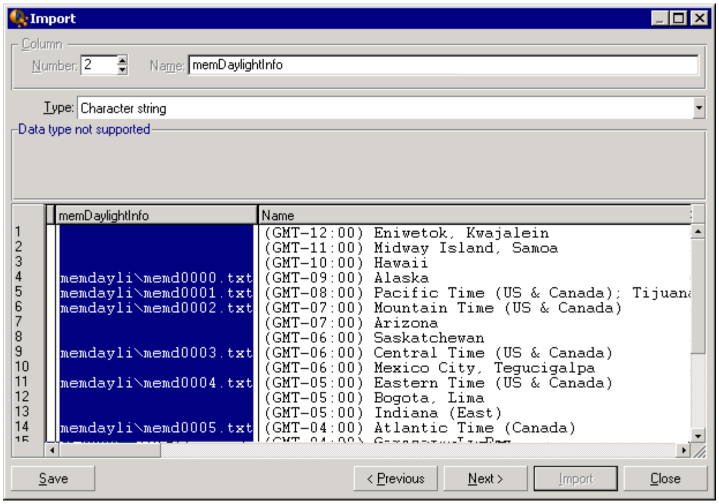
Enter the field type to be imported. Asset Manager lets you choose from the following possibilities:

- Number: All characters must be numbers. If Asset Manager 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 Asset Manager finds other formats when importing, the field's value is set to null.

Date format

If you indicate that the field contains a "Date", Asset Manager 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 Asset Manager.



Number: 2

Name: memDaylightInfo

Type: Character string

	memDaylightInfo	Name
1		(GMT-12:00) Eniwetok, Kwajalein
2		(GMT-11:00) Midway Island, Samoa
3		(GMT-10:00) Hawaii
4	memdayli\memd0000.txt	(GMT-09:00) Alaska
5	memdayli\memd0001.txt	(GMT-08:00) Pacific Time (US & Canada); Tijuana
6	memdayli\memd0002.txt	(GMT-07:00) Mountain Time (US & Canada)
7		(GMT-07:00) Arizona
8		(GMT-06:00) Saskatchewan
9	memdayli\memd0003.txt	(GMT-06:00) Central Time (US & Canada)
10		(GMT-06:00) Mexico City, Tegucigalpa
11	memdayli\memd0004.txt	(GMT-05:00) Eastern Time (US & Canada)
12		(GMT-05:00) Bogota, Lima
13		(GMT-05:00) Indiana (East)
14	memdayli\memd0005.txt	(GMT-04:00) Atlantic Time (Canada)

Note:

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

Step 4: Map source fields to target fields in the Asset Manager 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 Asset Manager database fields.

1. Start by mapping the text file to a table in the Asset Manager database using the "Target table" field.
2. Then map the fields in the text file to be imported (displayed in the "Source fields" table) to a field in the Asset Manager 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 a field in the Asset Manager 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 source table to a target table.
3. Next, for each source table/target table pair in the table on the right, map the fields from the source table to a field in the Asset Manager database: 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: Asset Manager 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 Asset Manager database structure

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

Step 6: Map the fields to be imported to fields in the Asset Manager 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 Asset Manager database).

Target table

When importing a single text file

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

When importing a set of text files or an ODBC database

Asset Manager 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

Asset Manager 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 🟥.

Asset Manager 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, Asset Manager 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, Asset Manager 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

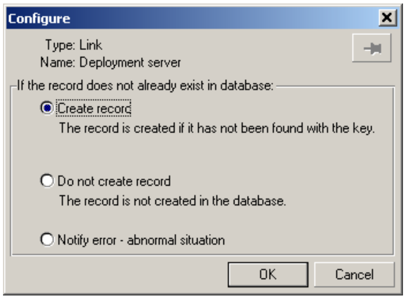
Let's say you want to import a file containing data that will be distributed to several different tables. This might be a file containing employee records and their corresponding portfolio-item records, for example. You would first select a main target table: the **Employees and departments** (amEmplDept)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

Asset Manager 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.



Configure

Type: Link

Name: Deployment server

If the record does not already exist in database:

● Create record

The record is created if it has not been found with the key.

○ Do not create record

The record is not created in the database.

○ Notify error - abnormal situation

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

Asset Manager 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).

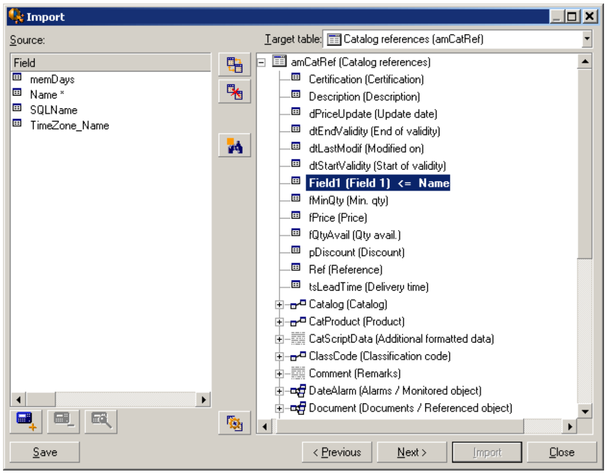
In the case of "Own" type links, the pushpin is automatically placed and cannot be removed. "Own" type links are links for which the linked records are automatically deleted if the main record is deleted. The employees-training link is an example of this type of link: If you delete an employee, you will 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 Employees and departments 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:

Asset Manager 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.


Target table: Catalog references (amCatRef)

Field1 (Field 1) <= Name

Import
Source:
Target table:
Field
memDays
Name *
SQLName
TimeZone_Name
amCatRef (Catalog references)
Certification (Certification)
Description (Description)
dPriceUpdate (Update date)
dtEndValidity (End of validity)
dtLastModif (Modified on)
dtStartValidity (Start of validity)
fMinQty (Min. qty)
fPrice (Price)
fQtyAvail (Qty avail.)
pDiscount (Discount)
Ref (Reference)
tsLeadTime (Delivery time)
Catalog (Catalog)
CatProduct (Product)
CatScriptData (Additional formatted data)
ClassCode (Classification code)
Comment (Remarks)
DateAlarm (Alarms / Monitored object)
Document (Documents / Referenced object)
Save
< Previous
Next >
Import
Close

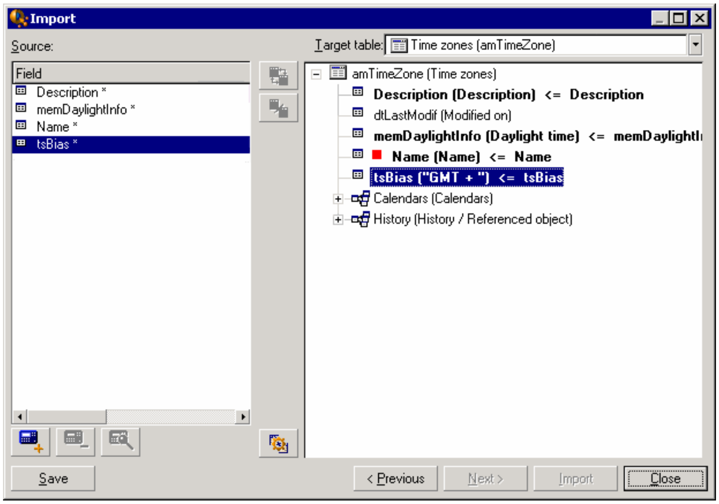
Symbols used in the tree structure of target tables

▤ 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.



Target table: Time zones (amTimeZone)

Field
- Description *
- memDaylightInfo *
- Name *
- tsBias *

amTimeZone (Time zones)
- **Description (Description) <= Description**
- dtLastModif (Modified on)
- **memDaylightInfo (Daylight time) <= memDaylightI**
- **Name (Name) <= Name**
- **tsBias ("GMT + ") <= tsBias**
- Calendars (Calendars)
- History (History / Referenced object)

Step 7: Add additional calculated fields in the source file

Asset Manager 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**. Asset Manager will skip these characters.
3. Enter the **Number of characters to use**: Asset Manager will keep this number of characters after having skipped the "Characters to ignore".
4. The **Start extraction from end of field** check box tells Asset Manager 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**. Asset Manager keeps all the data that follows these separators.
4. Enter the **Number of separators to include**. Asset Manager 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 Asset Manager 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.

Asset Manager divides the source field into sub-values. The number of sub-values equals the number of character strings separated by separation characters. Asset Manager 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 Asset Manager 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 **Employees and departments** (amEmplDept) table, 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.

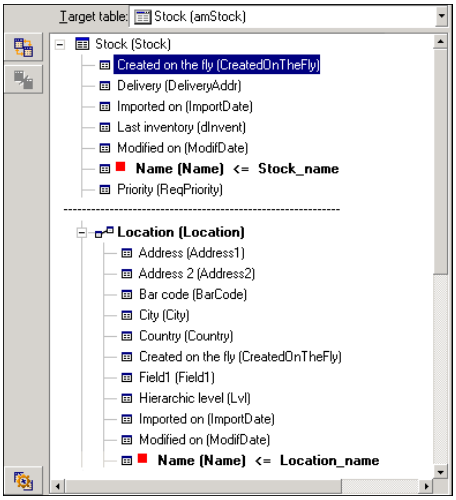
Importing documents

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 Asset Manager interprets your choice of keys:

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



Target table: Stock (amStock)

Name (Name) <= Stock_name

Name (Name) <= Location_name

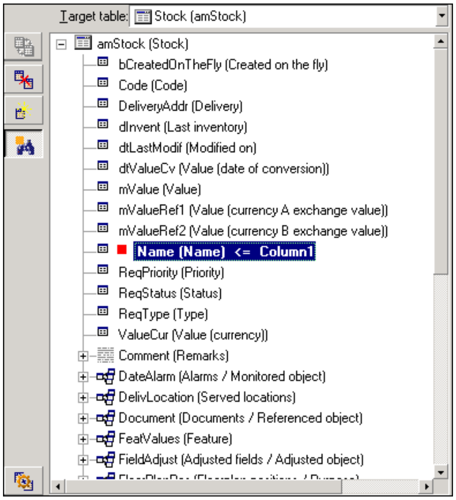
In this example, a stock is identified by two main keys:

- 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



Target table: Stock (amStock)

- amStock (Stock)
  - bCreatedOnTheFly (Created on the fly)
  - Code (Code)
  - DeliveryAddr (Delivery)
  - dInvent (Last inventory)
  - dtLastModif (Modified on)
  - dtValueCv (Value (date of conversion))
  - mValue (Value)
  - mValueRef1 (Value (currency A exchange value))
  - mValueRef2 (Value (currency B exchange value))
  - 🟥 **Name (Name) <= Column1**
  - ReqPriority (Priority)
  - ReqStatus (Status)
  - ReqType (Type)
  - ValueCur (Value (currency))
  - Comment (Remarks)
  - DateAlarm (Alarms / Monitored object)
  - DelivLocation (Served locations)
  - Document (Documents / Referenced object)
  - FeatValues (Feature)
  - FieldAdjust (Adjusted fields / Adjusted object)

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.

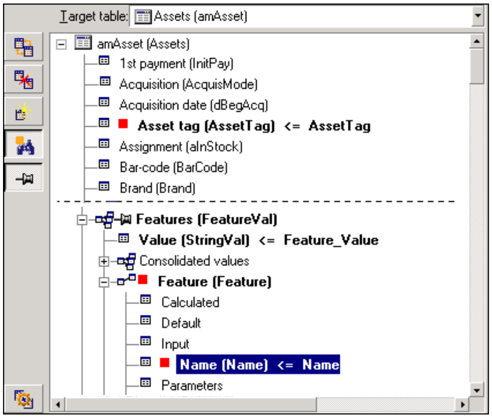
- To declare a field in a linked table as one of the keys in the main table, you must declare the field in the linked table AND the link as identification keys. If you only check the link, the key will only be used as key for the linked table itself.

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, Asset Manager 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.

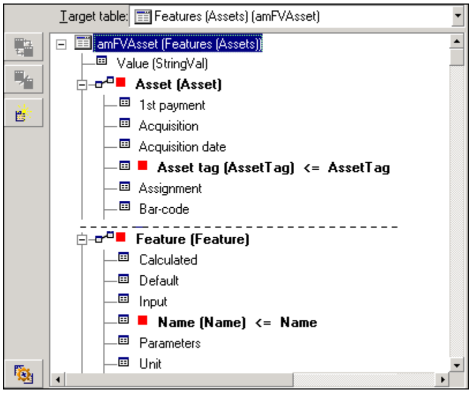
2 The icon on the Features (FeatureVal) * link specifies that you only want to find the features of this asset.

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):



Target table: Features (Assets) (amFVAsset)

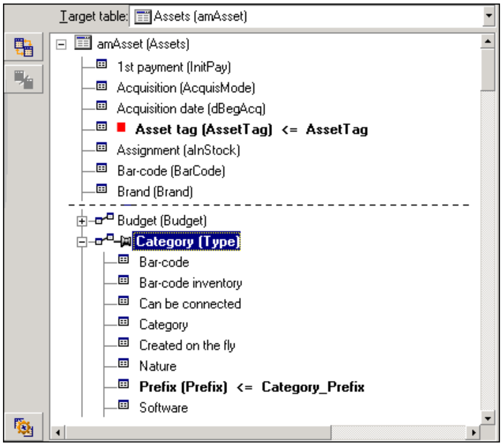
amFVAsset (Features (Assets))
- Value (StringVal)
- ■ Asset (Asset)
  - 1st payment
  - Acquisition
  - Acquisition date
  - ■ **Asset tag (AssetTag) <= AssetTag**
  - Assignment
  - Bar-code
- ■ Feature (Feature)
  - Calculated
  - Default
  - Input
  - ■ **Name (Name) <= Name**
  - Parameters
  - Unit

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.



Target table: Assets (amAsset)

amAsset (Assets)
- 1st payment (InitPay)
- Acquisition (AcquisMode)
- Acquisition date (dBegAcq)
- ■ **Asset tag (AssetTag) <= AssetTag**
- Assignment (alnStock)
- Bar-code (BarCode)
- Brand (Brand)
- Budget (Budget)
- **Category (Type)**
  - Bar-code
  - Bar-code inventory
  - Can be connected
  - Category
  - Created on the fly
  - Nature
  - **Prefix (Prefix) <= Category_Prefix**
  - Software

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. Asset Manager 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

- Asset Manager imports data line by line in the order the data appears in the source file.
- Asset Manager can import data into several different tables from a single line.
- If part of an imported line cannot be imported, Asset Manager creates what it can create.
- Asset Manager searches the Asset Manager database for a record whose identification keys have exactly the same values as the source record. If such a record is found, Asset Manager modifies it according to the information contained in the text file.
- If no records are found that match the identifier keys, Asset Manager 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, Asset Manager 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, Asset Manager inserts an empty field instead of the default value.

Saving and executing an import script

An import 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 and departments** table based on a file from the Human Resources department, for example.)

Asset Manager 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. Asset Manager 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 (via a batch file) using the import application **amimpl.exe**, which is located in the **bin** folder of Asset Manager'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.
- Asset Manager database connection name to import completely (as stated in the **File/ Manage connections** menu, **Name** field).
- The name of an Asset Manager 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
EngineName	Oracle	ODBC
DatabaseLocation	Server name	Data source name
User	Account name	MS SQL Server user name
Password	Account password	MS SQL Server user password

