Asset Manager

Software Version: 9.61

Windows® and Linux® operating systems

Common Tables

Document Release Date: September 2016

Software Release Date: September 2016

Hewlett Packard Enterprise

Legal Notices

Warranty

The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from Hewlett Packard Enterprise required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notice

Trademark Notices

Adobe™ is a trademark of Adobe Systems Incorporated.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

UNIX® is a registered trademark of The Open Group.

This product includes an interface of the 'zlib' general purpose compression library, which is Copyright © 1995-2002 Jean-loup Gailly and Mark Adler.

Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates or to verify that you are using the most recent edition of a document, go to: https://softwaresupport.hpe.com/.

This site requires that you register for an HPE Passport and to sign in. To register for an HPE Passport ID, click **Register** on the HPE Software Support site or click **Create an Account** on the HPE Passport login page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HPE sales representative for details.

Support

Visit the HPE Software Support site at: https://softwaresupport.hpe.com.

This website provides contact information and details about the products, services, and support that HPE Software offers.

HPE Software online support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support website to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HPE support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HPE Passport user and to sign in. Many also require a support contract. To register for an HPE Passport ID, click **Register** on the HPE Support site or click **Create an Account** on the HPE Passport login page.

To find more information about access levels, go to: https://softwaresupport.hpe.com/web/softwaresupport/access-levels.

HPE Software Solutions Now accesses the HPE Software Solution and Integration Portal website. This site enables you to explore HPE Product Solutions to meet your business needs, includes a full list of Integrations between HPE Products, as well as a listing of ITIL Processes. The URL for this website is http://h20230.www2.hp.com/sc/solutions/index.jsp.

Contents

Chapter 1: Employees and Departments

Organization of employees within a department

Departments and employees are created and managed from the same screen.

In Asset Manager, the list of departments and employees is organized hierarchically: Departments (which can contain optional sub-departments) include employees (employees may not contain other employees).

Organizing departments and employees

The best way to organize the list of departments and employees is to:

1. Create a hierarchy of the departments in your company,
2. Associate the employees to their respective departments.
   Employees are therefore at the last level of the branches.

> **Note:** To view all levels of the hierarchy, right-click the list of departments and employees and then select **Unfold/ All** from the contextual menu.

Asset Manager enables you to assign diverse roles to your company's employees:

- Users or supervisors of an asset.
- Authors of an asset view.
- Supervisors of a stock.
- Supervisors of a project.
- Requesters or technicians for a work order.
- Supervisors of a budget.
- Supervisors of a contract.

Viewing departments and employees

There are two ways to view the information pertaining to departments and employees: A flat view or a hierarchic view. These views are the standard views in the application.

You can view the departments and employees according to a field on screen.

For example, you can view the employees according to their title. This is equivalent to grouping the employees by this criteria.

To group the employees by their functions:

1. Display the Departments and Employees screen.
2. Right-click the **Title** field.
3. From the contextual menu, select **Group by this field**.
   This list of departments and employees is filtered according to the criteria **Title**. Each title is a hierarchic level and is shown grayed out. Under each title is listed the employees with that title.

Asset Manager users

In order to protect access to the database, only an Asset Manager user who is declared in the database can open the Asset Manager database.

Several users may work concurrently on the same database.

An Asset Manager user has a record in the list of departments and employees, and is assigned a **Login** and a **Password** (SQL names: UserLogin and LoginPassword) by the administrator.

Asset Manager administrators

An Asset Manager administrator is a user who has rights to all tables in the Asset Manager database.

Several users may be administrators. To create an administrator, another administrator simply assigns administrator rights to that user.

The Admin login record

By default, the Departments and Employees table includes an administrator whose **Login** is "Admin".

- When Asset Manager is first installed, this is the only login name that enables you to access the Asset Manager database for all administrative operations.
- This login enables you to connect if you cannot connect as an administrator under any other name. For security reasons, this record cannot be destroyed.
- Even though the access type for the "Admin" login is "Named", its license is not decremented from the number of authorized named users.
- The "Admin" login is the only login that you can use to delete connection slots via the **Delete** button in the connection slot management screen (**Administration/ Connection/ Connection slots** menu item).

Creating departments and employees

1. Select the **Organization management/ Organization/ Employees and departments** link on the navigation bar.
2. Do not use the fast search (click **Cancel** in the window automatically displayed)
3. Click **New**.

> **Note:** The basic information (fields and tabs) contained in the details of a department and that of an employee are not the same.