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

-dst: Asset Manager 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 Asset Manager account that will receive the imported data ("Admin" account or Asset Manager 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 -dstl
ogin:Gerald -dstpass:Password -log:'My Log File.txt'
```

12 Controlling access to the database

This chapter explains how to manage user access to Asset Manager databases.

Note:

Access rules can only be managed by database administrators.

Importance and overview of database access rights

Asset Manager is a program that can be used concurrently by several users: It uses a "shared" database.

Managing the database access rights in Asset Manager 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 Asset Manager License Keys installed with HP AutoPass License Management Tool.

Defining access conditions for each user

All users may not need to perform the same tasks and make the same changes in the database. This depends on their role in the company and the company's procedures; employee A may need to create assets, employee B may need to access the stocks and employee C may need to view work order slips, and so forth.

In order to allow an employee to access Asset Manager:

- The employee must be included in the table of departments and employees.
- The employee needs a "login" to connect to Asset Manager.
- The administrator must assign "Authorized user roles" and "role used at connection" to the employee, or give the employee administration rights.

Data security and confidentiality

To ensure data security and to make sure that your information is not wrongfully viewed, modified or destroyed, an Asset Manager administrator can manage three aspects of access security:

- Define user access rights for the network.
- Define Asset Manager user roles and profiles.
- Define how and when backups of the database are made.

Definitions in access management

This section defines the concepts related to access management.

- Definition of a user role [page 166]
- Definition of a user profile [page 167]
- Definition of a user right [page 167]
- Definition of an access restriction [page 167]
- Definition of a functional right [page 168]

A user role can be associated with several user profiles. Also, a user profile can be part of several user roles.

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.

Definition of a user role

A user role represents the role that a user takes on in Asset Manager. For example, you can have:

- An "Inventory Manager" for an employee who needs to manage all aspects relative to maintaining an inventory of all portfolio items.

- A "Purchasing Manger" role for an employee who needs to manage all asset procurement procedures.

A user can have several user roles.

The administrator must select a role, among all possible roles associated with a user, to assign to the user when he/she connects to the Asset Manager database. This role is the "role used at connection".

Important:

Any user that does not have administration rights must have a "role used at connection" in order to connect to the Asset Manager database. Users can change their role if necessary ▶ Selecting the user's role used at connection [page 178]

Definition of a user profile

A user profile is composed of:

- User rights and access restrictions to tables, fields and links in the Asset Manager database.
- Access rights (functional rights) to functional domains in Asset Manager.

For example, you can define:

- A "Helpdesk technician" profile, with access restricted to tables concerning the helpdesk.
- An "Accounting" profile, with access restricted to cost centers, budgets and expense lines only.

A user profile cannot be directly attached to a user. A profile is attached to a user only via a user role with which the profile is associated.

Definition of a user right

A user right is one of the component parts of a user profile. It involves rights granted for tables, fields and records in the Asset Manager database. Administrators can assign rights:

- For tables: Create, delete.
- For fields and links in tables: Read, insert and modify data

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 modify) records.

Definition of a functional right

A functional right is one of the component parts of a user profile. It is based on the functional domains available in Asset Manager (Procurement, Finance, Portfolio, etc). Functional rights provide access to different functional domains and to the various screens and wizards that are available in these domains. As an administrator, you can assign rights to users according to their job function and to what tasks they need to perform in Asset Manager.

Defining access conditions

This section describes how to define access conditions:

- Defining user roles [page 169]
- Defining user profiles [page 170]
- Defining user rights [page 170]
- Defining access restrictions [page 172]
- Defining functional rights [page 174]

Once defined, these access conditions can be associated with Asset Manager users.

Prerequisites: Import access management line-of-business data

The access management line-of-business data contains a set of typical user roles and profiles (along with user rights, access restrictions and functional rights) that you can use to create your own records.

Importing line-of-business data when creating the database

To insert the line-of-business data during database creation, ▶ Creating the database structure with Asset Manager Application Designer [page 59].

On the **Data to import** page, select **Administration - User roles profiles rights and restrictions**.

Importing line-of-business data into an existing database

To insert line-of-business data into an existing database:

1. Start Asset Manager Application Designer.
2. Select the **File/ Open** menu item.
3. Select the **Open database description file - create new database** option.
4. Select the `gbbase.xml` file which is located in the `config` sub-folder of the Asset Manager installation folder.

5 Start the database creation wizard (**Action/ Create database** menu).

6 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 database into which you wish to import the data.
Create	Select the **Import line-of-business data** option.
Use advanced creation options	Do not select this option.

Creation parameters page:

Fields	Value
Password	Enter the Asset Manager administrator's password. Note: The Asset Manager database administrator is the record in the **Employees and departments** (amEmplDept) table for which the **Name** (Name) field is set to **Admin**.

Data to import page:

Fields	Value
Available data	Select the **Administration - User roles profiles rights and restrictions** option
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.

7 Execute the wizard (**Finish** button).

Defining user roles

The definition of user roles is done on via the **Administration/ Rights/ User roles** navigation menu. Each user role groups the different user profiles that are associated with it.

For example: The "Purchase order management" and "Supplier management" profiles can be associated with a "Procurement manager" role.

Defining user profiles

User profiles are defined:

- Using the **Administration/ Rights/ User profiles** navigation menu.
- Or by clicking + (Windows client) or **Add** (Web client) in the list of user profiles associated with a user role (**General** tab of the user role detail).

Each user profile brings together:

- **User rights**, defining the rights of reading, writing, creating or deleting information in database tables, fields or links.
- **Access restrictions**, defining read and write conditions for records in a given table.
- **Functional rights**, defining access to available screens and wizards related to the Asset Manager functional domains.

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

Defining user rights

User rights can be managed via:

- The **Administration/ Rights/ User rights** menu.
- Or by clicking + (Windows client) or **Add** (Web client) to the right of the list of user rights associated with a user profile (**Roles, rights and restrictions** tab of the user profile detail).

User rights describe how users can access database tables, fields and links.

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

Precaution

Note:

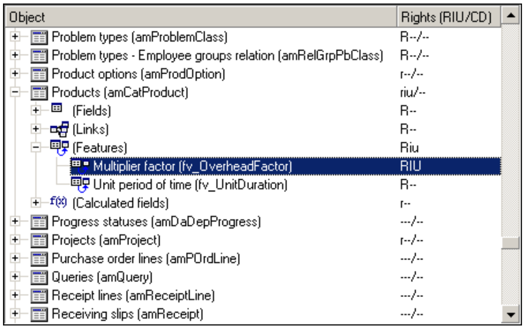
In order for a user to be able to see a "conditional" tab, they must have at least the right to view the fields that determines whether this tab is shown or not. For example, if a user does not have the right to view the **Nature of payments** field (SQL name: sePayType) in the **General** tab of a contract 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.

Windows client

A tree structure of the tables is used to present the database structure. 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 detail screen 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)	riu/--
(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.), Asset Manager 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.

Web client

The user rights detail screen is organized as follows:

- A list of tables
    - **Show entries** field: select the number of tables per display.
    - **Search** field: display the tables whose **Table name** or **SQL Name** contains the string you have entered in this field.
    - Table list: click the Table Name or SQL Name of the table of which you want to edit the access rights, then the components of this table will display in a category view at the lower part of the screen.
- A category list of the table components

  The components of each table are grouped by **Fields**, **Links**, **Calculated fields** and **Features** (if there is). You can assign rights to the table, or to specific fields, links and features of the table.
    - When you select a parent node (Table, Fields, Links, Features, etc.), Asset Manager automatically selects all of its child nodes; this authorizes you to edit user rights for the entire node. For a parent node, a lower case "r"/"i"/"u" indicates that partial items under the parent node have read/create/update rights; an upper case "R"/"I"/"U" indicates that all the items in the node have read/create/update rights.
    - **Cache Update** button: click **Cache Update** after you have selected the rights for one table; your selections are stored in the cache and you can continue selecting the rights for other tables.

Defining access restrictions

Access restrictions can be managed via:

- The **Administration/ Rights/ Access restrictions** menu.

- Or by clicking + (Windows client) or **Add** (Web client) in the list of access restrictions associated with a user profile (**Roles, rights and restrictions** tab of the user profile detail).

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

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 these access restrictions with user rights to create user profiles:

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

Note:

Access restrictions can not be defined on the following tables, which are loaded in cache when the client is opened.

- amAction
- amCalcField
- amCbkRule
- amCurRate
- amCurrency
- amDateAlarm
- amFeatMemo
- amFeatParam
- amFeature
- amFeatureClass
- amFieldAdjustTempl
- amFuncDomain
- amOption
- amReconcProposal
- amReport
- amScriptLibrary
- amSysConfig
- amTaxFormula
- amUnit
- amViewDef

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 Asset Manager query editor.

Read condition

Allows the user to view records that satisfy the criteriadefined 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 Asset Manager.

Verifying 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** 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 functional rights

Functional rights can be managed via:

- Using the **Administration/ Rights/ Functional rights** menu.

- Or by clicking + (Windows client) or **Add** (Web client) in the list of functional rights associated with a user profile (**Roles, rights and restrictions** tab of the user profile detail).

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

For example:

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

In the Windows client, the functional-right edition screen proposes a hierarchical list of all of Asset Manager's functional rights, which are defined by the user.

In the Web client, the screen is composed of two parts:

- A list of functional domains
  - **Show entries** field: select the number of functional domains per display.
  - **Search** field: display the functional domains whose **Name** or **SQL Name** contains the string you have entered in this field.
  - Table list: click the Name or SQL Name of the functional domain of which you want to edit the functional rights, then the functional right details of this domain will display in a hierachical view at the lower part of the screen.
- A hierachical functional right list of the selected functional domain

  You can define the functional rights for each item under the selected functional domain. After you have finished defining functional rights for one domain, click **Cache Update** to cache your selections, then you can continue setting other domains.

The logic of functional rights

Each functional domain (**Item** column) is composed of a set of items to which an access right is assigned (**Granted**, **Denied**, **Value of parent**). The items in a functional domain are, for the most part, the screens and wizards that are available in this domain. The following elaborates the rules of defining the functional rights.

- An item inherits the rights value of its parent if the **Value of parent** is assigned to this item.
- The **Granted by default** option only affects the top level items, the right is granted to those top level items whose **Value of parent** option is selected.
- Each child item of a domain can have a different rights value from that of the domain. In this case, you can assign **Granted** or **Denied** to a child item regardless of the rights value of the parent.

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

- The **Granted by default** option is not selected.
- 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 **Granted by default** option is selected.
- The functional domain has the value **Denied**.
- The set of items in the domain have the **Value of the parent** value.

or

- The **Granted by default** option is not selected
- The functional domain has the value **Denied**.
- The set of items in the domain have the **Value of the parent** value.

Associating access conditions with Asset Manager users

To assign access conditions with an Asset Manager user:

1. Display the list of employees (**Organization management/ Organization/ Employees** link on the navigation bar).
2. If the record corresponding to the employee does not exist yet, create it.
3. Move to the **Profile** tab in the employee detail.
4. Enter a user name and password in the **User name** (SQL name: UserLogin) and **Password** (SQL name: LoginPassword) fields. The user name 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 user name.

   Tip:

   If password management is handled by another system (LDAP server, Active Directory, SiteMinder, etc.), the password entered into the Asset Manager database is not taken into account, except for **Admin**.

5. Populate the **Access type** (SQL name: seLoginClass) field. Refer to the following paragraph to see the different options.

6. Assign the **Authorized user roles** (SQL name: MasterProfiles), the user roles can be associated with the employee.
    - Windows client: click **New** in the screen displayed by clicking ▤ to the right of the **Role used at connection** field.
    - Web client: click **Add** in the list at the lower right part of the detail screen.
7. In the **Role used at connection** (SQL name: DefMastProfile) field, select the user role for the employee to use when he/she connects to the database. Values available for this field are the values assigned for the **Authorized user roles** field (SQL name: MasterProfiles). The role used at connection (just like the authorized user roles) encompasses the access conditions (user profiles and user rights, access restrictions, associated functional rights) for the user.

    > Note:
    >
    > You can also grant administration rights to a user ▶ Defining a database administrator [page 177]. In this case, administration rights take precedence over specific access conditions that were previously assigned to the user and the **Role used at connection** (SQL name: DefMastProfile) and **Authorized user roles** (SQL name: MasterProfiles) fields no longer appear on the **Profile** tab of the employee's detail.

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 Asset Manager 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 Asset Manager installation, this is the only name available for accessing the Asset Manager 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.

Selecting the user's role used at connection

A user can modify his/her "role used at connection" via the **Select your Asset Manager user role** wizard. The wizard displays:

- Automatically when the user connects to Asset Manager, if the administrator enabled the **Select your Asset Manager user role** action for the user (**Action on login** (LoginAction) field on the **Profile** tab of the user's detail).
- From the **Administration/ User Actions/ Select your Asset Manager user role** link on the navigation bar.

The new "role used at connection" will take effect the next time the user connects to the database.

Managing user connections

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

Database access types

Asset Manager defines several types of access to the database.

The access type for an employee is defined on the **Profile** tab of the employee's detail page.

Access types govern connections made to the database via the Asset Manager Web and Windows interfaces or via Asset Manager API.

> Note:
>
> The access types do not take into account connections to the database made via Asset Manager Export Tool, Asset Manager Automated Process Manager, or Asset Manager Application Designer.

Table 12.1. Database access types

Access type	Floating	Named	Guest
Number of Logins that you can create	Unlimited	Limited by license	Unlimited
Number of simultaneous database connections	Limited by license	Unlimited	Limited by license

Access type	Floating	Named	Guest
Possibility to connect simultaneously with the same Login	Authorized but not recommended. Each connection reduces the number of remaining free connections.	Not authorized	Authorized but not recommended. Each connection reduces the number of remaining free connections.
Access to tables	No restrictions	No restrictions	Restrictions apply (▶ following tables). You cannot remove these restrictions. However, you can add additional access restrictions via the **Guest user** profile that is associated with the user's role. This profile is available in the access management line-of-business data: ▶ Prerequisites: Import access management line-of-business data [page 168] When you create a guest user, you must assign a user role with a profile that has the appropriate user and functional rights to that guest user, just as you would do for other access types.

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

A user whose access type is **Guest** only has access to the following tables:

Table 12.2. Guest access type - Tables may be accessed in read and/or write mode

SQL name of the table	Read access	Write access
amAccessRestr	X	
amAction	X	
amActionMemo	X	

SQL name of the table	Read access	Write access
amBrand	X	
amBudgCenter	X	
amBudget	X	
amCatalog	X	
amCatProduct	X	
amCatRef	X	
amClientResource	X	
amCnxType	X	X
amComment	X	X
amCompany	X	
amConnection	X	X
amContact	X	
amCostCategory	X	
amCostCenter	X	
amCounter	X	X
amCountry	X	
amCRType	X	
amCurrency	X	
amDocument	X	X
amDownTimePeriod	X	
amEmplDept	X	
amEmplGroup	X	
amEscalScheme	X	
amEscSchLevel	X	
amExpenseLine	X	X
amFuncDomain	X	
amFuncRight	X	
amFVExpenseLine	X	X
amHelp	X	
amHelpMemo	X	
amHistory	X	X
amImage	X	X
amItemizedList	X	
amItemListVal	X	
amKnowlBase	X	X
amLocation	X	
amLoginSlot	X	X
amMail	X	X
amModel	X	
amModelPort	X	
amModelSlot	X	
amNature	X	
amNews	X	
amPasswordHist	X	
amPortFolio	X	
amProblemClass	X	
amProdClassCode	X	

SQL name of the table	Read access	Write access
amProfile	X	
amProject	X	
amQuery	X	
amRelAstReplReq	X	X
amRelCalcFeat	X	X
amRelCatalogClients	X	X
amRelCatalogSuppliers	X	X
amRelCCatClass	X	
amRelCountLic	X	X
amRelCountSoft	X	X
amRelEmplEntitl	X	X
amRelEmplGrp	X	X
amRelEmplGrpCntr	X	X
amRelEntitlContract	X	X
amRelEntitlLoc	X	X
amRelEntitlPortf	X	X
amRelEntitlReq	X	X
amRelEscSchLoca	X	X
amRelEscSchSev	X	X
amRelFixedAsset	X	X
amRelFRProfile	X	X
amRelGrpLoca	X	X
amRelGrpPbClass	X	X
amRelItemAlarmTrig	X	X
amRelLoanAstSel	X	X
amRelLoanPayExpLin	X	X
amRelModelCompat	X	X
amRelModelEntitl	X	X
amRelNewsGrp	X	X
amRelPOrdReq	X	X
amRelPortfReql	X	X
amRelPortfUsers	X	X
amRelProfileRight	X	X
amRelRestrProfile	X	X
amRelSlotTypeModel	X	X
amRelSuppPCardType	X	X
amRelTermLoc	X	X
amRelTickAlarmTrig	X	X
amRelWfActivAct	X	X
amRelWOItems	X	X
amScriptLibrary	X	
amSeverity	X	
amSlot	X	
amSlotType	X	
amSoftLicCounter	X	
amStatistic	X	
amStock	X	

SQL name of the table	Read access	Write access
amSysConfig	X	X
amTaxFormula	X	
amTaxJuris	X	
amTaxRate	X	
amTaxType	X	
amTimeZone	X	
amTraining	X	
amUnit	X	
amUserRight	X	
amViewDef	X	
amWfActivAlarm	X	X
amWfActivity	X	
amWfEvent	X	X
amWfInstance	X	X
amWfOccurEvent	X	X
amWfOrgRole	X	
amWfScheme	X	
amWfSyncPoint	X	
amWfSysActiv	X	
amWfTransition	X	
amWfUserChoice	X	
amWfWorkItem	X	X
amWkEvtScript	X	
amWorkCalendar	X	

Table 12.3. Guest access type - Tables may be accessed in read-write mode under certain conditions

SQL name of the table	Read condition	Write condition
amReqLine	(CurrentUserId() IN (Request.Requester,Request.Requester.Supervisor) AND Request.seStatus<>1) OR (Request.seStatus=1)	CurrentUserId() IN (Request.Requester,Request.Requester.Supervisor)
amRequest	(CurrentUserId() IN (Requester,Requester.Supervisor) AND seStatus<>1) OR (seStatus=1)	CurrentUserId() IN (Requester,Requester.Supervisor)
amTicket	CurrentUserId() IN (Assignee,Contact,Contact.Supervisor)	CurrentUserId() IN (Assignee,Contact,Contact.Supervisor)
amTicketHistLine	(CurrentUserId()=Ticket.Contact.Parent.Supervisor OR CurrentUserId()=Ticket.Contact)	(CurrentUserId()=Ticket.Contact.Parent.Supervisor OR CurrentUserId()=Ticket.Contact)
amWorkOrder	CurrentUserId() IN (Requester,Requester.Supervisor,Technician)	CurrentUserId() IN (Requester,Requester.Supervisor,Technician)

SQL name of the table	Read condition	Write condition
amPhone	Unconditional read access	CurrentUserId() IN (PortfolioItem.User,PortfolioItem.User.Supervisor)
amPort	Unconditional read access	CurrentUserId() IN (User,User.Supervisor)
amSoftInstall	Unconditional read access	CurrentUserId() IN (Portfolio.User,Portfolio.User.Supervisor)
amAsset	Unconditional read access	CurrentUserId() IN (PortfolioItem.User,PortfolioItem.User.Supervisor)
amComputer	Unconditional read access	CurrentUserId() IN (Portfolio.User,Portfolio.User.Supervisor)
amMonitor	Unconditional read access	CurrentUserId() IN (PortfolioItem.User,PortfolioItem.User.Supervisor)

Length of validity of the connection

It is possible to define a length of validity for a given connection. This is defined on the employee' s detail page, **Profile** tab, **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

Whenever a user connects to the database via the Asset Manager Windows client or Web client, they are assigned a connection slot.

For as long as the user is connected to the database, the connection slot is regularly updated by Asset Manager.

- Windows client

  The update frequency is defined by the **Connection slot update delay (minutes)** option in the "Access control" section of the database options menu (**Administration/ Database options**) . By default, it is set to 5 minutes.
- Web client

  The update frequency is defined in the **slotUpdateInterval** configuration entry in the `<Tomcat installation folder>\webapps\AssetManager\WEB-INF\web.xml` file. By default, it is set to 300 seconds.

```
<env-entry>
<description>Loginslot  update interval in seconds</description>
<env-entry-name>slotUpdateInterval</env-entry-name>
<env-entry-type>java.lang.Integer</env-entry-type>
<env-entry-value>300</env-entry-value>
</env-entry>
```

Note:

Because connection slot refreshing involves additional queries to the database for each online user, if you have numerous concurrent users on Asset Manager and do not need real time monitoring of the user activities, we recommend you to extend the update frequency.

Note:

The Web connection slot information is not immediately reflected in the **amLoginSlot** table after the user login.

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

Detail of a connection slot

The Asset Manager administrator can view connection slots via the **Administration/ Connections/ Connection slots** navigation menu.

A connection slot is defined by:

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

Tip:

You can create a workflow on the connection slots (SQL name: amLoginSlot) table to count the named and floating users periodically. The historical statistics of the workflow can provide a clear uage of Asset Manager.

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 Asset Manager. To do this:

1. Display the connection slot screen via the **Administration/ Connections/ Connection slots** navigation 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 Asset Manager administrators.

Detecting accidental termination of the application

In some cases, the application that uses a connection slot may be terminated accidentally (e.g. 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 in the Windows client, or via clicking **Logout** in the Web client).

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.

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

1. Asset Manager 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, Asset Manager 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 filter to display all records in the table of connection slots whose **Modified on** field indicates a date earlier than a given date.

   Note:

   The **Modified on** (SQL name: dtLastModif) field cannot be viewed via the records detail window.

   ---

2. Select those slots and delete them.

Disconnecting inactive users

Asset Manager allows you to disconnect inactive users automatically according to a given timeout value.

Windows client

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

1. Activate the automatic disconnection procedure.
   - The **Enable automatic disconnection** in the "Access control" section option allows you activate the floating user automatic disconnection.
   - The **Enable automatic disconnection of named users** in the "Access control" section option allows you to activate the named user automatic disconnection.
2. After you have activated the automatic disconnection procedure, you need to define a value in the **Slot timeout (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 Asset Manager version 3.01 or earlier, the automatic disconnection is applied to the ensemble of users.

Web client

The inactive Web client users are automatically disconnected after the session timeout duration of the application server. The following instructs you how to set the session timeout value for different application servers supported by Asset Manager.

Apache Tomcat

The timeout value is set by the <session-timeout> parameter in the `<Tomcat installation folder>\conf\web.xml`.

The default timeout value is 30 minutes:

```
<session-config>
<session-timeout>30</session-timeout>
</session-config>
```

Oracle WebLogic

The timeout value can be defined in either `web.xml` or `weblogic.xml`.

- In the `web.xml` file, it is defined by the <session-timeout> parameter in minutes, the default value is 0.

  ```
  <session-config>
  <session-timeout>0</session-timeout>
  </session-config>
  ```

- In the `weblogic.xml` file, it is defined by the TimeoutSecs element in seconds, the default value is 3600.

  ```
  <session-param>
  <param-name>TimeoutSecs</param-name>
  <param-value>3600</param-value>
  </session-param>
  </session-descriptor>
  ```

IBM WebSphere

The timeout value is set by the <session-timeout> parameter in the `web.xml` file. The default value is 30 minutes.

```
<session-config>
<session-timeout>30</session-timeout>
</session-config>
```

Managing user authentication for the Asset Manager database

This section explains how to manage user access to the Asset Manager database via different authentication systems.

Some of these systems enable Asset Manager Windows and Web clients to access the Asset Manager database using authentication information that they have already provided when accessing another system, and do not require users to re-authenticate.

Note:

Single Sign-On (SSO) definition A directory-based software solution that provides an organization's network users seamless access to all authorized resources after users have completed a single network authentication procedure.

A single password lets a user access all multi-platform applications and systems in the organization. Once the user has entered his or her password during initial authentication, the directory (for example, Active Directory under Windows) sends the password automatically to all other applications that require it without requesting the password information from the user again.

Warning:

To implement Single Sign-On, the Windows and Web clients must be specially configured for use with Single Sign-On.

The password that controls user access to the Asset Manager database can be stored in several ways:

Asset Manager Windows and Web clients: The **Admin** login - a special case

The **Admin** login is always stored in the Asset Manager database, whether the other passwords are stored in the Asset Manager 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 Asset Manager, this is the only login name available for accessing the Asset Manager 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 Asset Manager database.

⚠ Warning:

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

⚠ Warning:

Anyone who knows the **Admin** password can perform any operation on the Asset Manager database and can access all the database information with no restrictions. Make sure you do not tell this password to any unauthorized employees.

✎ Note:

If despite the precautions specified above you forget this password, there is no way of recovering it using Asset Manager. Call HP support.

Modifying the password for the Admin login

Windows client

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;
- select **Administration/ Change password** from the navigation tree.

Web client

You can modify the password by logging in the Web client with **Admin** and select **Administration/ Change password** from the navigation tree.

Asset Manager Windows and Web clients: Authentication is handled entirely by Asset Manager

This section explains how to set up Asset Manager to manage all aspects of user authentication.

In this mode, user information is only stored in the Asset Manager database.

Users must enter their **login** and **password** each time they connect to the Asset Manager database.

Configure password management

Enter database options related to passwords

1. Start the Asset Manager Windows or Web client.
2. Connect to the Asset Manager database using a user login that has administrative rights.
3. Select the **Administration/ User actions/ Password management...** link.
4. Populate the **Password management...** (sysCorePassMgt) wizard.

This wizard enables you to define standard options concerning a password (**Default number of days a new password is valid**, **Password expiration notification**) as well as the options called by the script library (**Minimum password length**, **Minimum number of different characters**).

Note:

When users have logged in the Asset Manager Windows and Web clients, they will be notified to change their passwords if they are approaching the expiration. **Password expiration notification** defines how long in advance to change the passwords.

When users attempt to log in the Asset Manager Windows and Web clients with already expired passwords, they will be forced to change them before logging in.

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.

Define password format constraints

Password format constraints are defined in a script library. The default library is called **PasswordLib**. This library can be modified via the **Administration/ Scripts** link on the navigation bar.

Important:

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

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

- 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

Create a user who can connect to the Asset Manager database

1. Start the Asset Manager Windows or Web client.
2. Connect to the Asset Manager database using a user login that has administration rights.

   Note:

   Enable the **Administration rights** (bAdminRight) option on the profile tab of an employee detail page to enable these rights.

   ---
3. Display the employees (**/Organization management/ Organization/ Employees** link on the navigation bar).
4. Create the employee if she/he does not already exist.
5. Populate the information on the **Profile** tab.
6. Populate the **User name** (UserLogin) and **Password** (LoginPassword) fields.

   The values used for these fields are required when connecting to the database.

Modify a password

Modifications performed by the user

1. Start the Asset Manager Windows or Web client.
2. Connect to the Asset Manager database with your current user name and password.
3. Select the **Administration/ User actions/ Change password...** link.
4. Populate the wizard.

Modifications performed by the administrator

User passwords can be modified by users whose administrative rights option is enabled on the **Profile** tab of their employee detail page. This erases the old password.

Locking out a user

1. Start the Asset Manager Windows or Web client.
2. Connect to the Asset Manager database using a user login that has administration rights.
3. Display the employees (**/Organization management/ Organization/ Employees** link on the navigation bar).
4. Select the employee.
5. Display the **Profile** tab.

6. Set the **Status of the user account** field (seLoginStatus) to **Locked**.

Defining an access validity period for a user

1. Start the Asset Manager Windows or Web client.
2. Connect to the Asset Manager database using a user login that has administrative rights.
3. Display the employees (**/Organization management/ Organization/ Employees** link on the navigation bar).
4. Select the employee.
5. Display the **Profile** tab.
6. Populate the information in the **Time-based access control** frame.

Lost passwords

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

Asset Manager Windows and Web clients: Authentication is handled entirely by LDAP

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 and LDAP directory, you might find it useful to use the directory to control access to the Asset Manager 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 Asset Manager users can access the LDAP directory.

2. Create the Asset Manager users in the LDAP directory and assign them a login and password.
3. Deactivate the **Anonymous** access to the LDAP directory.
4. Launch Asset Manager.
5. Start the Asset Manager Windows client.
6. Connect to the working database via the **File/Connect to database** menu.
7. Display the database options via the **Administration/Database options** menu.
8. Populate the options having the section name **LDAP authentication**.
9. Create or modify the users of the **Employees and departments** table (amEmplDept) by populating the fields in the **Profile** tab.

   In particular, populate the following fields:

   - **User Name** (UserLogin): Login to use to connect to the Asset Manager 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 HP Connect-It.

   ---

10. Make sure that the LDAP DLLs are installed on the Asset Manager user computers.

    This is the case if you selected the **Full installation** option when you installed the Asset Manager 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.
11. Have the users reconnect to the database in order for the new options to be taken into account at the level of their Asset Manager client.

LDAP connection to the Asset Manager database

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

1. They start an Asset Manager Windows or Web client.
2. They connect to the production database:

- User name: User name that is stored in the **Employees** table (amEmplDept) in the **User name** (UserLogin) field of the user's detail.

  Asset Manager 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 Asset Manager.

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.

   > Note:
   >
   > This option is used more often for HP 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.

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

Asset Manager Windows client: Implementing Single Sign-On using Active Directory on IIS

This section explains how to use Active Directory to define Asset Manager Windows client users.

Thanks to Windows integrated authentication, a user that authenticated when Windows started does not need to authenticate when the Asset Manager Windows client is started.

Rights that users have in Asset Manager are defined in the Asset Manager database (user roles).

You must synchronize authentication information between Asset Manager and Active Directory to ensure that Single Sign-On works properly.

With this synchronization:

- You automatically import the list of Active Directory users into the Asset Manager database.
- Users who connect to an Asset Manager database don't have to populate the **Login** and **Password** fields when connecting to the database.

General overview

Synchronization between Asset Manager and Active Directory is performed in one of the following ways:

Synchronizing all Active Directory users

- Synchronization is performed via a HP Connect-It scenario.
- The HP Connect-It scenario that performs this task is programmed and launched by Asset Manager Automated Process Manager.
- HP Connect-It uses the following combination of fields to identify persons in the Asset Manager database:
  - **Name** (Name)
  - **First name** (FirstName)
  - **User Name** (UserLogin)
- A user is authorized to connect to the Asset Manager database using Windows integrated security with a given login if the following items have the same value:
  - **SID** (Identifier) field of the Asset Manager user.
  - Active Directory **SID** field.

Synchronizing a user in the Asset Manager database

On the users detail page, click the **NT user** button to start the wizard.

Main steps to implement full synchronization

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.

- Step 1: Create users and groups in Active Directory [page 196]

  Objective: Prepare the information to transfer to Asset Manager.
- Step 2: Configure Asset Manager Automated Process Manager and HP Connect-It [page 197]

  Objective: Define which **NT** domains should be retained and plan the transfers.
- Step 3: Activate the updating process for the first time [page 197]

  Objective: Transfer the information for the first time and test the login.
- Step 4: Complete the description of the employees in the Asset Manager database [page 197]
- Step 5: Activate the automatic triggering of the Asset Manager Automated Process Manager modules [page 197]

Step 1: Create users and groups in Active Directory

The fields to populate are:

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

    > Tip:
    >
    > When you create users in the Asset Manager database, Asset Manager Automated Process Manager takes the value of the **Full name** field from Active Directory and searches 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 name** field; everything to the right of the first space creates the **Name** field. If there is no space, only the **Name** field will be populated.
    >
    > You must not use a space for first names that are actually combined of two or more names. If you wish, you can use a hyphen instead.
  - **Description** field:

    This field is used to populate the **Comment** field (Comment) in the Asset Manager database.
- For groups: The **Name** field.

  This field is used to populate the **Name** field (Name) in the Asset Manager database.

Step 2: Configure Asset Manager Automated Process Manager and HP Connect-It

▶ **Administration Guide**, chapter **Asset Manager Automated Process Manager**, section **Configuring the modules monitored by Asset Manager Automated Process Manager**, sub-section **Add NT users to the database module (AddUser)**.

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**.

   Asset Manager begins the update.

> ✎ Note:
>
> If the number of users concerned by this operation exceeds the number of **Named** logins authorized by the License Keys activated in your database, Asset Manager Automated Process Manager declares the users as having **Floating** type logins.

Step 4: Complete the description of the employees in the Asset Manager 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 Asset Manager user. You must notably re-enter the value of the **Password** field.

Step 5: Activate the automatic triggering of the Asset Manager Automated Process Manager modules

1. Select the **Tools/ Configure modules** menu item.
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**.

Asset Manager connection

Once the previous operation is completed, the NT users can directly access Asset Manager. 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.

Rules applied by Asset Manager during the creation/modification of users

- The Asset Manager 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 Asset Manager database:
  - **Name** (Name)
  - **First name** (FirstName)
  - **User Name** (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 HP Connect-It scenario are then populated.

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

  Note:

  The **Password** field is populated with a special character that prohibits the user to access the Asset Manager 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 Asset Manager 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 Asset Manager.

Asset Manager Web client: Implementing Single Sign-On using Active Directory on IIS (prior to version 7)

Using Single Sign-On to access Asset Manager Web can be handled in several ways depending on the environment you are using.

This section does not describe all possibilities and is limited to a description of Single Sign-On authentication using Active Directory, and versions of IIS prior to IIS 7.

A procedure for IIS 7 is given, in Asset Manager Web client: Implementing Single Sign-On (SSO) using Active Directory on IIS 7 32-bit [page 203]

Note:

You can use a Single Sign-On system other than Active Directory.

Use this example as a guideline and and tailor it to your needs.

IIS (Internet Information Services) is the only Web server that correctly supports Single Sign-On on Windows. However, you can use different application servers (Tomcat, WebSphere, etc), given that they are correctly configured to communicate with IIS.

The installation and configuration of Single Sign-On with Tomcat as the application server is described in this section.

The steps to perform on the machines where you have installed Asset Manager Web Service and Asset Manager Web Tier are listed below (if they have been installed on different machines, each of the steps needs to be performed on each machine).

Prerequisites

Before starting, make sure that you have the following:

- An operational Active Directory implementation on your network.

  The steps in this section describe an implementation using the Active Directory version provided with Windows Server 2003 R2, but are also valid for previous versions of Active Directory.
- An operational DNS (Domain Name Service) server. To check this, at the command line (command prompt window) enter the following:

  ```
  ipconfig
  ```

  Then:

  ```
  nslookup < IP address of the DNS server>
  ```
- Asset Manager Web using Tomcat as application server.
- IIS (Web server provided with Windows).

Install the Jakarta ISAPI redirector plugin

You must first establish a connection between IIS and Tomcat. Apache provides an ISAPI ISAPI (Internet Server Application Programming Interface) redirector plugin that needs to be implemented.

Please refer to the official **Apache Tomcat Connector - Webserver HowTo** documentation at http://tomcat.apache.org/connectors-doc/webserver_howto/iis.html.

Tip:

The Apache Tomcat connectors distribution is available from the following site: http://tomcat.apache.org/download-connectors.cgi

---

Configure ISS

1. On the desktop, right click **My Computer**.
2. Click **Manage**.
3. Expand the **Services and Applications/ Internet Information Services/ Web Sites/ Default Web Site** branch.
4. Right click **jakarta** then click **Properties**.
5. Click the **Virtual Directory** tab and check the parameters.

   The following items must be enabled:

   - When connecting to this resource, the content should come from: **A directory located on this computer**.
   - **Local Path** section:
     - Script source access
     - Read
     - Log visits
   - **Application Settings** section:
     - Application name: **jakarta**
     - Execute Permissions: **Scripts and Executables**
     - (If field available) Application Protection: **Medium (Pooled)**
6. Next, return to the **Default Web Site**, Right-click on it and click **Properties**.

   On the **ISAPI Filters** tab, you should see the **jakarta** filter with a green arrow pointing up, which indicates that the filter is functioning correctly.

   If this is not the case:

   a. Click **Add**.
   b. **Filter Name** field: Type **Jakarta**.

c **Executable** field: Browse to the `isapi_redirect.dll` file located in the **bin** subfolder of the folder where you installed the Jakarta ISAPI redirector.

For example: `C:\Program Files\Apache Software Foundation\Jakarta Isapi Redirector\bin\isapi_redirect.dll`

If the installation folder of the Jakarta ISAPI redirector is different on your machine, replace `C:\Program Files\Apache Software Foundation\Jakarta Isapi Redirector\` with the path to the correct installation folder.

This installation folder example will be used throughout this guide.

d Click **OK**, then click **OK** again on the Default Web Site properties page.

e Re-open the properties page.

You should see the green arrow pointing up in front of the Jakarta filter.

7. Still in the **Default Web Site Properties** window, click the **Directory Security** tab.
8. In the **Authentication and access control** section, click **Edit...**.
9. Clear **Anonymous access** option.
10. Enable the **Digest authentication for Windows domain servers** and **Integrated Windows authentication** options.
11. Make sure your domain name appears in the **Default domain** field.

Configure the port transfer

1. The objective is to transfer the port between **IIS** (which, by default, listens on port 80) and **Tomcat** (which listens on port 8080) and to tell Tomcat not to display a login window requesting authentication.

   To do this, open `C:\Tomcat\conf\jk2.properties` (the location of this file may be different depending on your Tomcat installation folder. This example will continue to use this path.) with a text editor and add the two following lines to the end of the file:

```
request.tomcatAuthentication=false
request.registerRequests=false
```

2. To check that the port transfer works correctly, perform the following test:

   Open the default Tomcat application and enter the following URL: http://localhost:8080/jsp-examples/.

   Note:

   If it does not work, make sure that you have launched Tomcat.

   Now, open the following URL in your browser:

http://localhost/jsp-examples/.

You should see the same page as before. This confirms that the port transfer has been correctly configured.

3. Next, you must instruct Jakarta to manage the Tomcat default application and Asset Manager.

   To do this, open `C:\Program Files\Apache Software Foundation\Jakarta Isapi Redirector\conf\uriworkermap.properties` in a text editor and add the following lines to the appropriate section:

   - On machines where Asset Manager Web Service has been installed, add the following line in the file:

     ```
     /AssetManagerWebService/*=wlb
     ```

   - On the machine where Asset Manager Web Tier has been installed, add the following line in the file:

     ```
     /AssetManager/*=wlb
     ```

   Note:

   If Asset Manager Web Service and Asset Manager Web Tier are installed on the same machine, add the lines given above to the appropriate section of the `C:\Program Files\Apache Software Foundation\Jakarta Isapi Redirector\conf\uriworkermap.properties` file.

   Restart Tomcat.

4. If you are connected as a Windows user, you should be able to access the Asset Manager Web client without having to authenticate again.
5. However, if a login screen still appears, restart your system.

   If the problem persists, open the `C:\Tomcat50\conf\server.xml` file in write mode and remove the comment characters (**<!--** and **-->**) for the following line:

```
<Valve className="org.apache.catalina.valves.RequestDumperValve"/>
```

This forces Tomcat to generate an additional log file named: `catalina_log.YYYY-MM-DD.txt`.

Open the log file and check the entries for the following parameters:

```
YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: queryString=null
YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: remoteAddr=192.178.0.
1 (type your IP address here)
YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: remoteAddr=192.178.0.
1 (type your IP address here)
YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: remoteUser=<Domain na
me>\<User name>
YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: requestedSessionId=nu
```

```
ll
YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: scheme=http
```

This example indicates that Single Sign-On is operating correctly.

IIS transmits all of the user's Windows authentication certificates. Tomcat then authenticates the user.

If the **remoteUser** field is empty, make sure that all the previous steps have been correctly completed.

Asset Manager Web client: Implementing Single Sign-On (SSO) using Active Directory on IIS 7 32-bit

This section applies to specifically IIS7 32-bit.

Prerequisites

Before starting, make sure that you have the following:

- An operational Active Directory implementation on your network.

  The steps in this section describe an implementation using the Active Directory version provided with Windows Server 2003 R2, but are also valid for previous versions of Active Directory.
- An operational DNS (Domain Name Service) server. To check this, at the command line (command prompt window) enter the following:

  ```
  ipconfig
  ```

  Then

  ```
  nslookup <IP address of the DNS server>
  ```
- Asset Manager Web using Tomcat as application server.
- IIS 7 (Web server provided with Windows, only supported in Windows Server 2008 and Windows Vista).

  For Windows Server 2008, default component selection of Web Services is enough for the configuration below.

  For Windows Vista, manually select the following options: component ISAPI Filter, ISAPI Extension and Windows Authentication.

> Note:
>
> For further information about installing IIS 7, refer to the following links:
>
> http://learn.iis.net/page.aspx/29/installing-iis7-on-windows-server-2008/
>
> http://learn.iis.net/page.aspx/28/installing-iis-70-on-windows-vista/

Note:

A procedure for IIS prior to version 7 is given, in Asset Manager Web client: Implementing Single Sign-On using Active Directory on IIS (prior to version 7) [page 199]

Install the Jakarta ISAPI redirector plugin

You must first establish a connection between IIS and Tomcat. Apache provides an ISAPI ISAPI (Internet Server Application Programming Interface) redirector plugin that needs to be implemented.

Please refer to the official **Apache Tomcat Connector - Webserver HowTo** documentation at http://tomcat.apache.org/connectors-doc/webserver_howto/iis.html.

Tip:

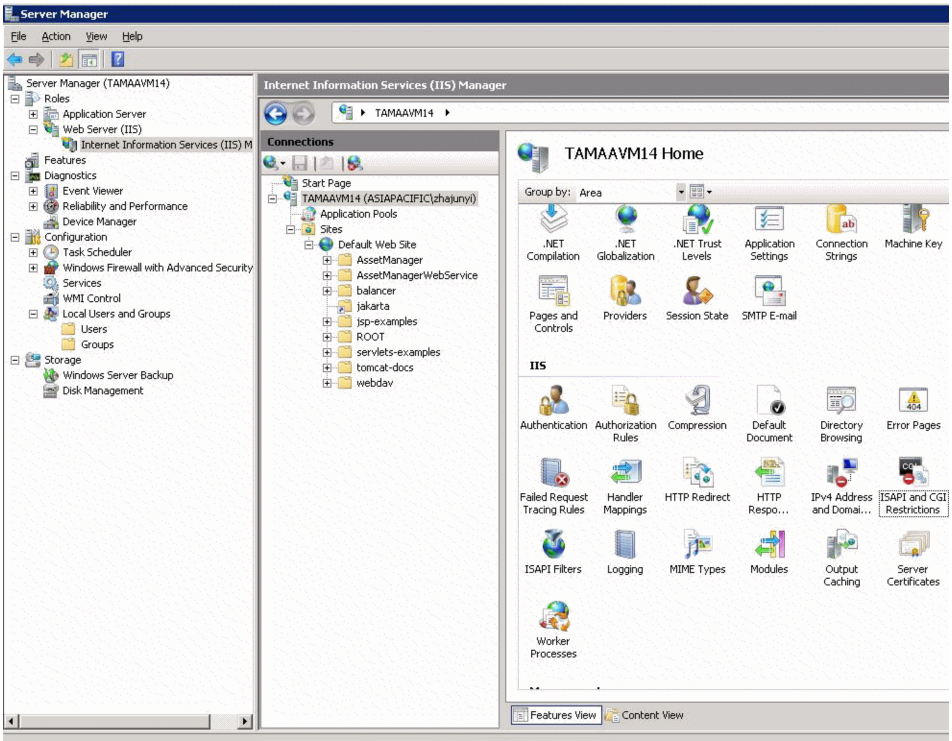
The Apache Tomcat connectors distribution is available from the following site: http://tomcat.apache.org/download-connectors.cgi

Configure IIS

1 On the desktop of the server where IIS is installed, right click **My Computer**.

2 Click **Manage**.

3 Expand the **Roles/ Web Server (IIS)/ Internet Information Services (IIS) Manager**, as in the example shown below:



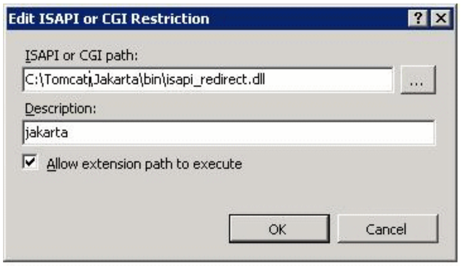
4 In the Internet **Information Services (IIS) Manager** panel, select your host, and then double click **ISAPI and CGI Restrictions**.

5 Click **Add** in the **Actions** list.

6 Set **ISAPI or CGI path** poiting to the `isapi_redirect.dll` file (e.g. `C:\tomcat\Jakarta\bin\isapi_redirect.dll`) and **Description** as **jakarta**.

Check **Allow extension path to execute**.

See the example below:


Edit ISAPI or CGI Restriction

ISAPI or CGI path: C:\Tomcat\Jakarta\bin\isapi_redirect.dll

Description: jakarta

☑ Allow extension path to execute

7 Click **OK** button.

8 Click on the **Default Web Site** and navigate to **ISAPI Filters**.

9 Click **Add** in the **Actions** list.

10 Add a new filter with **Executable** pointing to `isapi_redirect.dll` (e.g. `C:\tomcat\Jakarta\bin\isapi_redirect.dll`) and **Filter name** as **Jakarta**.

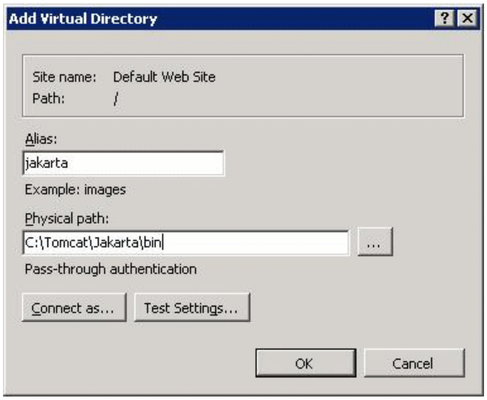
11 Click **OK**.

12 Click on the **Default Web Site** again.

13 Add a virtual directory and set **Alias** to **jakarta**.

Set the **Physical path** to point to the directory containing `isapi_redirect.dll`.

See the example below:


Add Virtual Directory

Site name: Default Web Site

Path: /

Alias: jakarta

Example: images

Physical path: C:\Tomcat\Jakarta\bin

Pass-through authentication

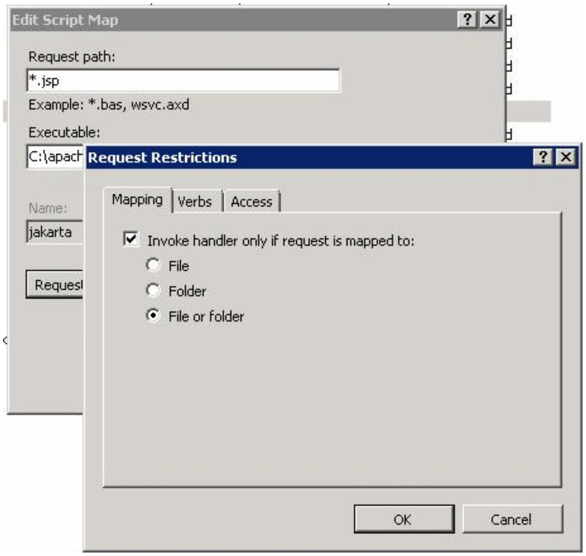
14 Click **OK**.

15. Click on the newly created virtual directory **jakarta**.
16. Double click the **Handler Mappings**.
17. Add a script map in the **Actions** list with **Request path** as ***.jsp**, **Executable** pointing to **isapi_redirect.dll**, **Name** as **jakarta**.
18. Click **Request Restrictions**.

    Check the following options:

    - Open **Mapping** tab, check **Invoke handler only if request is mapped to:**, select **File or Folder**
    - Open **Access** tab, select **Execute**.

      See the example below:



Edit Script Map

Request path: *.jsp

Example: *.bas, wsvc.axd

Executable: C:\apach

Name: jakarta

Request Restrictions

☑ Invoke handler only if request is mapped to:

○ File

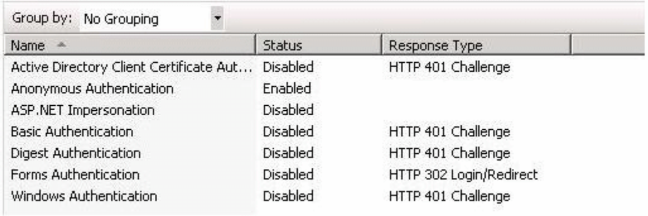
○ Folder

● File or folder

    - Click **OK** button to close the window.

19. Still in the **Default Web Site Properties** window, click **Authentication**.
20. Disable **Anonymous Authentication** and enable **Digest Authentication** and **Windows Authentication**.

See the example below:



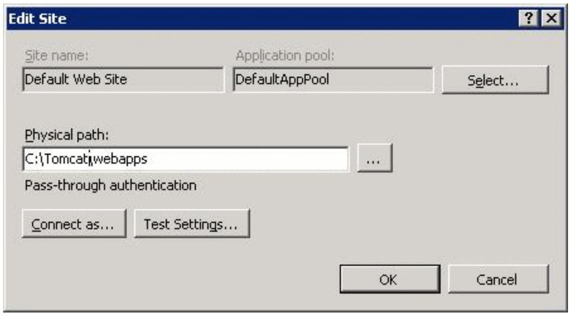
Name	Status	Response Type
Active Directory Client Certificate Aut...	Disabled	HTTP 401 Challenge
Anonymous Authentication	Enabled	
ASP.NET Impersonation	Disabled	
Basic Authentication	Disabled	HTTP 401 Challenge
Digest Authentication	Disabled	HTTP 401 Challenge
Forms Authentication	Disabled	HTTP 302 Login/Redirect
Windows Authentication	Disabled	HTTP 401 Challenge

21 Navigate to Default Web Site Properties.

22 Select **Basic Settings** in the **Actions** list.

23 Set **Physical path** to **<Tomcat installation folder>\webapps**.

See the example below.



Site name: Default Web Site

Application pool: DefaultAppPool

Physical path: C:\Tomcat\webapps

24 Click the **OK** button.

25 Navigate to your host again and click **Restart** in the **Actions** list.

Configure the port transfer

1 Open `<Tomcat installation folder>\conf\server.xml` with a text editor and disable tomcat authentication as in the following example:

```
<Connector port="8009" enableLookups="false" redirectPort="8443" protoc
ol="AJP/1.3" tomcatAuthentication="false" />
```

2 To check that the port transfer works correctly, restart tomcat and then perform the following test:

3 Open a browser and navigate to following URL:

http://localhost:8080/AssetManager/

Tip:

If it does not work, make sure that you have launched Tomcat.

4 Instruct Jakarta to manage the Tomcat default application and Asset Manager.

To do this, open `<Tomcat installation folder>\conf\uriworkermap.properties` in a text editor and add the following line in the file:

```
/AssetManager/*=wlb
```

5 Restart Tomcat and IIS

6 If you are connected as a Windows user, you should be able to access the Asset Manager Web client without having to authenticate again.

Trouble Shooting

Here are some of the most common problems, and potential solutions to them:

- I can open Asset Manager with 8080 port, but failed to open with port 80 with "Page not found" error.

  ANSWER: First, make sure IIS service is up, then open the Jarkata file (`<Tomcat installation folder>\Jarkata\conf\uriworkermap.properties`), check if the Asset Manager application has been added.

- After SSO is enabled, Asset Manager login is still required.

  ANSWER: Make sure all the previous steps are correctly completed. Open the `<Tomcat installation folder>\conf\server.xml` file in write mode and remove the comment characters (<!-- and -->) for the following line:

```
<Valve className="org.apache.catalina.valves.RequestDumperValve"/>
```

This forces Tomcat to generate an additional log file named: `Catalina_log.YYYY-MM-DD.txt`

Open this log file and check the entries for the following parameters:

```
YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: queryString=null
```

```
YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: remoteAddr=192.178.0.
1 (type your IP address here)
```

```
YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: remoteAddr=192.178.0.
1 (type your IP address here)
```

```
YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: remoteUser=<Domain na
me>\<User name>
```

```
YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: requestedSessionId=nu
ll
```

```
YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: scheme=http
```

If the **remoteUser** field is empty, check Tomcat authentication again. Open the `<Tomcat installation folder>\conf\server.xml` file, make sure Tomcat authentication is disabled.

- I get the following error:

```
Login failed with error "Error: com.peregrine.ac.AmException: Error (12
,008): Login 'xxx' unknown."
```

ANSWER: Check if the user exists in database and user has authorized role to access Asset Manager. Pay attention to the user login name. To check if login name contains domain name, open the `<Tomcat installation folder>\webapps\AssetManager\WEB-INF\classes\application-context.html` file and check the **keepDomain** value. If you want to keep domain name, update this value to "true". Otherwise, update it as "false". Restart Tomcat and try again.

Asset Manager Web client: Implementing Single Sign-On (SSO) using Active Directory on IIS 7 64-bit

This section applies to specifically IIS7 64-bit.

Prerequisites

Before starting, make sure that you have the following:

- An operational Active Directory implementation on your network.

  The steps in this section describe an implementation using the Active Directory version provided with Windows Server 2003 R2, but are also valid for previous versions of Active Directory.
- An operational DNS (Domain Name Service) server. To check this, at the command line (command prompt window) enter the following:

  ```
  ipconfig
  ```

  Then

  ```
  nslookup <IP address of the DNS server>
  ```
- Asset Manager Web using Tomcat as application server.
- IIS 7 (Web server provided with Windows, only supported in Windows Server 2008 64-bit and Windows Vista).

  Manually select the following component options if they are not already installed: **ISAPI Filter**, **ISAPI Extension** and **Windows Authentication**.

Note:

For further information about installing IIS 7, refer to the following links:

http://learn.iis.net/page.aspx/29/installing-iis7-on-windows-server-2008/

http://learn.iis.net/page.aspx/28/installing-iis-70-on-windows-vista/

Install Java for Microsoft Windows 64-bit

For details of how to install the 64-bit version of Java, as well as the download link, visit http://java.sun.com/j2se/1.4.2/install-windows-64.html

Once installed, you lust configure it:

1. On the Windows desktop, right click My Computer.
2. Select **Properties**, then navigate to the **Advanced** tab/ **Environment variables**
3. Under the **System variables** list, click **New**.
4. Create a new variable with the following properties:

   Variable name: **JAVA_HOME**

   Variable value: **<Java installation folder>\jdk1.5.0_19**

   Note:

   Adapt the above path to the folder where you installed Java 64-bit.

5. Click **OK** on each of the open configuration windows to save your changes.

Install the Jakarta ISAPI redirector plugin

You must first establish a connection between IIS and Tomcat. Apache provides an ISAPI ISAPI (Internet Server Application Programming Interface) redirector plugin that needs to be implemented.

1. Download the 64 bit redirector from the following site:

```
http://apache.oregonstate.edu/tomcat/tomcat-connectors/jk/binaries/win6
4/jk-1.2.28/amd64/isapi_redirect-1.2.28.dll
```

2. Put the `isapi_redirect-1.2.28.dll` file in your `<Tomcat installation folder>Jakarta\bin` folder. Rename it to `isapi_redirect.dll`

   Please refer to the official **Apache Tomcat Connector - Webserver HowTo** documentation at http://tomcat.apache.org/connectors-doc/webserver_howto/iis.html.

3. Use a text editor to create a file called `<Tomcat installation folder>\Jakarta\conf\workers.properties.minimal` with the following contents:

```
# workers.properties.minimal -
#
# This file provides minimal jk configuration properties needed to
# connect to Tomcat.
#
# The workers that jk should create and work with
#

worker.list=wlb,jkstatus

#
# Defining a worker named ajp13w and of type ajp13
# Note that the name and the type do not have to match.
#
worker.ajp13w.type=ajp13
worker.ajp13w.host=localhost
worker.ajp13w.port=8009

#
# Defining a load balancer
#

worker.wlb.type=lb
worker.wlb.balance_workers=ajp13w

#
# Define status worker
#

worker.jkstatus.type=status
```

4 Use a text editor to create a file called `<Tomcat installation folder>\Jakarta\conf\uriworkermap.properties` with the following contents:

```
# uriworkermap.properties - IIS
#
# This file provides sample mappings for example wlb
# worker defined in workermap.properties.minimal
# The general syntax for this file is:
# [URL]=[Worker name]

/AssetManager/*=wlb

#
# Mount jkstatus to /jkmanager
# For production servers you will need to
# secure the access to the /jkmanager url
#
/jkmanager=jkstatus
```

5 Add the redirector to the Windows Registry:
   a Use a text editor to create a file with a `.reg` extension and the following contents:

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SOFTWARE\Apache Software Foundation\Jakarta Isap
i Redirector\1.0]
@=""
"extension_uri"="/jakarta/isapi_redirect.dll"
"log_file"="C:\\Tomcat\\Jakarta\\log\\isapi_redirect.log"
"log_level"="info"
"worker_file"="C:\\Tomcat\\Jakarta\\conf\\workers.properties.minimal
"
"worker_mount_file"="C:\\Tomcat\\Jakarta\\conf\\uriworkermap.propert
ies"
```

> Note:
>
> Adapt the paths to your Tomcat installation folder.

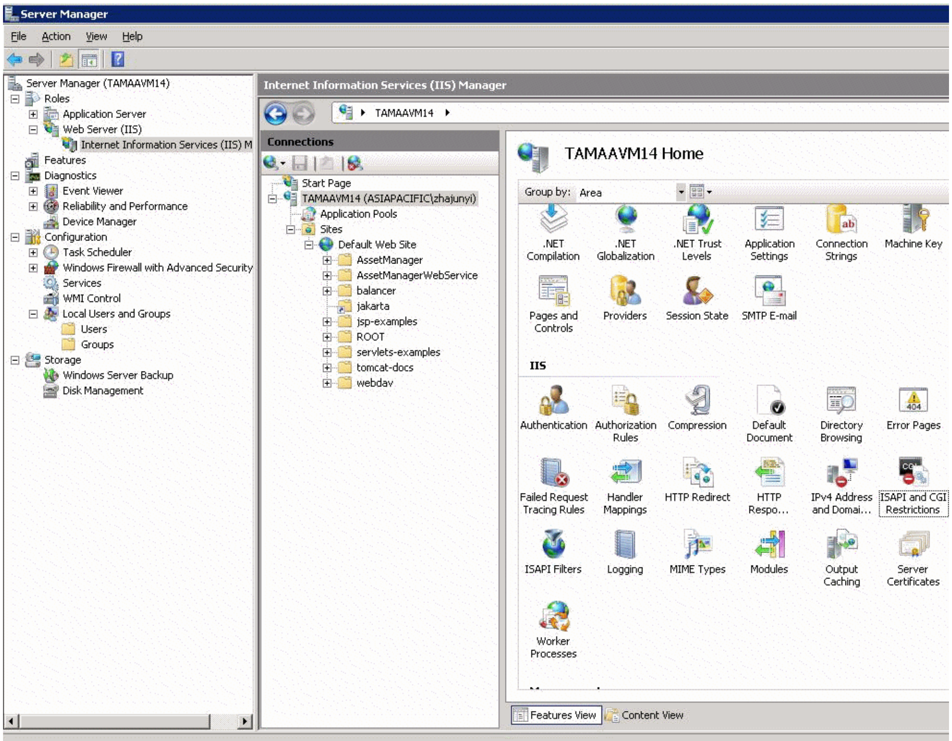
b Double click on the file to import the it into the Registry.

c Open the Registry Editor (Run **regedit** in the command line), expand **HKEY_LOCAL_MACHINE\SOFTWARE\Apache Software Foundation\Jakarta Isapi Redirector\1.0**, and verify that Jakarta is registered successfully.

Configure IIS

1. On the desktop of the server where IIS is installed, right click **My Computer**.
2. Click **Manage**.

3 Expand the **Roles/ Web Server (IIS)/ Internet Information Services (IIS) Manager**, as in the example shown below:



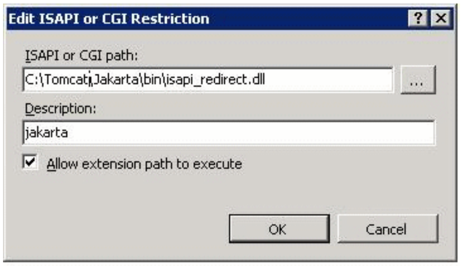
4 In the Internet **Information Services (IIS) Manager** panel, select your host, and then double click **ISAPI and CGI Restrictions**.

5 Click **Add** in the **Actions** list.

6 Set **ISAPI or CGI path** poiting to the `isapi_redirect.dll` file (e.g. `C:\tomcat\Jakarta\bin\isapi_redirect.dll`) and **Description** as **jakarta**.

Check **Allow extension path to execute**.

See the example below:



ISAPI or CGI path: C:\Tomcat\Jakarta\bin\isapi_redirect.dll

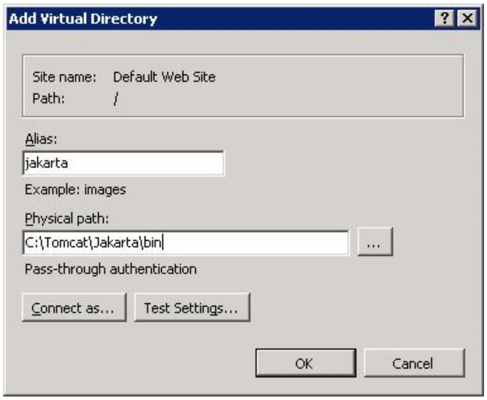
Description: jakarta

☑ Allow extension path to execute

7. Click **OK** button.
8. Click on the **Default Web Site** and navigate to **ISAPI Filters**.
9. Click **Add** in the **Actions** list.
10. Add a new filter with **Executable** pointing to `isapi_redirect.dll` (e.g. `C:\tomcat\Jakarta\bin\isapi_redirect.dll`) and **Filter name** as **Jakarta**.
11. Click **OK**.
12. Click on the **Default Web Site** again.
13. Add a virtual directory and set **Alias** to **jakarta**.

    Set the **Physical path** to point to the directory containing `isapi_redirect.dll`.

    See the example below:



Site name: Default Web Site

Path: /

Alias: jakarta

Physical path: C:\Tomcat\Jakarta\bin

14. Click **OK**.

15 Click on the newly created virtual directory **jakarta**.

16 Double click the **Handler Mappings**.

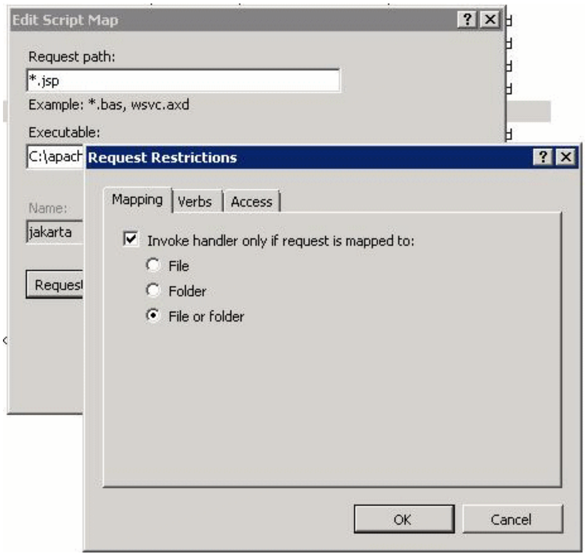
17 Add a script map in the **Actions** list with **Request path** as ***.jsp**, **Executable** pointing to **isapi_redirect.dll**, **Name** as **jakarta**.

18 Click **Request Restrictions**.

Check the following options:

- Open **Mapping** tab, check **Invoke handler only if request is mapped to:**, select **File or Folder**
- Open **Access** tab, select **Execute**.

  See the example below:



Edit Script Map

Request path: *.jsp

Example: *.bas, wsvc.axd

Executable: C:\apach

Name: jakarta

Request Restrictions

☑ Invoke handler only if request is mapped to:

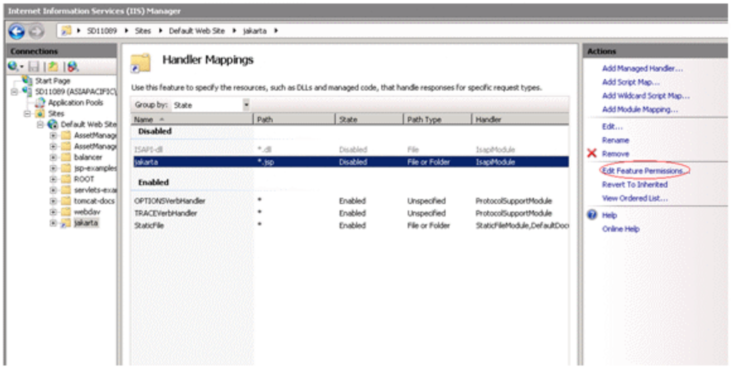
○ File

○ Folder

● File or folder

- Click **OK** button to close the window.

19 Select the new Jakarta handler mapping. In the righthand panel select **Edit Feature Permission** as shown in the example below.

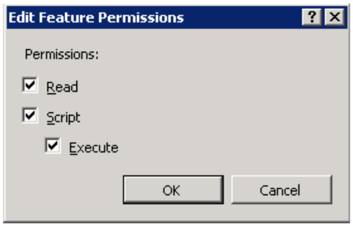


Use this feature to specify the resources, such as DLLs and managed code, that handle responses for specific request types.

Group by: State

Name	Path	State	Path Type	Handler
Disabled				
ISAPI-dll	*.dll	Disabled	File	IsapiModule
jakarta	*.jsp	Disabled	File or Folder	IsapiModule
Enabled				
OPTIONSVerbHandler	*	Enabled	Unspecified	ProtocolSupportModule
TRACEVerbHandler	*	Enabled	Unspecified	ProtocolSupportModule
StaticFile	*	Enabled	File or Folder	StaticFileModule,DefaultDoc

20 Select **Execute**, then click **OK**:



Permissions:

☑ Read

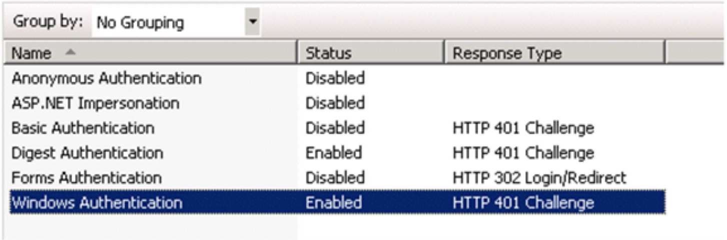
☑ Script

☑ Execute

21 Still in the **Default Web Site Properties** window, click **Authentication**.

22 Disable **Anonymous Authentication** and enable **Digest Authentication** and **Windows Authentication**.

See the example below:

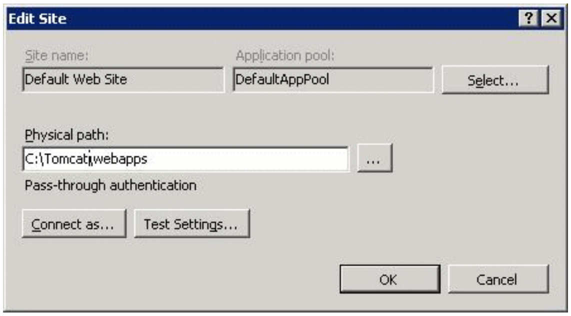


Group by: No Grouping

Name	Status	Response Type
Anonymous Authentication	Disabled	
ASP.NET Impersonation	Disabled	
Basic Authentication	Disabled	HTTP 401 Challenge
Digest Authentication	Enabled	HTTP 401 Challenge
Forms Authentication	Disabled	HTTP 302 Login/Redirect
Windows Authentication	Enabled	HTTP 401 Challenge

23. Navigate to Default Web Site Properties.
24. Select **Basic Settings** in the **Actions** list.
25. Set **Physical path** to **<Tomcat installation folder>\webapps**.

    See the example below.



Edit Site

Site name: Default Web Site

Application pool: DefaultAppPool

Physical path: C:\Tomcat\webapps

Pass-through authentication

26. Click the **OK** button.
27. Navigate to your host again and click **Restart** in the **Actions** list.

Configure the port transfer

1. Open `<Tomcat installation folder>\conf\server.xml` with a text editor and disable tomcat authentication as in the following example:

```
<Connector port="8009" enableLookups="false" redirectPort="8443" protoc
ol="AJP/1.3" tomcatAuthentication="false" />
```

2. To check that the port transfer works correctly, restart tomcat and then perform the following test:
3. Open a browser and navigate to following URL:

   http://localhost:8080/AssetManager/

> Tip:
>
> If it does not work, make sure that you have launched Tomcat.

4. Instruct Jakarta to manage the Tomcat default application and Asset Manager.

   To do this, open `<Tomcat installation folder>\conf\uriworkermap.properties` in a text editor and add the following line in the file:

```
/AssetManager/*=wlb
```

5. Restart Tomcat and IIS

6. If you are connected as a Windows user, you should be able to access the Asset Manager Web client without having to authenticate again.

Trouble Shooting

Here are some of the most common problems, and potential solutions to them:

- I can open Asset Manager with 8080 port, but failed to open with port 80 with "Page not found" error.

  ANSWER: First, make sure IIS service is up, then open the Jarkata file (`<Tomcat installation folder>\Jarkata\conf\uriworkermap.properties`), check if the Asset Manager application has been added.

- After SSO is enabled, Asset Manager login is still required.

  ANSWER: Make sure all the previous steps are correctly completed. Open the `<Tomcat installation folder>\conf\server.xml` file in write mode and remove the comment characters (<!-- and -->) for the following line:

  ```
  <Valve className="org.apache.catalina.valves.RequestDumperValve"/>
  ```

  This forces Tomcat to generate an additional log file named: `Catalina_log.YYYY-MM-DD.txt`

  Open this log file and check the entries for the following parameters:

  ```
  YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: queryString=null
  ```

  ```
  YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: remoteAddr=192.178.0.
  1 (type your IP address here)
  ```

  ```
  YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: remoteAddr=192.178.0.
  1 (type your IP address here)
  ```

  ```
  YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: remoteUser=<Domain na
  me>\<User name>
  ```

  ```
  YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: requestedSessionId=nu
  ll
  ```

  ```
  YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: scheme=http
  ```

  If the **remoteUser** field is empty, check Tomcat authentication again. Open the `<Tomcat installation folder>\conf\server.xml` file, make sure Tomcat authentication is disabled.

- I get the following error:

  ```
  Login failed with error "Error: com.peregrine.ac.AmException: Error (12
  ,008): Login 'xxx' unknown."
  ```

  ANSWER: Check if the user exists in database and user has authorized role to access Asset Manager. Pay attention to the user login name. To check if login name contains domain name, open the `<Tomcat installation folder>\webapps\AssetManager\WEB-INF\classes\application-context.html` file and check the **keepDomain** value. If you want to keep domain name, update this value to "true". Otherwise, update it as "false". Restart Tomcat and try again.

Asset Manager Web client: Implementing Single Sign-On (SSO) using identity management tools such as Siteminder or Webseal

▶ Asset Manager Web client: Sharing user authentication using Lightweight Single Sign-On (LW-SSO) [page 220]

Asset Manager Web client: Sharing user authentication using Lightweight Single Sign-On (LW-SSO)

Overview

When the Lightweight Single Sign-On (LW-SSO) framework is enabled, a user only needs to login once (to a single HP Software BTO application) during a session. The user's login information will be propagated to to other HP Software BTO applications deployed on same domain.

LW-SSO solves both the Web and Web Services SSO problems for HP Software BTO products, using lightweight technology, and without having to install any additional infrastructure.

LW-SSO use cases

LW-SSO can be used to address the following use cases:

1. **Web Single Sign-On:** A user logs in one BTO application using his credentials and then wants to access the second BTO application in the same browser. Since the user has already logged in, LW-SSO enables the user to enter the second application without requiring him to interactively re-authenticate his credentials.
2. **Web Services Single Sign On:** During a user request to one BTO application, a call must be executed to the second BTO application. LW-SSO transfers User Data to the second BTO application.
3. **Web Service calls:** A Web Service call must be executed from the first BTO application to the second BTO application using information from the non-interactive user. LW-SSO transfers user data to the second BTO application.
4. **LW-SSO is used as an abstraction layer** between BTO applications and external Identity Management (IdM) / SSO tools. This facilitates product integration with the IdM / SSO tool and overall developmental costs.

Key features

- LW-SSO does **not** provide or mediate user authentication as such; each application is responsible for authenticating the user.
- LW-SSO is responsible only for passing user information from Application1 to Application2 (via shared cookies).
- Thus LW-SSO is not a replacement for other IdM solutions.
- BTO Application1 and BTO Application2 can use either the same user storage or separate user storages. However, if using separate user storages, they need to be able to synch user names, which is not handled in the LW-SSO framework.
- LW-SSO is symmetric solution: user can first login in Application1 and than transparently open a link to Application2, or vise versa: first login in Application2 and then open a link to Application1.

Configuring LW-SSO

Configuring on the fly

The Asset Manager Web Service deployment includes an administration screen to configure basic LW-SSO parameters.

Connect to the following URL:

http://${AMWebServiceServer}:${AMWebServicePort}/{AssetManagerWebServiceContext}/ssoConfig.jsp

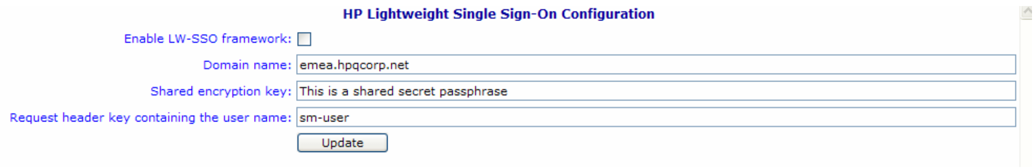
Example:

http://localhost:8081/AssetManagerWebService/ssoConfig.jsp

Note:

Your application server e.g. Tomcat must be correctly configured and running.

The sign-on screen displays, as in the following example:



HP Lightweight Single Sign-On Configuration

Enable LW-SSO framework: ☐

Domain name: emea.hpqcorp.net

Shared encryption key: This is a shared secret passphrase

Request header key containing the user name: sm-user

Populate the fields as described in the table below.

Note:

This screen requires administrative rights.

Field / control name	Usage
Enable LW-SSO framework	When this option is checked, the LW-SSO framework is enabled on the target domain, defined in the next field.
Domain name	Domain on which user login information is propagated between hosted HP Software BTO applications which subscribe to a compatible version of LW-SSO. This must be the full DNS name of the application server. For instance: `asiapacific.mycompany.net`
Shared encryption key	This is a secret text string that is used as the encryption key for LW-SSO cookies. This is used to populate the **initstring** parameter. Other HP Software BTO applications which subscribe to LW-SSO must be configured to use the same initstring to be able to decrypt and share common cookies. See also LW-SSO Security Considerations [page 222]
Request header key containing the user name	HTTP header responses must make reference to the specified user name. (This can be used to support SSO solutions such as for Siteminder, etc). As the label suggests, you need only to specify the name of the header key.