Employee training

The **Training** tab displays the list of training programs the employee has taken:

Windows client

- Use the + and - buttons to the right of the list to add or delete training programs.
- You may display or modify a training program in the list by clicking the 🔍 button.

> **Note:** Just like assets, training items each have a unique bar-code label. By default, this label is defined by a script that uses an Asset Manager counter. You can customize this script to adapt the bar-code label to your needs.

Web client

- Use the **Add** or **Delete** button to add or delete training programs.
- Click the training record to display the program details.

Create a delegation

Delegations let users transmit task assignments from one user (Employee delegating) to another user (Employee delegated to) and do so for a determined or undetermined period of time.

When a delegation from employee A to employee B is created and activated, all workflow tasks assigned to employee A, or to a group of which employee A is a member, are sent to employee A and also to employee B.

To create a delegation:

1. Display the list of employees (**Organization management/ Organization/ Employees and departments** link on the navigation bar).
2. Display the details of the employee (employee A) who wants to create a delegation.
3. Click the **Delegate** button (Windows client) or select **Delegate** from the **Actions...** drop-down list (Web client).
4. Next, complete the **Create/Modify delegations...** (sysCoreDelegation) wizard.
5. Make sure you activate the **Delegation active** option.
6. This creates the delegation in the delegations table (amDelegation).
7. The delegation can now be seen:
    - In the detail screen for employee A: **Delegations** tab, **List of granted delegations** sub-tab.

- In the detail screen for employee B: **Delegations** tab, **List of received delegations** sub-tab.

> **Note:** If the end date (dtEndDeleg) has not been specified, the delegation is considered permanent.

> **Note:** To enable a delegation, check the **Enabled** (bActive) box on the delegation's detail page.
>
> This option lets you disable temporarily a delegation without deleting it or modifying its end date

> **Caution:** No profiles or validation rights are transferred when a delegation is created. Consequently, employees receiving delegations must have the same validation rights as employees granting delegations to accept or deny tasks that are transmitted via the delegation.

Associating access conditions with Asset Manager users

To assign access conditions to an Asset Manager user: refer to the **Administration** guide, chapter **Controlling access to the database**, section **Associating access conditions with Asset Manager users**.

Employee groups

Use the **Organization management/ Operations/ Employee groups** navigation menu to create employee groups.

Employee groups are used in several areas of the software.

For example:

- They can be the assignees of **Internal maintenance Type** (SQL name: seType) work orders.
- They can be the assignees of workflow tasks.

Employee groups are stored in the table of employee groups (SQL name: amEmplGroup). This table is hierarchical.

To create an employee group:

1. Select the **Organization management/ Operations/ Employee groups** menu item (Windows client).
2. Click **New**.
3. Enter the name of the employee group.
4. Enter the group to which it belongs, if necessary.
5. Specify a group supervisor.
6. Specify the members of the group in the **Composition** tab.
7. Enter any locations dealt with by the group in the **Locations** tab.
8. Enter any contracts dealt with by the group in the **Contracts** tab.
9. You can also use the **Areas of expertise** tab to enter the problem types for which the group is competent.

> **Note:** Just like assets, employee groups each have a unique bar-code label. By default, this label is defined by a script which uses an Asset Manager counter. You can customize this script to adapt the bar-code label to your needs.

When employees change location

1. Display the list of locations (**Organization management/ Organization/ Locations** link on the navigation bar).
2. Create the new locations.
3. Display the list of departments and employees (**Organization management/ Organization/ Employees and departments** link on the navigation bar).
4. Select all the employees who will move to a new location.
5. In the **General** tab of the employee detail, select the new location.
6. Click the **Modify** button.
7. Display the list of portfolio items (**Asset lifecycle/ Infrastructure management/ Portfolio items** link on the navigation bar).
8. Select all the portfolio items also moving to the same location.
9. In the **General** tab of the portfolio-item detail, select the new location.
10. Click the **Modify** button.

The other solution, more complex but more powerful, involves exporting the records and modifying them outside of Asset Manager, then importing the modified records.