Additional configuration can be achieved by making manual changes to the LW-SSO API.

Disable shared cookies

When LW-SSO is enabled, by default logging out from one application that subscribes to LW-SSO will trigger a logout from all other subscribing HP Software BTO applications hosted on the same domain. To prevent this, thus requiring an explicit logout per application, add the following entry to the file `lwssofmconf.xml` located in **AssetManager.ear/AssetManager.war/WEB-INF/classes**.

```
<logoutURLs>
<url>
.*cwc/logoutcleanup.jsp.*
</url>
</logoutURLs>
```

LW-SSO Security Considerations

> Note:
>
> LW-SSO should be disabled unless it is specifically required

Confidential **initString** parameter in LW-SSO security

LW-SSO uses Symmetric Encryption to validate and create LW-SSO tokens. The **initString** or secret phrase configuration parameter is used to initialize the encryption key. An application creates a token, and each application that uses the same **initString** parameter validates the token.

Note:

Important Notes regarding the **initString**:

- It is not possible to use LW-SSO without setting the **initString** parameter.
- The **initString** parameter is confidential information and should be treated as such in terms of publishing, transporting, and persistency.
- The **initString** should be shared only between applications integrating with each other using LW-SSO.
- The minimum length of the **initString** is 12 characters.

How authentication security level is determined

The Application that uses the weakest authentication framework and issues a LW-SSO token that is trusted by other integrated applications, determines the level of authentication security for all the applications.

It is recommended that only applications using strong and secure authentication frameworks issue an LW-SSO token.

Symmetric encryption implication

LW-SSO uses symmetric cryptography for issuing and validating LW-SSO Tokens.

Therefore, any application using LW-SSO can issue a token to be trusted by all other applications sharing the same **initString**.

This potential risk is relevant when an application sharing the **initString** either resides or is accessible in an untrusted location.

User mapping (Synchronization)

LW-SSO does not ensure user mapping between the integrated applications. Therefore, the integrated application must monitor user mapping. It is recommended that you share the same user registry (as LDAP/AD) among all integrated applications.

Failure to map users may cause security breaches and negative application behavior. For example, the same user name may be assigned to different real users in the various applications.

In addition, in cases where a user logs onto an application (AppA) and then accesses a second application (AppB) that uses container or application authentication, the failure to map the user will force him/her to manually log onto AppB and enter a username. If the user enters a different username than he/she used to log onto AppA, the following unexpected behavior can arise: If the user subsequently accesses a third application (AppC) from AppA or AppB, then he/she will access it using the usernames that were used to log onto AppA or AppB respectively.

Identity Manager is used for authentication

All unprotected resources in the Identity Manager must be listed as **nonsecureURLs** in the LW-SSO configuration.

LW-SSO Limitations

Domain Names

To benefit from LW-SSO, a user must access their BTO applications such as Asset Manager using the Fully Qualified Domain Name (FQDN) in the login URL, e.g. http://fhedaraly.emea.hpqcorp.net:8080/AssetManager

LW-SSO does not support URLs with an IP Address, e.g. http://16.59.45.143:8080/WebApp

LW-SSO does not support URLs without a domain, e.g. http://flood:8080/WebApp

LW-SSO framework integration

Applications can leverage and use LW-SSO capabilities only if integrated within the LW-SSO framework in advance.

Multi-domain support limitations

- Multi-domain functionality is based on the HTTP referrer. Therefore, LW-SSO supports links from one application to another and does not support typing a URL into a browser window, except when both applications are in the same domain.
- Cross-domain linking using HTTP POST is not supported:

  Multi-domain functionality does not support the first HTTP POST request to a second application (only the HTTP GET request is supported). For example, if your application has an HTTP link to a second application, an HTTP GET request is supported, but an HTTP FORM request is not supported. All requests after the first can be either HTTP POST or HTTP GET.
- LW-SSO Token size limitation:

  The size of information that LW-SSO can transfer from one application in one domain to another application on another domain is limited to 15 Groups/Roles/Attributes (note that each item may be an average of 15 characters long).
- Linking from Protected (HTTPS) to non protected (HTTP) in a Multi domain scenario:

  Multi-domain functionality does not work when linking from a protected (HTTPS) to a non protected (HTTP) page. This is a browser limitation where the referrer header is not sent when linking from a protected to a non-protected resource. For an example, see: http://support.microsoft.com/support/kb/articles/Q178/0/66.ASP

- "Third-Party" cookies in Internet Explorer:

  In IE6, Microsoft added a module that supports the "Platform for Privacy Preferences (P3P) Project", which means that cookies coming from a "Third Party" domain are by default blocked in the "Internet" security zone. This means that session cookies are considered "Third party" cookies by IE as well, and therefore are blocked, causing LW-SSO to stop working.

  For details, see: http://support.microsoft.com/kb/323752/en-us

  > Note:
  >
  > A possible solution is to add the launched application (or a DNS domain subset as *.mydomain.com) to the "Intranet"/"Trusted" zone on your computer (using the Internet Explorer menu option**Tools/ Internet Options/ Security/ Local Intranet/ Sites/ Advanced**), which causes the cookies to be accepted.

  > Warning:
  >
  > Important: The LW-SSO session cookie is only one of the "Third party" application cookies that is blocked by this parameter.

SAML2 token related limitations

When using a Java based application integrated with LW-SSO to access another Java based application also integrated with LW-SSO, do not use the SAML2 token. Using the SAML2 token in this case can lead to unexpected behavior. Use the LW-SSO token instead.

The SAML2 token should only be used when one of the applications is not integrated with LW-SSO.

Logout functionality is not supported when the SAML2 token is used. Therefore, if the SAML2 token is used to access a second application, then a user who logs out in the first application will not be logged out in the second application.

The SAML2 token's expiration is not reflected in the application's session management.

Therefore, if the SAML2 token is used to access a second application, then each application's session management will be handled independently of each other. When your application uses Container Managed Security in Tomcat or JBoss servers, it is recommended that you enable the **AutoCookieCreation** feature for proper SAML2 support. If **AutoCookieCreation** is disabled, you must comply with the following rules, otherwise LW-SSO will not be able to properly authenticate requests with SAML2.

- The first resource (for example, a JSP page) should create an HTTP session.
- The first resource should not include any protected resources (for example, JPEG images)

- The HTTP session should not be erased until user closes browser.

Tomcat-related limitations

- JAAS Realm in Tomcat is not supported
- Using spaces in Tomcat directories is not supported.

  It is not possible to use LW-SSO when a Tomcat installation path (folders) includes spaces (e.g. Program Files) and the LW-SSO configuration file is located in the common\classes Tomcat folder.

Load balancer configuration

A load balancer deployed with LW-SSO must be configured to use sticky sessions.

LW-SSO Important Notes

LW-SSO Token expiration

The LW-SSO Token's expiration value determines the application's session validity. Therefore, its expiration value should be at least the same value as that of the application session expiration value.

Recommended configuration of the LW-SSO Token expiration

Each application using LW-SSO should configure token expiration. The recommended value is 60 minutes. For an application that does not require a high level of security, it is possible to configure a value of 300 minutes.

GMT time synchronization

All applications participate in an LW-SSO integration must use the same GMT time with a maximum difference of 15 minutes.

protectedDomains settings

Multi domain functionality requires that all applications participating in LW-SSO integration configure the protectedDomains settings, if they are required to integrate with applications in different DNS domains. In addition, they must also add the correct domain in the lwsso element of the configuration.

Get SecurityToken for URL functionality

To receive information sent as SecurityToken for URL from other applications, the host application should configure the correct domain in the lwsso element of the configuration.

nonsecureURLs setting

We recommend that you do not configure the application's Login and Logout URL's as the **nonsecureURLs** setting. Configuring them in this way may lead to unexpected behavior

LW-SSO Known Problems

IdM security and outbound web services

Applications that are integrated with LW-SSO, that use IdM security as the security token for outbound web service security, will not be able to create or issue LW-SSO security tokens for UI (either automatically created or else).

In this case, the LW-SSO configuration must include the following values:

```
enableAutoCreation = "false"
```

```
enableCookieCreation = "false"
```

No support for mulitple attribute values

The LW-SSO security context supports only one attribute value per attribute name. Therefore, when the SAML2 token sends more than one value for the same attribute name, only one value is accepted by the LW-SSO framework.

Similarly, if the IdM token is configured to send more than one value for the same attribute name, only one value is accepted by the LW-SSO framework.

Multi-domain logout functionality with Internet Explorer 7

Multi-domain logout functionality may fail when using Internet Explorer 7 and when the application is invoking more then 3 consecutive HTTP 302 redirect verbs in the logout procedure. In such a scenario Internet Explorer 7 may mishandle the HTTP 302 redirect response and display an "Internet Explorer cannot display the webpage" error page instead.

As a workaround, we recommend that, if possible, you reduce the number of application redirect commands in the logout sequence.

LW-SSO System Requirements

The following versions and configurations of browsers and related technology are supported by LW-SSO version 2.1.1

Name	Version	Comments
Java	1.5 and higher	
HTTP Servlets API	2.1 and higher	
Internet Explorer	6.0 and higher	Browser should enable HTTP session cookie and HTTP 302 Redirect functionality

Name	Version	Comments
FireFox	2.0 and higher	Browser should enable HTTP session cookie and HTTP 302 Redirect functionality
JBoss authentications	JBoss 4.0.3 JBoss 4.3.0	
Tomcat authentications	Standalone Tomcat 5.0.28 Standalone Tomcat 5.5.20	
Acegi authentications	Acegi 0.9.0 Acegi 1.0.4	
Web Services engines	Axis 1 - 1.4 Axis 2 - 1.2 JAX-WS-RI 2.1.1	

Troubleshooting

The following tables list possible problems associated with LW-SSO, together with possible causes and suggested solutions.

LW-SSO related use cases

Problem	Possible Cause	Possible Solution
Impossible to open web client of a BTO application, for instance after deploying additional web applications to Tomcat.	Tomcat repertory has not been correctly refreshed.	Example: delete `C:\Tomcat55\webapps\AssetManager` This folder will be dynamically recreated when you next attempt to access the web client. Manually delete the folder under `Tomcat<xx>\webapps` that corresponds to the application that does not open.

Problem	Possible Cause	Possible Solution
LW-SSO cookie is not created after logging in to a BTO web application.	A domain is not defined properly in the LW-SSO element of the configuration.	Ensure that the domain defined in the LW-SSO element of the configuration is equal to the application's domain.
	A domain that is passed as a parameter to the **enableSSO** function is not correct.	Make sure that the domain that is passed as a parameter to the enableSSO function is equal to the application's domain.
	You did not access the application with the Fully Qualified Domain Name (FQDN) in the login URL (for example: http://localhost:8080/WebApp)	Make sure that you access the application with the Fully Qualified Domain Name (FQDN) in the login URL (for example: http://flood.mercury.global:8080/WebApp).
LW-SSO fails to create a cookie for AutoCookieCreation functionality.	A domain is not defined properly in the LW-SSO element of the configuration.	Make sure that the domain defined in the LW-SSO element of the configuration is equal to the application's domain.
The LW-SSO token is not validated.	The two applications have different initString parameters in the crypto element of the configuration (or other crypto parameters).	Use the same initString in both applications (in addition to all other crypto parameters in the LW-SSO creation element).
	The GMT time difference between the two applications is greater than 15 minutes.	Make sure that all applications participating in a LW - SSO integration are set to the same GMT time, with a maximum difference of 15 minutes.

Problem	Possible Cause	Possible Solution
LW-SSO fails to validate the LW-SSO token in a multi-domain environment.	In the configuration of one of the applications, a domain is not defined properly in the LW-SSO element.	The domain defined in the LW-SSO element of the application's configuration must be the same as the application's domain according to real domains in use.
	In the configuration of one of the applications, a domain is not defined correctly in the protectdDomains list.	Make sure that the domains in the protectedDomains list of all of the applications' configurations are defined correctly.
	The LW-SSO session cookie is blocked/denied when using the Internet Explorer 6.0, 7.0 or 8.0 browsers.	Add all LW-SSO servers to the "Intranet"/"Trusted" zone in the Internet Explorer security zones on your computer (**Tools/ Internet Options/ Security/ Local Intranet/ Sites/ Advanced**). This will allow all cookies to be accepted.
	Some applications have different **initString** parameters in the crypto element of the configuration (or other crypto parameters).	Use the same **initString** in all applications (in addition to all other crypto parameters in the LW-SSO creation element).
	Some applications have a GMT time difference greater than 15 minutes.	Make sure that all applications participating in a LW-SSO integration are set to the same GMT time with a maximum difference of 15 minutes.
	Multi domain link goes from the protected (HTTPS) to the Non protected (HTTP) resource.	When linking/crossing from one domain to another, make sure that the first link/cross request goes from one protected resource (HTTPS) to another protected resource (HTTPS).

SAML2 related use cases

Problem	Possible Cause	Possible Solution
LW-SSO fails to issue a SAML2 Token	A keystore configuration for SAML2 creation is not valid and most likely does not point to a valid keystore.	Make sure that the keystore configuration for SAML2 creation is valid and points to a valid keystore.

Problem	Possible Cause	Possible Solution
	A privateKeyAlias or privateKeyPassword configuration for SAML2 creation is not valid. They might not be pointing to a valid private key.	Make sure that the privateKeyAlias or privateKeyPassword configuration for SAML2 creation is valid and point to a valid private key.
LW-SSO fails to validate the SAML2 token.	A keystore configuration for SAML2 validation is not valid. It most likely does not point to a valid keystore.	Make sure that the keystore configuration for SAML2 validation is valid and points to a valid keystore.
	A certificate used for signing is not imported in the configured keystore with a required public key alias. **Note:** The public key alias may be configured in the SAML2 validation configuration or it can be passed in SAML2 token by the issuer.	Make sure that the certificate used for signing in is imported in the configured keystore with a required public key alias.
LW-SSO fails to receive all roles or groups.	The element's **roleAttributeName** or **groupAttributeName** defined in the SAML2 configuration is different to the one configured in the applications' configurations element.	Make sure that the elements **roleAttributeName** or **groupAttributeName** is defined equally in all of the applications' configurations as in the SAML2 configuration.

13 Asset Manager Automated Process Manager

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 Asset Manager Automated Process Manager program, which is independent of Asset Manager.

Overview of Asset Manager Automated Process Manager

Asset Manager includes a system for monitoring deadlines and automatically triggering actions: This program, called Asset Manager Automated Process Manager, functions independently of Asset Manager.

Asset Manager Automated Process Manager automatically monitors all expiration dates in the designated database:

- Alarms (end of term dates of contracts for example).
- Deadlines specific to helpdesk (for example, ticket assignment and resolution time limits).
- 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, Asset Manager Automated Process Manager performs actions, such as issuing reminder messages in the Asset Manager database via the internal messaging system. If necessary, it calculates contract rent, lease contract loss-values, etc.

Warning:

When you exit Asset Manager Automated Process Manager, all automatic monitoring functions are suspended if they have not been launched as services.

It is possible to launch Asset Manager Automated Process Manager on several different machines. Therefore, deadlines to monitor and tasks to perform can be shared by different instances of Asset Manager Automated Process Manager. This enables you to improve the performance of Asset Manager Automated Process Manager.

Note:

You must make sure that a given task is only performed by one single instance of Asset Manager Automated Process Manager.

You can use the same Login to connect to the database. This login must have administration rights.

Executing Asset Manager Automated Process Manager

Recommendations

Asset Manager Automated Process Manager 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 Asset Manager Automated Process Manager 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 Asset Manager Automated Process Manager directly on the server.

In case of modification of the structure of the database

If you modify the structure of the database via Asset Manager Application Designer or via the **Configure object** command in the shortcut menu, you must disconnect Asset Manager Automated Process Manager from the database and then reconnect.

Launching Asset Manager Automated Process Manager

Manually launching Asset Manager Automated Process Manager

You can launch the Asset Manager Automated Process Manager program from the programs in the **Start** menu or in the Asset Manager program group.

Warning:

When you disconnect Asset Manager Automated Process Manager from the database, all monitoring functions and automatic triggering of action are suspended. When you connect to the database in Asset Manager, Asset Manager displays a warning message indicating that Asset Manager Automated Process Manager has not accessed the database in the last hour.

Automatically launching Asset Manager Automated Process Manager as a service

To launch Asset Manager Automated Process Manager as a service:

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

From the DOS command prompt

You can automate the launching of Asset Manager Automated Process Manager by using the following command:

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

Warning:

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

Strings between <> should have no spaces.

Example:

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

This command can be placed in a batch file.

Executing Asset Manager Automated Process Manager manually in Windows

Connecting Asset Manager Automated Process Manager to a database

Only an administrator can connect to the database via Asset Manager Automated Process Manager. This can be the "Admin" user or a user with administration rights.

You need to enter the relevant **Login** and password. Select the **Use this connection in service mode** check box if you want to use this connection as the default connection if Asset Manager Automated Process Manager is running in service mode (as in an NT service).

To connect Asset Manager Automated Process Manager to the database, use one of the following:

- The **File/ Connect to database** menu.
- The icon.

Disconnecting Asset Manager Automated Process Manager from a database

To disconnect Asset Manager Automated Process Manager from the database, use one of the following:

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

Executing Asset Manager Automated Process Manager on Unix

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

- `-webadmin` to administer the Asset Manager Automated Process Manager 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: ▶ **Asset Manager - Installation and upgrade** guide, chapter **.ini and .cfg files**.

Refer to the **Installation and upgrade** guide for information concerning the compatibility and installation of Asset Manager on Unix.

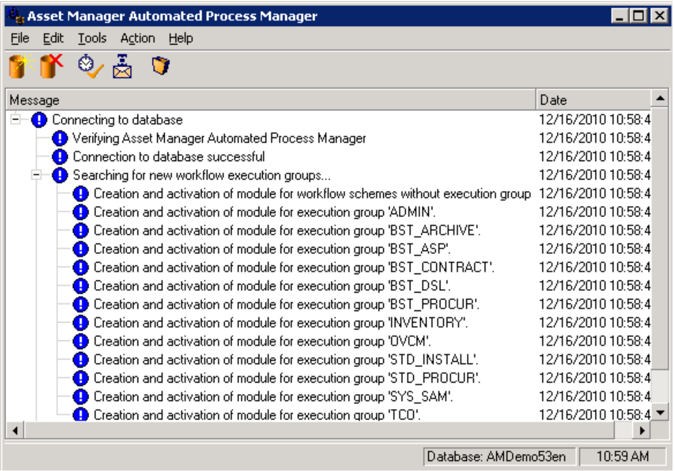
If you exit Asset Manager Automated Process Manager

Nevertheless, if Asset Manager Automated Process Manager 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.

Asset Manager Automated Process Manager and the messaging system

The computers on which Asset Manager Automated Process Manager 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 Asset Manager Automated Process Manager



```
Connecting to database                                                          12/16/2010 10:58:4
  Verifying Asset Manager Automated Process Manager                             12/16/2010 10:58:4
  Connection to database successful                                             12/16/2010 10:58:4
  Searching for new workflow execution groups...                                12/16/2010 10:58:4
    Creation and activation of module for workflow schemes without execution group 12/16/2010 10:58:4
    Creation and activation of module for execution group 'ADMIN'.              12/16/2010 10:58:4
    Creation and activation of module for execution group 'BST_ARCHIVE'.        12/16/2010 10:58:4
    Creation and activation of module for execution group 'BST_ASP'.            12/16/2010 10:58:4
    Creation and activation of module for execution group 'BST_CONTRACT'.       12/16/2010 10:58:4
    Creation and activation of module for execution group 'BST_DSL'.            12/16/2010 10:58:4
    Creation and activation of module for execution group 'BST_PROCUR'.         12/16/2010 10:58:4
    Creation and activation of module for execution group 'INVENTORY'.          12/16/2010 10:58:4
    Creation and activation of module for execution group 'OVCM'.               12/16/2010 10:58:4
    Creation and activation of module for execution group 'STD_INSTALL'.        12/16/2010 10:58:4
    Creation and activation of module for execution group 'STD_PROCUR'.         12/16/2010 10:58:4
    Creation and activation of module for execution group 'SYS_SAM'.            12/16/2010 10:58:4
    Creation and activation of module for execution group 'TCO'.                12/16/2010 10:58:4
```

The main window displays all events handled by Asset Manager Automated Process Manager.

It enables you to access the following program menus and icons:

Table 13.1. Asset Manager Automated Process Manager program menus and icons

Icon	Menu	Function
	File/ Connect to database	Connect to a database.
	File/ Disconnect from database	Disconnect from the database.
	Action/ Activate	Selectively activate deadline monitoring agents.
	Action/ Empty list	Clear messages from main screen.
	Action/ Connect to messaging	Try to connect to external messaging systems. This button is grayed out if the connection is successful. If the connection fails, this button is available. Click to test the connection.

Whenever the list of events becomes too large, you can use the **Action/ Empty list** command to empty it.

General options of Asset Manager Automated Process Manager

To access the main monitoring options, select the **Tools/ Main options...** menu.

These govern the general operation of Asset Manager Automated Process Manager.

Reconnection delays frame, **Database** frame

This field is used if Asset Manager Automated Process Manager receives an error message when trying to connect to the database after the initial connection has been made successfully.

In this case, Asset Manager Automated Process Manager considers that the connection to the database has been lost and stops monitoring deadlines. Asset Manager Automated Process Manager 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 Asset Manager Automated Process Manager manages to reconnect to the database.

The data-entry format for this field is "Duration".

Reconnection delays frame, **Messaging** frame

If a problem occurs with the external messaging system, Asset Manager Automated Process Manager will stop sending external messages.

Asset Manager Automated Process Manager 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 Asset Manager Automated Process Manager.

Max. size

This field enables you to limit the size of the file that logs the messages shown in the main screen of Asset Manager Automated Process Manager.

When this limit is reached, the oldest messages are deleted as and when new messages are recorded.

Time zones

In the **Time zones** frame of the configuration screen, you configure the type of test to be performed:

- **Verify time zone of database server**.
- **Verify local time compared to that of the server**.

These two tests compare the time of the database server with that of the machine on which Asset Manager Automated Process Manager is installed. The time difference is expressed as [(n * 30minutes) + m] where m is between -15 minutes and + 15 minutes.

In both cases

If the time difference exceeds 5 minutes, Asset Manager Automated Process Manager 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 Asset Manager Automated Process Manager, 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 Asset Manager is updated (if the number (n * 30 minutes) does not correspond to the time zone of the server).

To do this, the machine on which Asset Manager Automated Process Manager 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 Asset Manager, is recovered.

Note:

Whichever option you select, the tests are performed when Asset Manager Automated Process Manager connects to the database, then regularly according to the frequency defined in Asset Manager Automated Process Manager's **Configuration of modules** window (**Tools/ Configure modules...** menu).

Configuring the modules monitored by Asset Manager Automated Process Manager

Introduction

To consult or configure a model monitored by Asset Manager Automated Process Manager, 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 Asset Manager Automated Process Manager 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 Asset Manager Automated Process Manager. These modules are identified by the description and the name that is listed in Asset Manager Automated Process Manager.

Asset Manager Automated Process Manager 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 13.2. Days of monitoring

Value of the **Days** field	Monitoring is performed
"Daily"	Every day of the year, without exception.
"Day of the week:"	The weekday selected from an itemized list in the field to the right of the **Days** field. **Example** Every Monday
"Day of the year:"	A day or selection of days. These are defined via the **Day**, **Month** and **Year** check boxes. **Example** July 20
"The first" "The second" "The third" "The fourth" "The next to last" "The last"	The days of the week defined using the **Day** check box, and for the month(s) and year(s) defined using the **Month** and **Year** check boxes. **Examples** "The first" Friday of each month. "The second" Monday of the month of September. "The next to last" Wednesday of the month of November. "The last" Tuesday of every month throughout 2000.

Monitoring times

Periodical

You can define two frequencies for the verification of a module that depend on the time of day.

The first **During period** frequency applies to scheduled periods that you can create in two ways:

- Graphically, using the schedule bar. Click and drag on this bar to create a schedule period.
- Directly, by entering an itemized list of values in the field to right of the schedule bar. The following syntax is used:

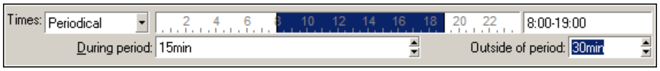
  ```
  <Start time of period> - <End time of period>
  ```

  The time format defined in Asset Manager 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:



Times: Periodical

8:00-19:00

During period: 15min

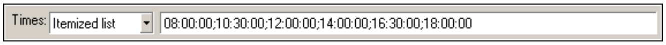
Outside of period: 30min

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 Asset Manager must be respected.
- The semi-colon ";" is used to separate them.

  Example:



Times: Itemized list

08:00:00;10:30:00;12:00:00;14:00:00;16:30:00;18:00:00

Preview

You can have a preview of:

- A rule defining the monitoring schedules of a module in the **Preview** field of a rule description sub-tab.
- All the rules concerning a module in the **Preview** sub-tab of the **Modules** tab, once the module is selected.

Add the computers listed in the NT domain to the database module (AddCpu)

Asset Manager Automated Process Manager enables you to program the recovery of those computers declared in the NT domain.

The domain to analyze is specified at the HP 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 Asset Manager Automated Process Manager module **Add the computers listed in the NT domain to the database**.
- Parameters of the HP Connect-It `addcpu.scn` scenario, which is located in the `scenario\ntsec\ntac93` sub-folder of the HP 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 -wpplog '$connectit_exedir$/../sce
nario/ntsec/ntac$version$/addcpu.scn'
-dc:AssetCenter.SERVER=$cnx$ -dc:AssetCenter.LOGIN=$login$ -dc: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 HP Connect-It's **Once** scheduler.).

  Do not modify this parameter; you define the programming at the level of the Asset Manager Automated Process Manager application.
- **-wpplog** is used to generate an execution log of the scenario in a format readable by Asset Manager Automated Process Manager.
- `$connectit_exedir$/../scenario/ntsec/ntac93/addcpu.scn` is the path to the HP Connect-It scenario to use.

  Modify this parameter if you want to use another HP Connect-It scenario.
- `-d:Asset Manager.SERVER=$cnx$ -d:Asset Manager.LOGIN=$login$ -d:Asset Manager.TEXTPASSWORD=$pwd$` stores the Asset Manager connection name opened by Asset Manager Automated Process Manager, as well as the login and password used for the connection.

  These parameters overwrite the values defined at the level of the Asset Manager connector in the **addcpu.scn** scenario.

  You have no need to modify these parameters.

Parameters of the `addcpu.scn` HP Connect-It scenario

To modify the `addcpu.scn` scenario:

1. Execute the HP Connect-It scenario editor.
2. Open the `addcpu.scn` scenario, which is located in the `scenario\ntsec\ntac93` sub-folder of the HP 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 Asset Manager Automated Process Manager is connected.
> - Those domains that are **trusted** by the one to which the Asset Manager Automated Process Manager is connected.

> Tip:
>
> To find out whether or not a computer has been recovered:
>
> 1. Execute the Windows explorer on the Asset Manager Automated Process Manager machine.
> 2. Display the network neighborhood.
> 3. The computers that you see are those from which Asset Manager Automated Process Manager could recover information.

To find out which information is populated in the Asset Manager 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 HP Connect-It **Connectors** guide, chapter **Application connectors**, section **NT Security connector**.

To learn how the **Asset Manager** connector works, refer to the HP Connect-It **Connectors** guide, chapter **Hewlett-Packard connectors**, section **Asset Manager connector**.

Add NT users to the database module (AddUser)

Asset Manager Automated Process Manager 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 Asset Manager database that uses integrated NT security.

The domain to analyze is specified at the HP 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 Asset Manager Automated Process Manager module **Add NT users to the database**.
- Parameters of the HP Connect-It `adduser.scn` scenario, which is located in the `scenario\ntsec\ntac93` sub-folder of the HP 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 -wpplog '$connectit_exedir$/../sce
nario/ntsec/ntac$version$/adduser.scn'
-dc:AssetCenter.SERVER=$cnx$ -dc:AssetCenter.LOGIN=$login$ -dc: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 HP Connect-It's **Once** scheduler.).

  Do not modify this parameter; you define the programming at the level of the Asset Manager Automated Process Manager application.
- **-wpplog** is used to generate an execution log of the scenario in a format readable by Asset Manager Automated Process Manager.
- `$connectit_exedir$/../scenario/ntsec/ntac93/adduser.scn` is the path to the HP Connect-It scenario to use.

  Modify this parameter if you want to use another HP Connect-It scenario.
- `-d:Asset Manager.SERVER=$cnx$ -d:Asset Manager.LOGIN=$login$ -d:Asset Manager.TEXTPASSWORD=$pwd$` stores the Asset Manager connection name opened by Asset Manager Automated Process Manager, as well as the login and password used for the connection.

  These parameters overwrite the values defined at the level of the Asset Manager connector in the **adduser.scn** scenario.

You have no need to modify these parameters.

Parameters of the `adduser.scn` HP Connect-It scenario

To modify the `adduser.scn` scenario:

1. Execute the HP Connect-It scenario editor.
2. Open the `adduser.scn` scenario, which is located in the `scenario\ntsec\ntac93` sub-folder of the HP 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 HP Connect-It scenario and Asset Manager Automated Process Manager module per domain.

> Warning:
>
> The users that can be recovered are:
>
> - Those on the same domain as the one to which the Asset Manager Automated Process Manager user is connected.
> - Those domains that are **trusted** by the one to which the Asset Manager Automated Process Manager user is connected.

> Tip:
>
> To find out if a machine can be recovered:
>
> 1. Execute the Windows explorer on the Asset Manager Automated Process Manager 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 Asset Manager Automated Process Manager could recover information.

To find out which information is populated in the Asset Manager database, examine in detail the `adduser.scn` scenario.

To learn how the **NT Security** connector works, refer to the HP Connect-It **Connectors** guide, chapter **Application connectors**, section **NT Security connector**.

To learn how the **Asset Manager** connector works, refer to the HP Connect-It **Connectors** guide, chapter **Hewlett-Packard connectors**, section **Asset Manager 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

Asset Manager Automated Process Manager 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, Asset Manager Automated Process Manager 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, Asset Manager Automated Process Manager 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, Asset Manager Automated Process Manager modifies the **Recalculation effective from** field, which is found in the rent sub-tabs of the **Rents** tab of the contract detail.

Asset Manager Automated Process Manager 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

Asset Manager Automated Process Manager 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 set the number of days for which the calculation is made. For example, if you want to calculate rent over a period of 90 days, enter the following value:

```
90d
```

Note:

The maximum number of rent calculations made per transaction is specified by the `UserData` entry in the `Amsrv.cfg` configuration file.

Location of this file: ▶ **Asset Manager - Installation and upgrade** guide, chapter **.ini and .cfg files**.

Asset Manager Automated Process Manager 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.

- Asset Manager Automated Process Manager verifies rent payments every 2 months and generates projected rent payments for the next 12 months.

On July 1, 2005, Asset Manager Automated Process Manager 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.

Asset Manager Automated Process Manager 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)

Asset Manager Automated Process Manager recalculates, at regular intervals, the loss values for lease schedules whose calculation method is set to **Calculate for all periods** (**Calculation** field (SQL name: seLossValCalcMode)) on the **Leasing** tab of the lease schedule detail). In this way, loss values pertaining to any loss value rules which have been modified since the last time Asset Manager Automated Process Manager 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 Asset Manager.
- Select the **Administration/ Database options** menu.
- Select the **Procurement/ Let Asset Manager Automated Process Manager 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 Asset Manager Automated Process Manager rather than the Asset Manager 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.

Trigger helpdesk alarms module (HDAlarms)

Asset Manager Automated Process Manager also monitors deadlines specific to the helpdesk.

Note:

Deadlines take business calendar periods into account as specified in the **General** tab of the escalation scheme detail. When deadlines are calculated this result is converted into hours.

Alarms defined at escalation scheme level

Each helpdesk ticket is associated with an escalation scheme, indicated in the **Escalation** field (SQL name: EscalScheme) in the **Tracking** tab of the ticket detail.

The deadlines to be monitored by Asset Manager Automated Process Manager are defined in the **Alarms** tab of the escalation scheme detail. These deadlines correspond to certain stages in the processing of a ticket:

- **Time elapsed since opening ticket**: The date of opening of the ticket is specified in the **Opened on** field (SQL name: dtOpened) in the ticket detail.
- **% of resolution time**: This is a percentage of the duration specified in the **Duration** (SQL name: tsResolDelay) in the escalation scheme associated with the ticket.
- **Time before resolution deadline**: The ticket's time limit is indicated in the **Expected resol.** field (SQL name: dtResolLimit) in the **Tracking** tab of the ticket detail.

Note:

Asset Manager Automated Process Manager does not trigger the actions defined in the **Transitions** and **Assignment** tabs of an escalation scheme detail. They are triggered by an agent of Asset Manager.

Alarms defined at suspension slip level

Asset Manager Automated Process Manager monitors suspended Helpdesk tickets.

When the suspension time limit (in the **Suspend** tab of the suspension slip detail) is overrun, Asset Manager Automated Process Manager automatically triggers the action defined in the **Suspend** tab of the suspension slip detail.

Execute workflow rules for execution group (WkGroup*) modules

Once a workflow execution group (Example: **ADMIN**) is detected, Asset Manager Automated Process Manager executes the appropriate workflow rules.

Monitoring of workflow execution groups

Asset Manager Automated Process Manager monitors the deadlines specific to workflow instances associated with the execution group.

Deadlines to be monitored by Asset Manager Automated Process Manager 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, Asset Manager Automated Process Manager triggers the event if the activation conditions are met.

Then, the role of Asset Manager Automated Process Manager 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, Asset Manager Automated Process Manager saves it to the table with SQL name "amWfOccurEvent".

  Then, Asset Manager Automated Process Manager activates the transition according to the frequency of verification as defined in the configuration screen of Asset Manager Automated Process Manager.

- **Log event and process immediately**: As soon as the event occurs, Asset Manager Automated Process Manager saves it to the table with SQL name "amWfOccurEvent", and activates the transition.
- **Process event immediately without logging**: As soon as the event occurs, Asset Manager Automated Process Manager activates the transition.

Activation of transitions

Asset Manager Automated Process Manager 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

Asset Manager Automated Process Manager 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 Asset Manager Automated Process Manager verifies and performs the tasks it has to carry out is indicated in the configuration screen of Asset Manager Automated Process Manager.

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 Asset Manager Automated Process Manager if it is Asset Manager Automated Process Manager that activates the transition creating the task. In this case, Asset Manager Automated Process Manager performs the task as soon as the transition it creates is activated.
- Otherwise, the Asset Manager client machine executes the task.

Update statistics for tables module (Stats)

This module updates the database statistics.

These statistics are used by all the DBMSs supported by Asset Manager 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 Asset Manager application.

Purge the output 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 Asset Manager application.

Search for new workflow execution groups module (WorkflowFinder)

Asset Manager Automated Process Manager monitors the creation of new workflow execution groups.

As soon as Asset Manager Automated Process Manager 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 Asset Manager Automated Process Manager.

Signal presence of database server module (UpdateToken)

Asset Manager Automated Process Manager 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 Asset Manager Automated Process Manager for over one hour, a message is displayed when a user connects to the database in Asset Manager.

This message indicates that Asset Manager Automated Process Manager 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 Asset Manager Automated Process Manager for over a week, it is no longer possible to connect to the database.

Split expense lines in cost centers module (CostCenter)

Asset Manager Automated Process Manager handles split operations for expense lines.

Overview

Asset Manager Automated Process Manager 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).

Asset Manager Automated Process Manager 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, Asset Manager 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, Asset Manager gives you three possibilities:

- Delete all 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, Asset Manager deletes:

- The expense lines of the deleted cost center.
- The expense lines resulting from split operations on the deleted cost center.

An Asset Manager 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 Asset Manager Automated Process Manager 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, Asset Manager Automated Process Manager deletes the expense lines that, when split, generated the expense lines belonging to the deleted cost center.

Then Asset Manager Automated Process Manager 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 (navigation menu **Administration/ System/ 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

Asset Manager Automated Process Manager 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 Asset Manager.
2. Select **Asset lifecycle/Procurement lifecycle/ Requests/ 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

Asset Manager Automated Process Manager 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 (leasing)**, alarms can be attached to the end of term option notice dates. These dates are shown to the right of the **Buyout notice period**, **Renewal notice period** or **Return notice period** fields on the sub-tabs where the possible end of term options, **Renewal**, **Buyout**, **Return**, are described.
- If the contract **Type** (SQL name: seType) is **Lease schedule**, alarms can be attached to contract rent end dates in the **Schedule** frame of the **Rents** 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 on the **Financing** tab of the 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 Asset Manager'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 Asset Manager 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.

Asset Manager Automated Process Manager 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, Asset Manager Automated Process Manager stores the date and time of execution in the **Last execution of the rule** field (dtLastExec).

If an expense line cannot be generated, Asset Manager Automated Process Manager 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 Asset Manager 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. Asset Manager Automated Process Manager verifies if there are any such history lines; if it finds any it destroys them.

Verify stocks module (Stock)

Asset Manager Automated Process Manager monitors stock reorder levels.

For each stock, Asset Manager Automated Process Manager refers to the stock rules defined in the **Manage** tab of the stock detail.

For each stock rule concerning a model:

- Asset Manager Automated Process Manager 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, Asset Manager Automated Process Manager 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, Asset Manager Automated Process Manager 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, Asset Manager Automated Process Manager:
    - Readjusts the stock levels.
    - Erases the contents of the **Request line** field (SQL name: ReqLine) in the stock rule detail.
    - Reactivates the stock rule.

Update the database using Enterprise Discovery inventory results module (EdAc)

This module lets you program the import of the results of the inventory executed by the software distribution tool into the Asset Manager database.

Importing is done by HP Connect-It scenario `edac.scn`.

Before activating the **Update the database using Enterprise Discovery inventory results** module, you need to verify that the following parameters are correct:

- Parameters of the Asset Manager Automated Process Manager **Update the database using Enterprise Discovery inventory results** module.
- Parameters of the HP Connect-It `edac.scn` scenario, which is located in the `scenario\ed\ed<HP Discovery and Dependency Mapping Inventory version number>ac<Asset Manager version number>` sub-folder of the HP Connect-It installation folder.

Once these parameters are correct:

- Display the detail of the **Update the database using Enterprise Discovery inventory results** module.
- Select the **Enabled** option.
- Enter the options of the **Verification schedules** frame.

Parameters of the **Update the database using Enterprise Discovery inventory results** module (`User data item` field).

Here is the parameter line by default:

```
"$connectit_exedir$/conitsvc.exe" -once -wpplog '$connectit_exedir$/../sce
nario/ed/ed2ac$version$/edac.scn'
-dc:'Asset Management.SERVER'=$cnx$ -dc:'Asset Management.LOGIN'=$login$ -
dc:'Asset Management.TEXTPASSWORD'=$pwd$
```

Information about certain parameters of 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 HP Connect-It's **Once** scheduler).

  Do not modify this parameter; you define the programming at the level of the Asset Manager Automated Process Manager application.
- **-wpplog** is used to generate an execution log of the scenario in a format readable by Asset Manager Automated Process Manager.
- `$connectit_exedir$/../scenario/ed/ed2ac$version$/edac.scn` is the path to the HP Connect-It scenario to use.

  Modify this parameter if you want to use another HP Connect-It scenario.
- `-dc:'Asset Management.SERVER'=$cnx$ -dc:'Asset Management.LOGIN'=$login$ -dc:'Asset Management.TEXTPASSWORD'=$pwd$` stores the Asset Manager connection name opened by Asset Manager Automated Process Manager, as well as the login and password used for the connection.

  These parameters overwrite the values defined at the level of the Asset Manager connector in the **edac.scn** scenario.

  You do not need to modify these parameters.

Parameters of HP Connect-It scenario `edac.scn`.

To modify the `edac.scn` scenario:

1. Start the HP Connect-It scenario editor.
2. Open the `edac.scn` scenario, which is located in the `ed/ed2ac<Asset Manager version number >` sub-folder of the HP Connect-It installation folder.
3. Select the **Enterprise Discovery** connector in the **Scenario diagram** window by clicking the **Enterprise Discovery** title bar (but not anywhere in the box).
4. Select the **Tools/ Configure** menu.
5. Populate the pages in the wizard.

6 Click **Finish** to confirm.

To find out which information is populated in the Asset Manager database, examine in detail the `edac.scn` scenario.

Update mobile devices in the database using Enterprise Discovery inventory results (EdAcMD) module

This module lets you program the import into the Asset Manager database of the results of the mobile devices inventory executed by the software distribution tool.

The import is carried out by the `edac-mobiledevices.scn` HP Connect-It scenario.

Before activating the **Update mobile devices in the database using Enterprise Discovery inventory results** module, you need to verify that the following parameters are correct:

- Parameters of the Asset Manager Automated Process Manager **Update mobile devices in the database using Enterprise Discovery inventory results** module.
- Parameters of the HP Connect-It `edac-mobiledevices.scn` scenario, which is located in the `scenario\ed\ed<HP Discovery and Dependency Mapping Inventory version number>ac<Asset Manager version number>` sub-folder of the HP Connect-It installation folder.

Once these parameters are correct:

- Display the detail of the **Update mobile devices in the database using Enterprise Discovery inventory results** module.
- Select the **Enabled** option
- Enter the options in the **Verification schedules** frame.

Parameters of the **Update mobile devices in the database using Enterprise Discovery inventory results** module (`User data item` field).

The default command line parameters are as follows:

```
"$connectit_exedir$/conitsvc.exe" -once -wpplog '$connectit_exedir$/../sce
nario/ed/ed2ac$version$/edac-mobiledevices.scn'
-dc:'Asset Management.SERVER'=$cnx$ -dc:'Asset Management.LOGIN'=$login$ -
dc:'Asset Management.TEXTPASSWORD'=$pwd$
```

Information about certain parameters of 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 HP Connect-It's **Once** scheduler).

Do not modify this parameter. Scheduling information is defined in Asset Manager Automated Process Manager.

- **-wpplog** is used to generate an execution log of the scenario in a format which can be read by Asset Manager Automated Process Manager.
- `$connectit_exedir$/../scenario/ed/ed2ac$version$/edac-mobiledevices.scn` specifies the path of the HP Connect-It scenario to use.

  Modify this parameter if you want to use another HP Connect-It scenario.
- `-dc:'Asset Management.SERVER'=$cnx$ -dc:'Asset Management.LOGIN'=$login$ -dc:'Asset Management.TEXTPASSWORD'=$pwd$` stores the Asset Manager connection name opened by Asset Manager Automated Process Manager, as well as the login and password used for the connection.

  These parameters overwrite the values defined for the Asset Manager connector in the `edac-mobiledevices.scn` scenario.

  You do not need to modify these parameters.

Parameters of the `edac-mobiledevices.scn` HP Connect-It scenario

To modify the `edac-mobiledevices.scn` scenario:

1. Start the HP Connect-It scenario editor.
2. Open the `edac-mobiledevices.scn` scenario, which is located in the `ed/ed2ac<Asset Manager version number>` sub-folder of the HP Connect-It installation folder.
3. Select the **Enterprise Discovery** connector in the **Scenario diagram** window by clicking the **Enterprise Discovery** title bar (do not click anywhere else in the box).
4. Select the **Tools/ Configure** menu.
5. Populate the pages in the wizard.
6. Click **Finish** to validate your input.

To find out which information is populated in the Asset Manager database, examine in detail the `edac-mobiledevices.scn` scenario.

Immediately activate the verification of Asset Manager Automated Process Manager

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 Asset Manager Automated Process Manager.

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

Administering Asset Manager Automated Process Manager via the Web

The installation program of Asset Manager in Windows installs Asset Manager Automated Process Manager as an NT service.

Thus, you have the choice between launching:

- The graphical interface of Asset Manager Automated Process Manager.
- Or Asset Manager Automated Process Manager as an NT service.

If you launch Asset Manager Automated Process Manager as an NT service, you can control its functioning via the Web.

This section deals with:

- Starting Asset Manager Automated Process Manager as an NT service.
- Accessing the Asset Manager Automated Process Manager service via the Web.
- Driving the Asset Manager Automated Process Manager service via the Web.

Starting Asset Manager Automated Process Manager as a service

During the installation of Asset Manager:

- Asset Manager Automated Process Manager is installed as a service that is not launched.
- Web access to the Asset Manager Automated Process Manager service is not enabled.

Note:

To correctly install the Asset Manager Automated Process Manager 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 Asset Manager Automated Process Manager 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 Asset Manager Automated Process Manager service is installed.

   This is a reminder that the local system account only accesses by default the system environment variables.
2. Install the Asset Manager Automated Process Manager in this account.