When an employee changes department

1. Display the list of departments and employees.
2. Display the details of the employee who is changing departments.
3. In the **Department** field (SQL name: Parent), select the employee's new department.
4. If necessary, modify the **Location (SQL Name: Location)**, **Telephone (SQL Name: Phone)**, **Fax (SQL Name: Fax)** and **E-mail (SQL Name: EMail)** fields on the **General** tab, as well as the **Cost center (SQL Name: CostCenter)** field on the **Finances** tab.
5. If necessary, modify the **Location**, **Supervisor** and **Cost center** fields (SQL names: Location, Supervisor and CostCenter) in the **General** tab of detail of portfolio items used by the employee. In order to do this:
   - Display portfolio item detail one-by-one from the **Portfolio** tab of the employee detail, and modify the fields accordingly.
   - Or display the list of portfolio items and apply a filter on the **User** link (SQL name: User) by selecting the name of the user concerned. Select all the user's portfolio items, which enables you to simultaneously modify the fields for the portfolio items.

Chapter 2: Locations

Definition of a location

Your company's locations are described in a separate, hierarchical table. A location is used to describe the physical placement of an asset, a stock, a place, a work order, and so on.

Location floor plans

The **Plan** tab lets you select a floor plan from the list of plans. The floor plans stored here may cover a wide range of images: drawings, maps, charts, blueprints, diagrams, and so on.

Location of portfolio items

All the portfolio items at a certain location appear in the list in the **Portfolio** tab of the location detail.

Use the + and - buttons to the right of the list to add or delete portfolio items from the list. You may display or modify portfolio items in the list by clicking the 🔍 button.

When you add a portfolio item to a location, it is automatically removed from its previous location. If you detach a portfolio item from a location, it is no longer linked to a location.

Taxes associated with a location

Tax jurisdictions can be used to associate taxes with locations. To associate a tax jurisdiction with a location, you just need to populate the **Jurisdiction** link (SQL name: TaxJuris) in the **General** tab of the location detail screen.

Chapter 3: Companies

Definition of a company

Asset Manager lets you describe third party companies and their contacts in a specific table. These entities are not to be confused with companies or branches of your own group.

Companies appear in the database as parties involved in a contract, maintenance providers, suppliers, and so on.

Company contacts

The **Contacts** tab of the company detail contains sub-tabs for each of the contacts at the company.

To add, duplicate or delete a contact, right-click within the sub-tab and select **Add linked record** or **Delete link** from the shortcut menu.

The full list of contacts is only available via the **Administration/ List of screens** menu item.

Contracts signed with companies

The **Contracts** tab in the company detail displays the list of contracts signed with the company.

Use the + and - buttons to the right of the list to add or delete contracts.

You can display or modify a contract in the list by clicking the 🔍 button.

Company catalogs

The **Catalogs** tab in the company detail displays its catalogs.

Use the + and - buttons to the right of the list to add and remove purchase terms

View or modify an existing purchase term by clicking the magnifier 🔍.

Chapter 4: Floor plans

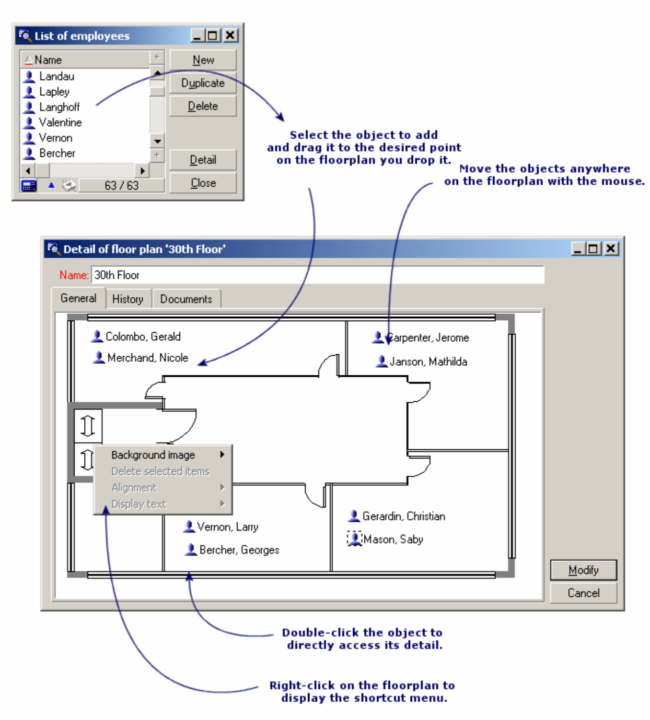
Definition of a floor plan

A floor plan is a graphical area in which you can place icons that symbolize records from such or such table. All tables can be used, in particular locations, employees and assets.