Enabling Web access to the Asset Manager Automated Process Manager 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: ▸ **Asset Manager - Installation and upgrade** 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 Asset Manager Automated Process Manager 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 Asset Manager Automated Process Manager service

To start the Asset Manager Automated Process Manager NT service:

1. In Control Panel, click the **Services** utility.
2. Select the Asset Manager Automated Process Manager service.

Then, if you want to start the service straightaway:

- Click **Start**. In the case of the Asset Manager Automated Process Manager service, we advise against specifying any parameters directly in the **Startup Parameters** field.

If you want to configure the Asset Manager Automated Process Manager 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 Asset Manager Automated Process Manager service via the Web

Note:

Warning: In order to be able to access the Asset Manager Automated Process Manager service via the Web, the service must be started.

Then, to access the Asset Manager Automated Process Manager service:

1. Launch your usual Web browser.
2. Enter the address of the computer on which the Asset Manager Automated Process Manager service is running, followed by the TCP/IP port number used by the Asset Manager Automated Process Manager 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 Asset Manager Automated Process Manager 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 Asset Manager Automated Process Manager 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 Asset Manager Automated Process Manager service via the Web

This section describes the commands you have at your disposal once connected to the Asset Manager Automated Process Manager service.

Connect to new database

Using this command, you can:

- Connect manually to an Asset Manager database.
- Configure Asset Manager Automated Process Manager service to connect automatically to an Asset Manager database at startup. To do this:
  1. Check the **Reconnect at startup** option.

2. Enter the name of the database connection to which the Asset Manager Automated Process Manager service must reconnect automatically.
3. Specify the **Login** name and password.

> Note:
>
> You can also configure the automatic connection of the Asset Manager Automated Process Manager 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: ▶ **Asset Manager - Installation and upgrade** guide, chapter **.ini and .cfg files**.

Status of the server

This command displays the last 100 messages from the Asset Manager Automated Process Manager log file. These messages are similar to those shown in the main window in the GUI version of Asset Manager Automated Process Manager.

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 Asset Manager Automated Process Manager'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 Asset Manager Automated Process Manager 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: ▶ **Asset Manager - Installation and upgrade** guide, chapter **.ini and .cfg files**.

---

14 Messaging

Asset Manager gives you the possibility of managing two types of messages:

- Messages issued from Asset Manager and sent to the Asset Manager database via its internal messaging system.
- Messages created in Asset Manager and sent via an external messaging system.

This chapter details the configuration required to use the messaging system depending on the protocol used.

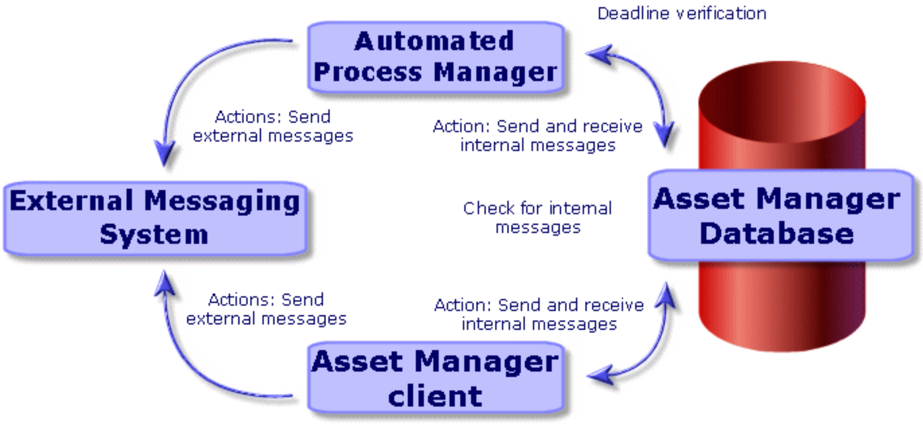
Overview of messaging

Asset Manager manages message-sending by using the following protocol types:

- AM (Asset Manager)
- SMTP
- MAPI
- VIM

When receiving messages, Asset Manager only manages those messages using the AM (Asset Manager) protocol.

Figure 14.1. Messaging system - overview


Deadline verification
Automated Process Manager
Actions: Send external messages
Action: Send and receive internal messages
External Messaging System
Check for internal messages
Asset Manager Database
Actions: Send external messages
Action: Send and receive internal messages
Asset Manager client

Refer to the **Advanced use** guide, section **Messaging**, for information about how to send, consult and receive messages.

Configuring Asset Manager to use messaging systems

The configuration of Asset Manager depends on the type of protocols used.

In order for Asset Manager and Asset Manager Automated Process Manager 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 Asset Manager Automated Process Manager.
- Configure the frequency of testing for new messages.

  Using the **Edit/ Options** menu, **Messages** tab.

Warning:

When you use Asset Manager, you cannot use another messaging protocol other than the one already used.

Only the internal Asset Manager (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@hp.com, SMTP:administrator@prgn.com
```

SMTP protocol

Detail of the user

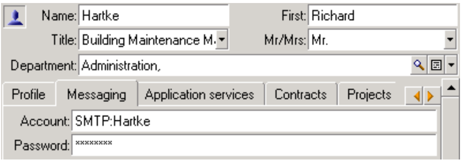
In order to send a message, Asset Manager needs to know the account of the sender, specified in the **Employees and departments** (amEmplDept) table (**Messaging** tab), and the messaging address of the recipient, specified in an action.

Configuring the sender

- The account of the sender is identified by the following fields:
  - **Account** (SQL name: MailLogin): this must be in the form:

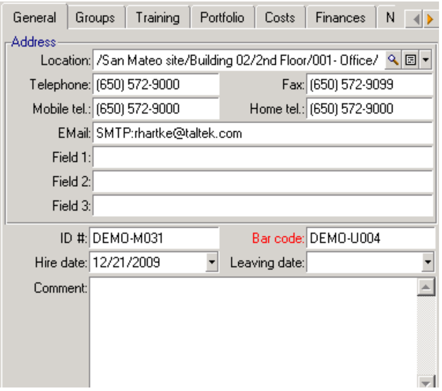
    ```
    SMTP:[name]
    ```

  - **Password** (MailPassword): Leave this field empty, unless your SMTP server requires a login.


Name: Hartke
First: Richard
Title: Building Maintenance M.
Mr/Mrs: Mr.
Department: Administration,
Account: SMTP:Hartke
Password: ********

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

```
SMTP:[name@address.domain]
```



Location: /San Mateo site/Building 02/2nd Floor/001- Office/
Telephone: (650) 572-9000
Fax: (650) 572-9099
Mobile tel.: (650) 572-9000
Home tel.: (650) 572-9000
EMail: SMTP:rhartke@taltek.com
Field 1:
Field 2:
Field 3:
ID #: DEMO-M031
Bar code: DEMO-U004
Hire date: 12/21/2009
Leaving date:
Comment:

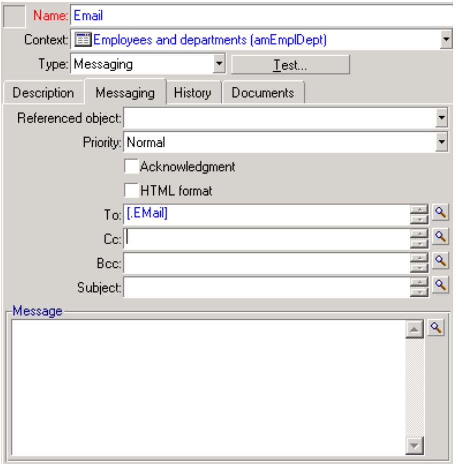
Configuring the receiver

The recipient is identified by the **To** field (SQL name: MsgTo) in the detail of a messaging-type action. The Email can be sent either in plain text or HTML format.

- 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 and departments (amEmplDept)
Type: Messaging
Referenced object:
Priority: Normal
☐ Acknowledgment
☐ HTML format
To: [EMail]
Cc:
Bcc:
Subject:

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.

- Select the **HTML format** (bHTML) option and enter the message contents in HTML tags if you want the Email to be sent in HTML format.

> Note:
>
> The HTML format is only applicable for messages sent in SMTP protocol.

Messaging parameters in `mail.ini`

To use the messaging system in Asset Manager, you must manually create the `mail.ini` file and add the following command lines:

```
[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
```

> Note:
>
> The `mail.ini` file must be created under a specific folder, the location of this file is: ▶ **Asset Manager - Installation and upgrade** guide, chapter **.ini and .cfg files**. This applies to both Windows and UNIX platforms.

MAPI protocol

Detail of the user

In order to send a message, Asset Manager needs to know the account of the sender, specified in the Employees and departments table (**Messaging** tab), and the messaging address of the recipient, specified in an action.

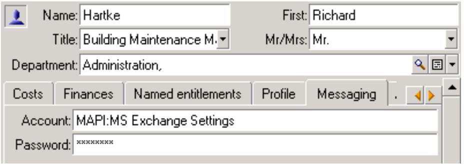
Configuring the sender

- 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**.

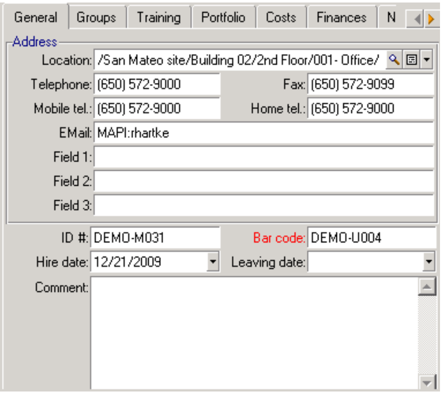
- **Password**: (MailPassword): Populate the field with your messaging password.



Name: Hartke
First: Richard
Title: Building Maintenance M.
Mr/Mrs: Mr.
Department: Administration,
Account: MAPI:MS Exchange Settings
Password: ********

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

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



Location: /San Mateo site/Building 02/2nd Floor/001- Office/
Telephone: (650) 572-9000
Fax: (650) 572-9099
Mobile tel.: (650) 572-9000
Home tel.: (650) 572-9000
EMail: MAPI:rhartke
Field 1:
Field 2:
Field 3:
ID #: DEMO-M031
Bar code: DEMO-U004
Hire date: 12/21/2009
Leaving date:
Comment:

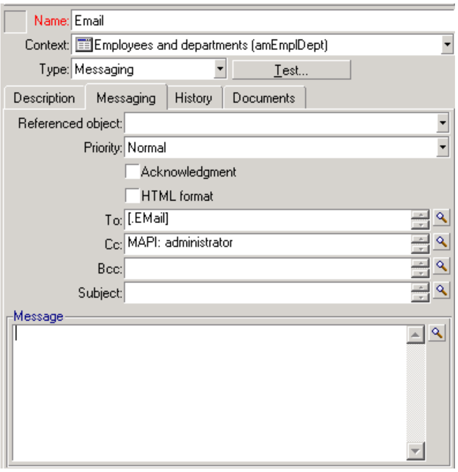
Configuring the receiver

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.



Name: Email
Context: Employees and departments (amEmplDept)
Type: Messaging
Referenced object:
Priority: Normal
☐ Acknowledgment
☐ HTML format
To: [.EMail]
Cc: MAPI: administrator
Bcc:
Subject:
Message

In this example, the **To** field is populated with the contents of the **Email** field, in the **General** tab of a user's detail.

Messaging parameters in `mail.ini`

To use the messaging system in Asset Manager, you must manually create the `mail.ini` file and add the following command lines:

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

Note:

The `mail.ini` file must be created under a specific folder, the location of this file is: ▶ **Asset Manager - Installation and upgrade** guide, chapter **.ini and .cfg files**. This applies to both Windows and UNIX platforms.

VIM protocol

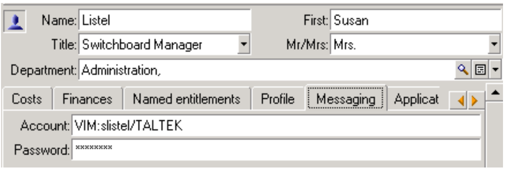
Detail of the user

In order to send a message, Asset Manager needs to know the account of the sender, specified in the Employees and departments table (**Messaging** tab), and the messaging address of the recipient, specified in an action.

Configuring the sender

- The account of the sender is identified by the following fields:
  - **Account** (SQL name: MailLogin): this must be in the form:

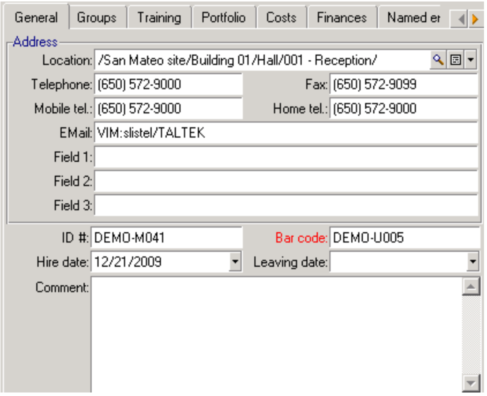
```
VIM:[name/domain]
```



Name: Listel
First: Susan
Title: Switchboard Manager
Mr/Mrs: Mrs.
Department: Administration,
Account: VIM:slistel/TALTEK
Password: ********

  - **Password**: (MailPassword): Populate the field with your messaging password.
- Populate the **Email** field (SQL name: Email) of the **General** tab with the following syntax:

```
VIM:[name/domain]
```



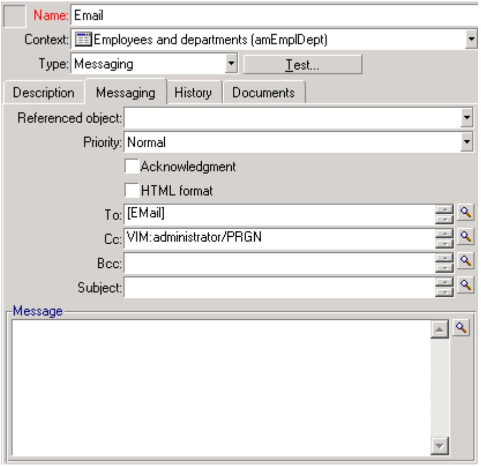
Location: /San Mateo site/Building 01/Hall/001 - Reception/
Telephone: (650) 572-9000
Fax: (650) 572-9099
Mobile tel.: (650) 572-9000
Home tel.: (650) 572-9000
EMail: VIM:slistel/TALTEK
Field 1:
Field 2:
Field 3:
ID #: DEMO-M041
Bar code: DEMO-U005
Hire date: 12/21/2009
Leaving date:
Comment:

Configuring the receiver

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]
```



Name: Email
Context: Employees and departments (amEmplDept)
Type: Messaging
Referenced object:
Priority: Normal
☐ Acknowledgment
☐ HTML format
To: [EMail]
Cc: VIM:administrator/PRGN
Bcc:
Subject:

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 Asset Manager to be able to manage the VIM protocol. Depending on your version of Windows, the procedure differs slightly.

1. Note the access path to the `vim32.dll` file (located in your Lotus Notes folder).
2. Edit the computer's Path variable to include the path to this directory.

Messaging parameters in `mail.ini`

To use the messaging system in Asset Manager, you must manually create the `mail.ini` file and add the following command line:

```
SMI=1
```

Note:

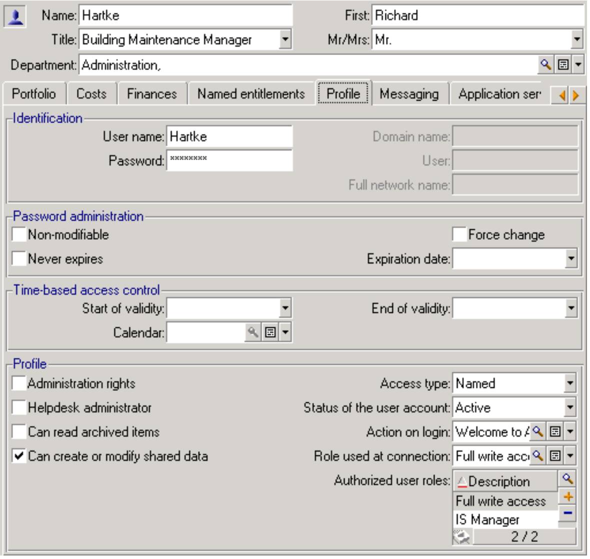
The `mail.ini` file must be created under a specific folder, the location of this file is: ▶ **Asset Manager - Installation and upgrade** guide, chapter **.ini and .cfg files**. This applies to both Windows and UNIX platforms.

AM protocol

Configuring the sender

In order for an internal message to be sent, the user must be registered in the Employees and departments table (amEmplDept) and have a login:

- The login of the user is specified in the **Employees and departments** (amEmplDept) table (**Profile** tab).



Name: Hartke
First: Richard
Title: Building Maintenance Manager
Mr/Mrs: Mr.
Department: Administration,

Identification
User name: Hartke
Password: ********
Domain name:
User:
Full network name:

Password administration
☐ Non-modifiable
☐ Force change
☐ Never expires
Expiration date:

Time-based access control
Start of validity:
End of validity:
Calendar:

Profile
☐ Administration rights
Access type: Named
☐ Helpdesk administrator
Status of the user account: Active
☐ Can read archived items
Action on login: Welcome to A
☑ Can create or modify shared data
Role used at connection: Full write acc
Authorized user roles: Description
Full write access
IS Manager
2 / 2

- You must also populate the **Email** field in the **General** tab of the user with the following syntax:

```
AM:[user login]
```

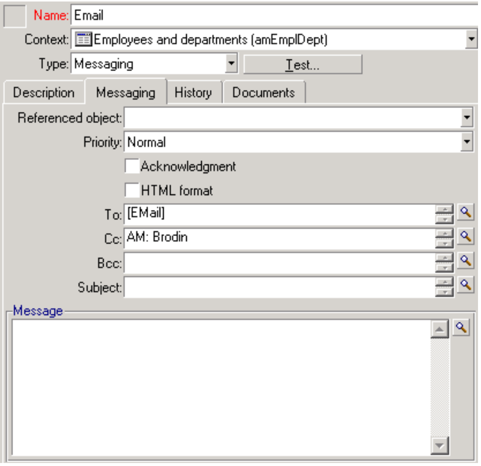
Configuring the receiver

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.


Name: Email
Context: Employees and departments (amEmplDept)
Type: Messaging
Priority: Normal
☐ Acknowledgment
☐ HTML format
To: [EMail]
Cc: AM: Brodin

In this example, the **To** field is populated with the contents of the **Email** field, in the **General** tab of a user's detail.

Sending and receiving messages

How to issue messages

Messages are generated via **Messaging** actions.

These actions must first be created in order for a message to be able to be sent.

To send HTML format message, select the **HTML format** option (SQL name: bHTML) in the **Messaging** tab of the messaging-type action.

Note:

The HTML format is only applicable to the SMTP protocol messages.

The action is triggered in different ways:

- Manually,
  - Windows client: by selecting the action in the list given by the **Tools/ Actions** menu item or from the shortcut menu on the context table (defined by the **Context** field populated when creating the action) screen.

  - Web client: by selecting the action from the **Action** drop-down list or the shortcut menu on the context table screen.
- Automatically, via Asset Manager Automated Process Manager.
- Automatically, via Asset Manager.

The creation of messaging-type actions is described the **Advanced use** guide, **Actions** chapter, section **Creating an action/ Populating the Messaging tab**.

How to consult messages

Consulting messages sent to the internal messaging system

An agent test for the arrival of new internal messages and informs the users of the presence of new messages.

These messages can be consulted:

- Windows client: using the **Tools/ Messages** menu item.
- Windows and Web client: from the message box which informs of the presence of new messages.

Consulting messages sent to an external messaging system

The recipient of these messages can consult them in the usual way under the external messaging system client.

Acknowledgement of receipt

> Note:
>
> You cannot receive an acknowledgement for a message sent via the Asset Manager internal messaging system, or via a MAPI or SMTP messaging system.

To receive acknowledgment, check the **Acknowledgment** box (SQL name: bAcknowledgment) field in the detail of the **Messaging** type action.

This acknowledgement will be sent to the address specified by the **EMail** field (SQL name: EMail) in the **General** tab of the employee who opened the Asset Manager database (in the Employees and departments table).

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 Asset Manager Automated Process Manager.
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 `mail.ini` file.

In order for Asset Manager to work correctly with MAPI messaging systems, the `mail.ini` file must contain the following lines in the [Mail] section:

MAPI=1

MAPIX=1

In order for Asset Manager to work correctly with VIM messaging systems, the `mail.ini` file must contain the following line in the [Mail] section:

SMI=1

In order for Asset Manager to work correctly with SMTP messaging systems, the `mail.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 `mail.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. Asset Manager 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 Asset Manager 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.

15 Customizing planner viewer screens

Overview

A planner viewer page can be added to:

- The detail of a table using Asset Manager Application Designer.
- To a wizard, using Asset Manager.

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 Asset Manager Application Designer:

1. Launch Asset Manager Application Designer.
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]|Complet
   eDates=[J]|LineLabel=[K]|ItemLabel=[L]|Progress=[M]
   ```

   Syntax: ▶ References: parameter syntax of the planner viewer pages [page 292].
9. Add the field thus defined (▶ button).
10. Save the page by clicking **Create**.
11. Display the details of the table (**View/ Screens** 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 Asset Manager Application Designer.

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-lWorkOrderId|StartDate=dtActualFixStart|EndDate=dtActualF
ixed|EndBefore=dtSchedFixed|StartBefore=dtSchedFixStart|ItemColor=lTech
Id|GenerateColors=true|CompleteDates=true|LineLabel=self|ItemLabel=Titl
e
```

The parameter values thus defined are interpreted as follows:

Parameter	Value	Effect
PlannerViewer	lWorkOrderId	The events are the work orders themselves.
StartDate	dtActualFixStart	The start of the events is defined by the **Actual start** field.
EndDate	dtActualFixed	The end of the events is defined by the **Actual end** date.
EndBefore	dtSchedFixed	If an event is not finished at the moment defined by the **Planned end** field, the corresponding circle ● and square ■ will appear in red.
StartBefore	dtSchedFixStart	If an event does not start at the moment defined by the **Planned start** field, the corresponding circle ● and square ■ will appear in red.
GenerateColors	true	A color is automatically assigned to the box ▭ that represents the events according to the value of the **Technician** field.
ItemColor	lTechId	
CompleteDates	true	If the **Actual start** or **Actual end** fields (dtActualFixStart and dtActualFixed) are empty, their values are replaced by the current time and date.
LineLabel	self	The description string of the **Work orders** table (amWorkOrder) is used to identify the events in the list to the left of the window.
ItemLabel	Title	The value of the **Title** field is displayed in the box ▭ 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/ Screens** 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 Asset Manager.

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 (**/Asset lifecycle/ Work orders/ 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	July 19, 2002 11:00:00	July 19, 2002 11:00:00
General	Type	seType	Internal maintenance	Internal maintenance
General	Title	Title	Analysis	Repair
Tracking/ Technician	Technician	Contact	Select the technician of your choice.	Select another technician of your choice.
Tracking/ Scheduling	Planned start	dtSchedFixStart	July 20, 2002 14:00:00	July 21, 2002 14:00:00
Tracking/ Scheduling	Planned end date	dtSchedFixed	July 21, 2002 12:00:00	July 24, 2002 17:00:00

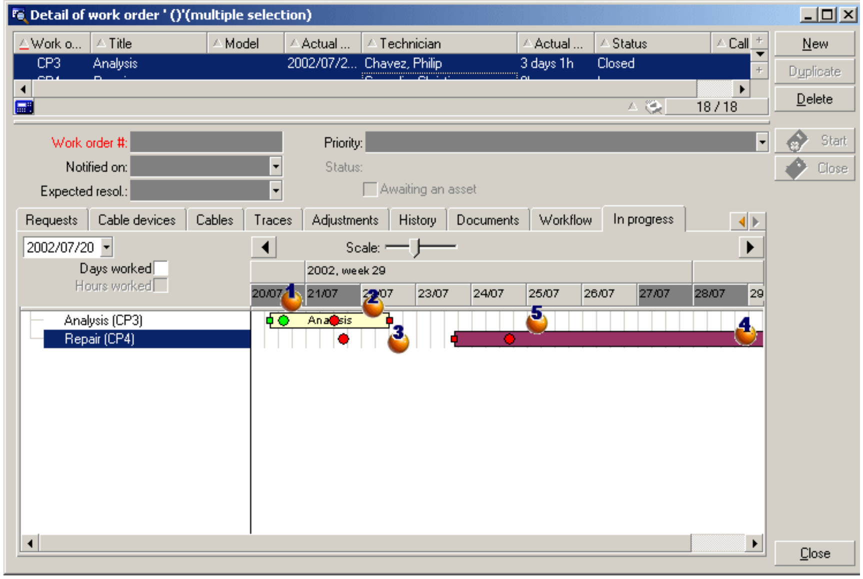
Tab	Name of the field	SQL name of the field	Value for work order 1	Value for work order 2
Tracking/ Scheduling	Actual start	dtActualFix-Start	July 20, 2002 08:00:00	July 23, 2002 08:00:00
Tracking/ Scheduling	Actual end	dtActualFixed	July 22, 2002 12:00:00	

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 15.1. Planner viewer - Example with work orders


Detail of work order ' ()'(multiple selection)

Work o...	Title	Model	Actual ...	Technician	Actual ...	Status	Call
CP3	Analysis		2002/07/2...	Chavez, Philip	3 days 1h	Closed	

18 / 18

Work order #:

Notified on:

Expected resol.:

Priority:

Status:

☐ Awaiting an asset

2002/07/20

Scale:

Days worked ☐

Hours worked ☐

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 Asset Manager wizard editor.

1. Launch Asset Manager.
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 Asset Manager 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]|Comp
leteDates=[J]|LineLabel=[K]|ItemLabel=[L]|WhereCond=[N]|OrderBy=[O]|
Progress=[M]"
```

Syntax: ▶ References: parameter syntax of the planner viewer pages [page 292].

- **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 Asset Manager.
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 Asset Manager Application Designer).

  Syntax and the list of usable parameters: ▸ Adding a planner viewer page to a table [page 286].
- A page in a wizard (with Asset Manager).

  Syntax and the list of usable parameters: ▸ Adding a planner viewer page to a table [page 286].

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

Use	Specify which events to display.

Values	- If the events are part of the table:
	`<Primary key of the table>`
	The table providing context to the other parameters is either:
	- The table to which you add the planner viewer page.
	- The **strTableSqlName** table of the API **amGeneratePlanningData(strTableSqlName, strProperties, strIds)** that defines the wizard.
	- If the events are part of a linked table:
	`<SQL name of link1>.<SQL name of link2>...<SQL name of linkN>`
	The table providing context to the other parameters is then the target table of **LinkN**.
	Exception: for the parameter `LineLabel=[K]`, it's the source table of **linkN** that provides context.

Example 1	If you add a planner viewer page to the **Work orders** table (amWorkOrder) and you want to view the work orders themselves, `[A]` equals **lWorkOrderId**.
Example 2	If you add a planner viewer page to the **Work orders** table (amWorkOrder) and you want to view the internal technician absences, `[A]` equals **Technician.Absences**.
Default value	None
Mandatory	Yes

`StartDate=[B]`

Use	Defines the start of events.

Values	A **Date** or **Date and time** type field specified as follows: ■ **<SQL name of a direct field in the table providing context>** ■ **<SQL name of link1 in the table providing context>.<SQL name of link2>...<SQL name of linkN>.<SQL name of the field>** Note: The table providing context is defined by the parameter `PlannerViewer-[A]` (table) or `MainField=[A]` (wizard)
Example	For the **Absences** table (amAbsence), the **Out date** field, `[B]` equals **dtOut**.
Default value	None
Mandatory	Yes

EndDate=[C]

Use	Defines the end of events.
Values	A **Date** or **Date and time** type field specified as follows: ■ **<SQL name of a direct field in the table providing context>** ■ **<SQL name of link1 in the table providing context>.<SQL name of link2>...<SQL name of linkN>.<SQL name of the field>** Note: The table providing context is defined by the parameter `PlannerViewer-[A]` (table) or `MainField=[A]` (wizard)
Example	For the **Work orders** table (amWorkOrder), **Planned end** field, `[C]` equals **dtSchedFixed**.
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 `StartAfter` parameter at the same time as the `StartBefore` parameter.
Values	A **Date** or **Date and time** type field specified as follows: ■ **<SQL name of a direct field in the table providing context>** ■ **<SQL name of link1 in the table providing context>.<SQL name of link2>...<SQL name of linkN>.<SQL name of the field>** Note: The table providing context is defined by the parameter `PlannerViewer-[A]` (table) or `MainField=[A]` (wizard)
Example	For the **Work orders** table (amWorkorder), **Expected resol.** field, `[D]` equals **dtResolLimit**.

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 `EndAfter` parameter at the same time as the `EndBefore` parameter.
Values	A **Date** or **Date and time** type field specified as follows: ■ **<SQL name of a direct field in the table providing context>** ■ **<SQL name of link1 in the table providing context>.<SQL name of link2>...<SQL name of linkN>.<SQL name of the field>** Note: The table providing context is defined by the parameter `PlannerViewer-[A]` (table) or `MainField=[A]` (wizard)
Example	For the **Work orders** table (amWorkorder), **Planned end** field, `[E]` equals **dtSchedFixed**.
Default value	None
Mandatory	No

`StartAfter=[F]`

Use	Defines the deadline after which the event must start.
Precautions	You must not use the `StartBefore` parameter at the same time as the `StartAfter` parameter.
Values	A **Date** or **Date and time** type field specified as follows: ■ **<SQL name of a direct field in the table providing context>** ■ **<SQL name of link1 in the table providing context>.<SQL name of link2>...<SQL name of linkN>.<SQL name of the field>** Note: The table providing context is defined by the parameter `PlannerViewer-[A]` (table) or `MainField=[A]` (wizard)
Example	For the **Work orders** table (amWorkorder), **Notified on** field, `[F]` equals **dtNotif**.
Default value	None
Mandatory	No

`EndAfter=[G]`

Use	Defines the deadline after which the event must end.

Precautions	You must not use the `EndBefore` parameter at the same time `EndAfter` parameter.
Values	A **Date** or **Date and time** type field specified as follows: ■ **<SQL name of a direct field in the table providing context>** ■ **<SQL name of link1 in the table providing context>.<SQL name of link2>...<SQL name of linkN>.<SQL name of the field>** Note: The table providing context is defined by the parameter `PlannerViewer-[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 ▭ that represents events.
Precautions	The color that is the reference for the generation of the other colors is `ItemColor=[H]`.
Values	■ **true**: the colors will be automatically modified each time the reference color changes. ■ **false**: the colors are defined in RGB format by a given field.
Default value	true
Mandatory	No

ItemColor=[H]

Use	Determines the reference color to use to color the box ▭ 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. `[H]` can take one of the following forms: ■ **<SQL name of a direct field in the table providing context>** ■ **<SQL name of link1 in the table providing context>.<SQL name of link2>...<SQL name of linkN>** ■ **<SQL name of link1 in the table providing context>.<SQL name of link2>...<SQL name of linkN>.<SQL name of the field>** Note: The table providing context is defined by the parameter `PlannerViewer-[A]` (table) or `MainField=[A]` (wizard)

Values if `GenerateColors=[I]` equals **false**	`[H]` indicates which type of **Integer (32 bit)** stores the color of the box in RGB format. `[H]` can take one of the following forms: ■ **<SQL name of a direct field in the table providing context>** ■ **<SQL name of link1 in the table providing context>.<SQL name of link2>...<SQL name of linkN>.<SQL name of the field>** Note: The table providing context is defined by the parameter `PlannerViewer-[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 `StartDate` and `EndDate` is not populated.
Example of utilization	Managing work orders in progress.
Values	■ **true**: the dates not populated by these fields are replaced by the current date. ■ **false**: the dates not populated by these fields stay empty.
Default value	false
Mandatory	No

`ItemLabel=[L]`

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

Values	- **self**: the title takes the value of the table's description string. - **<SQL name of a direct field in the table providing context>**: the name takes the value of the field. - **<SQL name of link1 in the table providing context>.<SQL name of link2>...<SQL name of linkN>**: the name takes the value of the field. - **<SQL name of link1 in the table providing context>.<SQL name of link2>...<SQL name of linkN>.<SQL name of the field>**: the name takes the value of the field. Note: The table providing context is defined by the parameter `PlannerViewer-[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 ⑧
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 `PlannerViewer-[A]` (table) or `MainField=[A]` (wizard)
Default value	- If `[A]` is the table's primary key: description string of the table. - If `[A]` is a succession of links: description string of the last link's table.

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 Asset Manager Application Designer. Note: You cannot use a calculated field.
Values	■ **<SQL of a direct field in the table>** ■ **<SQL name of link1>.<SQL name of link2>...<SQL name of linkN>.<SQL name of field>**
Default value	None
Mandatory	No

WhereCond=[N]

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

OrderBy=[O]

Use	Sorts the events to display.

Values	List of fields or links separated by the **,** character. The fields are represented by using one of the following syntaxes: ▪ **<SQL name of a direct field in the table providing context>** ▪ **<SQL name of link1 in the table providing context>.<SQL name of link2>...<SQL name of linkN>.<SQL name of the field>** The links are represented by using one of the following syntaxes: ▪ **<SQL name of link1 in the table providing context>**: This returns the value of the primary key in the linked record. ▪ **<SQL name of link1 in the table providing context>.<SQL name of link2>...<SQL name of linkN>.<SQL name of the field>** ▪ **<SQL name of link1 in the table providing context>.<SQL name of link2>...<SQL name of linkN>.self**
Example	If you add a planner viewer page to the **Work orders** table (amWorkOrder), and you want to sort the work orders by technician, then the work order number `[O]` equals **Technician.Name, WONo**.
Default value	None
Mandatory	No

16 Using Asset Manager as a DDE server

This section describes in detail the DDE calls recognized by Asset Manager 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 Asset Manager 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 Asset Manager 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 Asset Manager, this topic is **Asset Manager**.

DDE command

These are commands that will be sent to Asset Manager 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 associated with taking calls.

Commands belonging to these groups can be of two types:

- "Execute", which enables you to execute a task in Asset Manager.
- "Request", which enables you to ask Asset Manager for information.

How to find the SQL name of a table, field or link

When you right-click any field of a table, Asset Manager displays a shortcut menu. If you select the **Configure object** menu, the window displayed by Asset Manager 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 Asset Manager as a DDE server, the "Topic" is always "Asset Manager".

Note:

Once the context is defined, it is used by default in all following DDE commands until a new context is defined.

2. Launch the command itself. Two types of command are available.
   - **Execute: <command>(<parameters>)**
   - **Request: <command>(<parameters>)**
3. Close the previously defined context.

Particularities

Each application 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 Asset Manager can receive. The syntax of each of these commands is described in detail.
- Examples of how Asset Manager 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 Asset Manager (which can be found in the `amdb.ini` file).

  Location of this file: ▶ **Asset Manager - Installation and upgrade** 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 engi
ne>;<Password for the database engine>]
```

<User>

This argument contains the name of the Asset Manager 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 Asset Manager. 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 Asset Manager. 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 Asset Manager from the current database.

Example

The following command terminates the connection to the Asset Manager 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 Asset Manager 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 by 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 Asset Manager: "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 Asset Manager.

<ViewName>

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

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 Asset Manager.

<Table>

This argument contains the SQL name of the table to be closed.

Example

The following command closes the table of assets (SQL name: amAsset):

```
Execute:CloseTable(amAsset)
```

<Table>.GetRecordCount()

Type of action

Request

Description

Returns the number of records in the table with SQL name "<Table>". The table concerned by this command must first be opened in order for this command to work.

<Table>

This argument contains the SQL name of the table for which you want to know the number of records.

Example

The following command returns the number of records in the table of asset (SQL name: amAsset):

```
Request:amAsset.GetRecordCount()
```

<Table>.SetViewMode(<Mode>)

Type of action

Execute

Description

Defines the view mode of a table that has been opened.

<Table>

This argument contains the SQL name of the table concerned.

<Mode>

This argument can be set to one of the following values:

- "Arbo": Records in the table "<Table>" are displayed in tree view.
- "List": Records in the table "<Table>" are displayed in list view.
- "ListOnly": Only the records from the table "<Table>" are displayed.
- "DetailOnly": Only the detail of the selected record in the table "<Table>" is displayed.
- "ListDetail": Displays both the list of records in the table "<Table>" and the detail of the selected record in this list.

Example

This following command puts the table of products (SQL name: 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": Deletes the selected record 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-Objet.AddLink()
- Table-Object.GetValue()
- Table-Object.Highlight()
- Table-Object.RemoveLink()
- Table-Object.SetFocus()
- Table-Object.SetValue(Value)
- Table-Link.SetValueWhere(Condition)
- Table-Object.Show()

<Table>:<Object>.AddLink()

Type of action

Execute

Description

Simulates clicking the + button of a list zone. This command allows you to add a link to a record in a table.

<Table>

This parameter contains the SQL name of the table concerned by the operation.

<Object>

This parameter contains the SQL name of the object concerned.

Example

The following command adds a value in an itemized list.

```
Execute:amItemizedList:ItemListVals.AddLink()
```

<Table>:<Object>.GetValue()

Type of action

Request

Description

Returns the value of an "<Object>" (field or link) of the table "<Table>" for the selected record.

<Table>

This argument contains the SQL name of the table concerned.

<Object>

This argument contains the SQL name of the field or link of the table "<Table>" for which you want to recover the value.

Example

The following command returns the values of the **Field1** field (SQL name: Field1) in the table of assets (SQL name: amAsset):

```
Request:amAsset:Field1.GetValue()
```

The following command returns the value of the **Model** link (SQL name: Model) of the table of assets (SQL name: amAsset):

```
Request:amAsset:Model.GetValue()
```

The following command returns the value of the **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.Hilight()
```

<Table>:<Object>.RemoveLink()

Type of action

Execute

Description

Simulates clicking the - button of a list zone. This command allows you to delete a link to a record in a table.

<Table>

This parameter contains the SQL name of the table concerned by the operation.

<Object>

This parameter contains the SQL name of the object concerned.

Example

The following command deletes the selected value in an itemized list.

```
Execute:amItemizedList:ItemListVals.RemoveLink()
```

<Table>:<Object>.SetFocus()

Type of action

Execute

Description

Sets the focus on the field or link "<Object>" of the table "<Table>" for the selected record.

<Table>

This argument contains the SQL name of the table containing the field or link to which you want to set the focus.

<Object>

This argument contains the SQL name of the field or link or the table "<Table>" to which you want to set the focus.

Example

The following command sets the focus on the **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 field that 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()
```

Commands associated with taking calls

These commands are designed to be used to take calls via DDE mechanisms.

This chapter contains information about the following commands:

- NewTicket()
- NewTicket.SetTicketMode(Mode)
- NewTicket.GetTicketNumber()
- NewTicket-Item.SetValue(Value)
- NewTicket-Item.SetValueWhere(Condition)
- NewTicket-Item.GetValue ()

NewTicket()

Type of action

Execute

Description

Triggers the answer-call tool in Asset Manager.

Example

The following command triggers the answer-call tool and allows you to populate a new helpdesk ticket:

```
Execute:NewTicket()
```

NewTicket.SetTicketMode(<Mode>)

Type of action

Execute

Description

Simulates clicking of the buttons in the answer-call screen.

<Mode>

This argument can be set to one of the following values:

- "Save": Save the current call. This mode corresponds to clicking the **Save Call** button.
- "Solved": Close the current call. This mode corresponds to clicking the **Close the ticket** button.
- "Proceed": Proceed with ticket. This mode corresponds to clicking the **Proceed** button.
- "SaveNOpen": Save the call and open the ticket. This mode corresponds to clicking the **Fill out** button.
- "Cancel": Cancels the call. This mode corresponds to clicking the **Cancel** button.
- "Close": Closes the answer-call screen. This mode corresponds to clicking the **Close** button.
- "ForceNewFile": Triggers the creation of a sub-ticket for the ticket entered in the **Ticket** link of the answer call screen.

Example

The following command triggers the answer-call tool then cancels it:

```
Execute:NewTicket()
Execute:NewTicket.SetTicketMode(Cancel)
```

NewTicket.GetTicketNumber()

Type of action

Request

Description

Returns the number of the current ticket from the answer-call screen.

Example

The following command triggers the answer-call tool and returns the number of the ticket being created:

```
Execute:NewTicket()
Execute:NewTicket.GetTicketNumber()
```

NewTicket:<Object>.SetValue(<Value>)

Type of action

Execute

Description

Populates one of the ("<Object>") fields in the answer-call screen with the value of "<Value>".

<Object>

This argument can take one of the following values:

- "Description": The command populates the **Description** field in the answer-call screen.
- "DescComp": The command populates the **Additional description** field in the answer-call screen.
- "Notes": The command populates the **Notes** field in the answer-call screen.

<Value>

This argument contains the value to be assigned to the "<Object>" field in the answer-call screen.

Example

The following command triggers the answer-call tool, then populates the **Description** and **Notes** fields:

```
Execute:NewTicket()
Execute:NewTicket:Description.SetValue(Second problem on this hardware)
Execute:NewTicket:Notes.SetValue(To fix urgently)
```

NewTicket:<Object>.SetValueWhere(<Condition>)

Type of action

Execute

Description

Populates one of the ("<Object>") links in the answer-call screen according to the condition "<Condition>"

<Object>

This argument can be set to one of the following values:

- "Caller": The command populates the **Caller** link in the answer-call screen.
- "File": The command populates the **Ticket** link in the answer-call screen.
- "Asset": The command populates the **Asset** link in the answer-call screen.
- "ProblemClass": The command populates the **Type** link in the answer-call screen.
- "EscalScheme": The command populates the **Escalation** link in the answer-call screen.
- "Contact": The command populates the **Contact** link in the answer-call screen.

<Condition>

This argument contains the condition governing the command. It is an AQL clause.

Example

The following command triggers the answer-call tool, then populates the **Caller** and **Type** links. A person called "Dernham, Elliot" and a problem type called "Printer problem" must exist in the database:

```
Execute:NewTicket()
Execute:NewTicket:Caller.SetValueWhere(Name='Dernham')
Execute:NewTicket:ProblemClass.SetValueWhere(Name='Printer broken')
```

NewTicket:<Object>.GetValue ()

Type of action

Request

Description

Returns the value of the one of the ("<Object>") fields or links in the answer-call tool.

<Object>

This argument can be set to one of the following values:

- "Caller": The command returns the value of the **Caller** link in the answer-call screen.
- "File": The command returns the value of the **Ticket** link in the answer-call screen.
- "Asset": The command returns the value of the **Asset** link in the answer-call screen.
- "ProblemClass": The command returns the value of the **Type** link in the answer-call screen.
- "Description": The command returns the value of the **Description** field in the answer-call screen.
- "DescComp": The command returns the value of the **Additional description** field in the answer-call screen.
- "EscalScheme": The command returns the value of the **Escalation** link in the answer-call screen.
- "Contact": The command returns the value of the **Contact** link in the answer-call screen.
- "Notes": The command returns the value of the **Notes** link in the answer-call screen.

Example

The following command returns the value of the **Caller** field in the answer-call tool:

```
Execute:NewTicket:Caller.GetValue
```

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: Asset Manager 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 Asset Manager application, therefore:

- Service: **am**
- Topic: **Asset Manager**

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)
```

Asset Manager 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 data entry mode; in this case **New**

The command is written as follows:

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

Entering data

Next, we need to give Asset Manager 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)
  ```

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

Note:

Asset Manager 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 Asset Manager, 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 Asset Manager.

Note:

In this way, you will be able to see the results of the orders issued by **DDE TestCenter.exe** in Asset Manager 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 1:

The following **Execute** command opens the table of features:

```
OpenTable(amLocation)
```

Example 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 1:

The following **Request** command displays the list of SQL names of all the tables in the current connection:

```
ListAllTables()
```

Example 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()
```

17 Optimizing Asset Manager for use in a WAN environment

WAN networks are often characterized by:

- Low bandwidth
- High latency

It is possible to configure Asset Manager in order to minimize these disadvantages. However, these configurations are made to the detriment of certain functions of Asset Manager.

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 that 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 Asset Manager 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 management/ Infrastructure managament/ Assets** navigation 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 Asset Manager 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 Asset Manager database and are the same for all users who access it.

For more information on configuring the interface options, consult the **Tailoring** guide, part 1 **Customizing client computers**, chapter **Customizing a client workstation**, section **General options**.

Setting list parameters at the Asset Manager client level

The **FetchingArraySize** parameter is used by all the DBMSs supported by Asset Manager for an Asset Manager data connection.

To learn about this parameter's utility, you need to know that a DBMS sends the requested records in packets from an Asset Manager 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 Asset Manager client in the `amdb.ini` file and for each Asset Manager connection.

Location of this file: ▸ **Asset Manager - Installation and upgrade** 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 Asset Manager 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, Asset Manager fetches just once.

To modify `amdb.ini`

1. Edit the `amdb.ini` file.

   Location of this file: ▶ **Asset Manager - Installation and upgrade** guide, chapter **.ini and .cfg files**.
2. Search for the selection `[<Asset Manager 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 Asset Manager 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: ▸ **Asset Manager - Installation and upgrade** guide, chapter **.ini and .cfg files**.

Index