Creating a floor plan

To create a floor plan, use the shortcut menu in the plan, described below.

Floor plan detail



Select the object to add and drag it to the desired point on the floorplan you drop it.

Move the objects anywhere on the floorplan with the mouse.

Double-click the object to directly access its detail.

Right-click on the floorplan to display the shortcut menu.

Entries in the **Floor plans/ Background image** menu

- **Paste** lets you paste an image to the Clipboard.
- **Delete** lets you delete the background image.
- **Resize** lets you resize the background image. Drag the handles of the image with the mouse to re-size the image.
- **Import** lets you import an image as a background.

Alignment

Lets you align several icons in the floor plan. You can select icons by holding down SHIFT or CTRL and clicking them, or by drawing a frame around them with the mouse.

Display text

Lets you position text on the floor plan that describes the icon representing this item.

Add objects to the plan

Add records from other lists by performing a drag-and-drop operation on the records.

You may do this by selecting records in lists or by selecting the icons in the record detail windows if they exist.

Asset Manager adds the object's icon and text to the floor plan.

Move the records around the floor plan using the mouse.

How to view objects from a floor plan

Once the icons are in place, you just need to double-click on one of them to access its record detail.

Using floor plans

Floor plans can be created within floor plans to create more detailed floor plans.

For example, to create a tree structure of locations such as "State/ Site/ Building/ Floor/ Office":

1. Associate the corresponding plan to each location: country, site, building, floor and office.
   Associate each location with the corresponding floor plan: First display the list of locations using the (**Organization management/ Organization/ Locations** menu item. For each location, display the **Plan** tab and select the floor plan corresponding to the location.
2. Drag and drop each icon of the site onto the plan of its country using the two lists of plans.
   In the first list of plans, display the detail of the country plan. Use the second list to drag and drop the site plans on the country plan.
3. Use the same technique to position the building plans on the site plans, the floor plans on the building plans and the office plans on the floor plans.
4. Finally, position the users and the assets on the office floor plans.
   To do this, display the list of plans, of employees and of assets. Then, drag and drop the employees and the assets on the detail of the floor plans.

You can then display:

- A floor plan from the floor plan list (the **Organization management/ Organization/ Floor plans** menu item) or from the **Plan** tab in a location detail.
- A sub-plan by double-clicking its icon in the main floor plan to which it is attached.
- The detail of assets and users by double-clicking their icons on the office floor plans.

Chapter 5: Features

Definition of features

Features enable you to complete the description of Asset Manager objects (assets, models, orders, and so on). They are attached to objects with a value and are displayed directly in the object's **Features** tab.

You can create as many features as you wish, and specify their entry mode (Numerical, Text, and so on). This makes Asset Manager very flexible and allows for extensive customization.

By using features, you can include additional information in areas of particular importance to you (financial, technical, or any other specific area).

Features provide additional "fields" for describing the records in your database. For example, you can fill in the type of processor or the clock speed for a computer, or the horsepower of an automobile.

Finally, features can be queried using the Asset Manager query language.

Description of the features

Creating a feature involves identifying it and determining how it behaves.

Identifying a feature

The upper part of the feature detail screen is used to identify and classify the feature (using feature classes). A feature is uniquely identified by its SQL name. The other fields in this part of the screen provides extra information on the feature and are described in detail in the extended help (press SHIFT+F1 to access the extended help on a field).

Behavior of a feature

The behavior of a feature depends on several conditions:

- Its input type, which determines the type of control used in the user-interface to populate the value of the feature for a record. The input type is associated with a unit.

  > **Note:** The "Link" input type, which is complex, is the subject of a separate chapter in this manual.

- Or how it is calculated by consolidation.
- Its parameters.

Consolidated features

The information provided in the **Consolidation** tab in the detail of a feature is used when consolidating a feature. This tab concerns the features for which you want to calculate the value by consolidating the values of other features.

This is possible for features whose **Input type** field (SQL name: seDataType) in the **Input** tab is set to **Numerical** or **Duration**.

If you check the **Calculated by consolidation** box (SQL name: bCalculated), the value of the feature that you create is calculated by adding the values of other features. The features to be taken into account appear in the list in this tab. The calculation is performed as soon as you click the **Create** or **Modify** button.

Use the +, - and 🔍 buttons to add, delete, view or modify features to be included in the consolidation.

The features that you add:

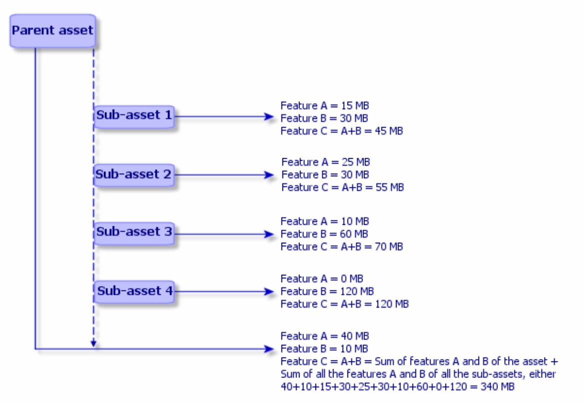
- Must have the same input type (format) and unit as the consolidated feature,
- May not themselves be calculated by consolidation.

When an included feature is modified the consolidated value is automatically recalculated.

In the consolidation calculation:

- Only those features effectively added to records are taken into account.
- All child records are included if the table is hierarchic, as shown in the following example:

Consolidated features and hierarchic tables



> **Note:** Modifying the list of features making up a consolidated feature can draw heavily on the resources of the database server. In effect, the software recalculates the value of the feature for all the records that contain it.
>
> Consult the **Tailoring** guide, **Database options** chapter, **Database options** table.

Parameters of a feature

The parameters of a feature can be found in the **Parameters** tab of the detail of the feature.

Once you have created a feature, the parameters are used to specify:

- The names of the tables that can use this feature.
- The default value for the feature.
- Any data entry and display constraints concerning this feature.

Editing the parameters of a feature

If you click the + or 🔍 buttons, Asset Manager opens the screen for editing and creating parameters.

This screen for editing parameters is not available until the feature has been created.

The parameters are linked to a table defined by the **Table** (SQL name: TableName). This field is filled in from a system itemized list (a list whose values cannot be modified) containing all the tables in Asset Manager.

> **Note:** A given feature may have different parameters for different tables.

The parameters of a feature include data-entry constraints, a default value and the contents of the extended help for this feature.

Data-entry constraints

The possible values for each constraint are as follows:

- Yes: The constraint is valid for all records in the table with which the feature is associated.
- No: The constraint is not valid for any record in the table with which the feature is associated.
- Script: Application of the constraint is subject to a Basic script.

> **Note:** It is not possible to edit or modify the Basic script used for a feature parameter..

The table below summarizes the different data-input constraints applicable to a feature:

The different data-input constraints applicable to a feature

Constraint	Description
Available	Determines the availability of the feature.
Force display	Determines whether the feature is displayed by default.
Mandatory	Determines whether populating the feature is mandatory.
Historization	Determines whether history is kept for the values of the feature.

Default value

You can define a default value for a feature. This function is identical to the one that is provided for other fields in the database.

Extended help

As for all other fields in the database, you can define up to three sections of extended help for a feature.

Feature classes

A feature class groups features with common properties. For example, features such as "Processor level 1 cache" and "Processor level 2 cache" may be grouped in a feature class named "CPU".

Use the **Administration/ Features/ Feature classes** menu item to access the screen for editing and creating feature classes.

Managing features

This section explains how to manage features.

To add a feature to a record, simply move to the **Features** tab in the detail of a record in the table in question. This tab includes two parts:

- The right-hand side lists the features already associated with the record. Here you can add or remove features using the + and - buttons.
- The left-hand side displays a tree view of the feature classes; it is used to filter the features in the right-hand side.

You can only add a feature to a record if the feature applies to the table containing that record, and if the feature is available. In other words:

- The table must appear in the **Parameters** tab in the feature detail.
- The value of the **Available** field (SQL name: seAvailable) must be set to Yes or to Script if the Basic script updates the value of the field to Yes.
- The features for which input is mandatory cannot be removed and are not offered in add mode (by the + button).

Detail of the class tree-structure

The left-hand side of the **Features** tab enables you to filter the features displayed in the right-hand side of the screen. It displays a tree view of the feature classes.

- When you select the (All) node in the tree, Asset Manager displays all the features associated with the record (except for inheritable features) in the right-hand side of the tab.

- When you select a branch in the tree (therefore a class or a sub-class), Asset Manager displays the features for that class or sub-class in the right-hand side of the tab.

Two check boxes enable you to refine the display filtering:

- When you check the **With the inheritable features** box, Asset Manager also displays the inheritable features, that is, those whose **For inheritance purposes** field (SQL name: bForInheritance) is set to Yes, in the right-hand side of the tab.
- When you check the **With sub-classes** box, Asset Manager also displays the features associated with the sub-classes of the selected classes, in the right-hand side of the tab.

Detail of features associated with a record

The right-hand side of the **Features** tab enables you to:

- View the features associated with the record
- Associate a feature with a record
- Remove a feature from a record

> **Note:** The **Features** tab in the record detail appears only if at least one available feature exists for this record's table (the feature is attached to the table and the **Available** field (SQL name: seAvailable) is set to Yes (either when specified directly, or defined with a script).

Associating a feature with a record

When you add a feature by clicking the + button, Asset Manager opens the window for choosing the features available for the current record. This window displays the features in a tree structure, organized by class. This tree displays available features only, that is, those whose **Available** field (SQL name: seAvailable) is set to Yes (either when specified directly, or defined with a script).

Now simply select the feature of your choice and click the **OK** button; Asset Manager will add it to your record.

> **Note:** This screen supports multiple selections using the CTRL and SHIFT keys. Thus you can add several features in a single operation.

When you add a feature to a record in a table involved in the inheritance process (as the target table), two cases may arise:

- Either the feature is also attached to the source table, and it is inherited by the target table along with its default value.
- Or the feature is not attached to the source table. It is not inherited, and its default value is the one defined in the feature's parameters for the target table.

Removing a feature from a record

When you remove a feature from a record by pressing the − button, Asset Manager opens a window for choosing the features already associated with the record. This window displays the features in a tree structure organized by class.

Simply select the feature you want to remove and then click the **OK** button; Asset Manager removes the feature from your record.

> **Note:** This screen supports multiple selections using the CTRL and SHIFT keys. Thus you can remove several features in a single operation.
>
> Features that are mandatory or displayed by default (that is, those whose **Mandatory** parameter (SQL name: seMandatory) is set to Yes and those whose **Force display** parameter (SQL name: seForceDisplay) is set to Yes) cannot be removed from a record.

Viewing features in a list

Features are displayed in the **Features** tab of a record, but you can also view them in the list of records in a table.

To do this, right-click the list to display the shortcut menu and select **Utilities/ Configure list**. Asset Manager opens the list configuration window.

The features associated with the table appear at the end of the list. First expand the tree to obtain a detailed list.

Then select a feature by clicking the → button. This adds a column to the list with the name of the feature in the header and the value of the feature for each record in the table. The ← button removes a column from the list, and the ↓ and ↑ buttons change the order of the columns.

Inherited features

In some cases, a feature and its default value may be inherited, that is, automatically copied from one table to another. The feature is added to the **Features** tab of a record, and the default value for the feature is copied for that record. This section explains how to create an inherited feature.

You will find information on the following topics:

- Cases where applied
- Defining an inheritable feature
- Methodology
- Example

For the remainder of this section, "Source table" refers to the table with the feature(s) to inherit and "Target table" refers to the table that will inherit one or more feature(s).

> **Note:** The inheritance process requires that the feature in question be associated with both the source and the target table..

Cases where applied

Inheritance only applies in the following cases:

- From the table of "Problem types" to the table of "Helpdesk tickets": a helpdesk ticket inherits the features associated with a type of problem.
- From the "Models" table to the "Assets", "Assets under contract", "Trainings", "Contracts" and "Work orders" tables. The records in these tables inherit the features associated with a model.
- From the "Products" table to the "Catalog references" table.

> **Note:** In these cases, features and default values in Asset Manager are inherited by default. This default behavior cannot be modified.

Defining an inheritable feature

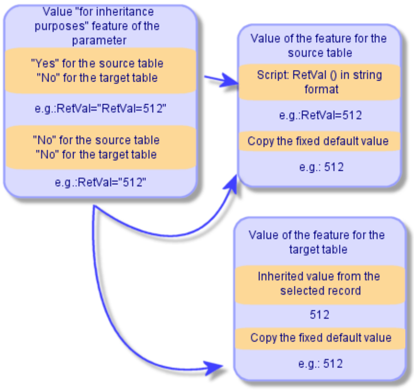
The **For inheritance purposes** field (SQL name: bForInheritance) in the **Default value** tab (SQL name: RetValScript) in the feature parameter detail determines the type of default value for the feature:

- If this field is set to Yes, the feature's default value is a default value inherited by the records in the target table, and defined by a formula containing the `RetVal()` function. This function represents the feature's inherited value.
- If this field is set to No, the feature's default value is not considered to be inheritable by the target table. Therefore it is only used to define the feature's default value for records in the source table.

> **Note:** In order for the target table to inherit a feature from the source table, the value of the **For inheritance purposes** field (SQL name: bForInheritance) must be set to Yes for the source table and to No for the target table. ("No" is the default value that cannot be modified for all those tables that can inherit feature values).

The following diagram summarizes the inheritance process:

Inheritance of features between a source table and a target table


Value "for inheritance purposes" feature of the parameter
"Yes" for the source table "No" for the target table
e.g.:RetVal="RetVal=512"
"No" for the source table "No" for the target table
e.g.:RetVal="512"
Value of the feature for the source table
Script: RetVal () in string format
e.g.:RetVal=512
Copy the fixed default value
e.g.: 512
Value of the feature for the target table
Inherited value from the selected record
512
Copy the fixed default value
e.g.: 512

To summarize, we have seen that the inheritance process for the feature's default value depends on the value of the feature's **For inheritance purposes** field (SQL name: bForInheritance) for the source and target tables:

- If this field is set to Yes for the source table and No for the target table, the feature's value for a record in the target table is defined by the `RetVal()` function of the feature associated with the record in the source table
- If this field is set to No in both the source table and the target table, the feature's value is the default value defined in the feature's parameters.

Methodology

Before looking at a practical example of this property of a feature, here is the method for efficiently implementing an inheritable feature:

1. Analyze the situation: what are the source and target tables?
2. Create or edit the feature.
3. Add the feature to the source table's **Features** tab.

4. Edit the value for the feature in that tab.
5. Check that the inheritance process works correctly by creating a new record in the target table.

Example of an inheritable feature

In this example, we will create a feature whose default value will be inherited by all the assets attached to a single model.

- **Step #1 - Analyze the requirements**

  The feature is inherited from the table of models by the table of assets. Therefore:

  - The source table is the table of models.
  - The target table is the table of assets.

- **Step #2 - Create the inheritable feature: For inheritance purposes (SQL name: bForInheritance)**

  Select the **Administration/ Features/ Features** navigation menu item to display the screen for creating feature. Fill in the fields as shown below:

  Fields for creating the inheritable feature

Field name	Field location	Field value
Title (SQL name: Title)	Feature detail screen	"Inheritance"
SQL name (SQL name: SQLName)	Top of the feature detail screen	"Inheritance" (This field is mandatory, but it is automatically populated once you place yourself in another field.)
Input type (SQL name: seDataType)	**Input** tab in the feature detail	**Numerical**

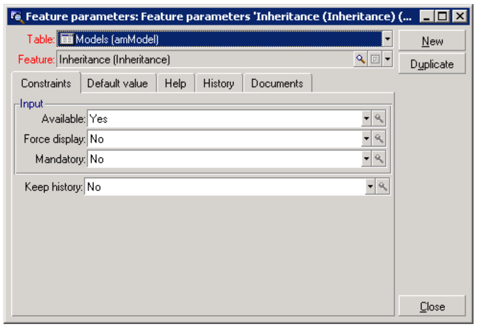
  Then click **Create** to create the feature.

  For the source table

  Now select the **Parameters** tab and click the + button next to the list of parameters. You must tell Asset Manager that this feature is associated and available for the source table and for the target table.

  Fill in the **Add parameters to feature 'inheritance'** screen as illustrated below:

Parameters for the source table (table of models)



Feature parameters: Feature parameters 'Inheritance (Inheritance) (...

Table: Models (amModel)
Feature: Inheritance (Inheritance)

Input
Available: Yes
Force display: No
Mandatory: No
Keep history: No

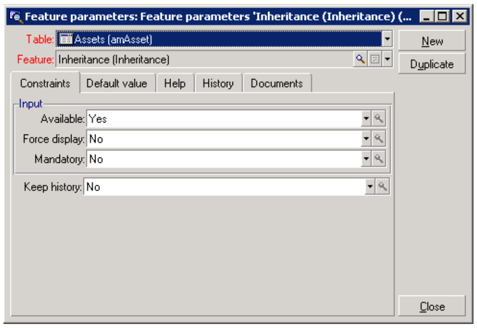
Finally, select Yes for the **For inheritance purposes** property in the **Default value** tab. Click **Add** to add this parameter to the feature.

For the target table

Now select the **Parameters** tab and click the + button next to the list of parameters. You must inform Asset Manager that this feature is associated and available for the target table.

Fill in the **Add parameters to feature 'inheritance'** screen as illustrated below:

Parameters for the target table (table of assets)



Feature parameters: Feature parameters 'Inheritance (Inheritance) (...

Table: Assets (amAsset)
Feature: Inheritance (Inheritance)

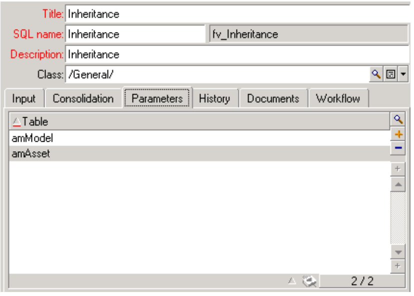
Input
Available: Yes
Force display: No
Mandatory: No
Keep history: No

Click **Add** to add this parameter to the feature.

Check

The list of parameters for our feature now looks like this:

Parameters for the "Inheritance" feature



Title: Inheritance
SQL name: Inheritance fv_Inheritance
Description: Inheritance
Class: /General/

Table
amModel
amAsset

2 / 2

Click **Modify** to validate the modifications made to the feature's parameters.

- **Step #3 - Add the feature to the Features tab in the source table**

  Now you need to add the feature to a record in your source table. You can use any model. In this example, our model is named "Deskpro 4000S". Open the table of models and select a model. Click the + button in the **Features** tab for this model and choose the "Inheritance" feature from the tree.

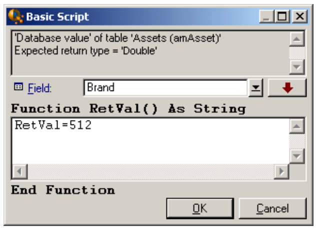
  Asset Manager adds this feature to the **Features** tab as a programmable field.

- **Step #4 - Edit the value of the feature**

  Click 🔍 to edit the Basic script for this field. This script determines the value of `RetVal()` (that is, representing the "Inherited value"). In this example, we will use a purely numeric value that will be inherited by assets with the brand "Compaq". This can be any value; in our example, we use "512".

  Therefore the script is as follows:

  Basic script for the inherited value



'Database value' of table 'Assets (amAsset)'
Expected return type = 'Double'

Field: Brand

```
Function RetVal() As String
RetVal=512
End Function
```

  To finish, click the **Modify** button to confirm your changes.

- **Step #5 - Check that the inheritance works correctly by creating a new record in the target table**

  a. In order to check the inheritance, we will create a new asset.

  b. To open the screen for creating a new asset, select the **Asset lifecycle/ Infrastructure management/ Assets and batches** menu item or click the button.

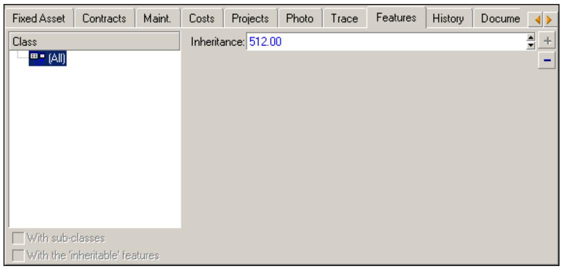
  c. Click **New** to create a new asset.

  d. Select the **Features** tab. No features are displayed.

e. Now enter in the **Model** field (SQL name: Model) the name of the model for which you added an inheritable feature (in step #3). To make sure our example is working correctly, check that the selected value is set to "Deskpro 4000S"

f. Select the **Features** tab. The "Inheritance" feature was added automatically and its value (defined in step #4) was inherited.

Inheritance of the feature in the target table



Inheritance: 512.00

☐ With sub-classes
☐ With the 'inheritable' features

Populating features by selecting a record from another table

To populate this type of feature, select a record from another table: the feature is assigned the value of that field. We have chosen an example to illustrate this Asset Manager feature.

This example involves the following steps:

1. Scenario for the example
2. Creating the feature
3. Editing the filter for the source table of the link
4. Verification

Scenario for the example

We will create a feature whose value is selected from the table of models. This feature, named "LinkTest", will be associated with the table of assets; its value will be provided by a field in the table of models.

In the remainder of this example, the table of models will be named the "Source table".

Creating the feature

Select **Administration/ Features/ Features** to display the screen for creating features. Populate the fields as follows:

Declaring the link

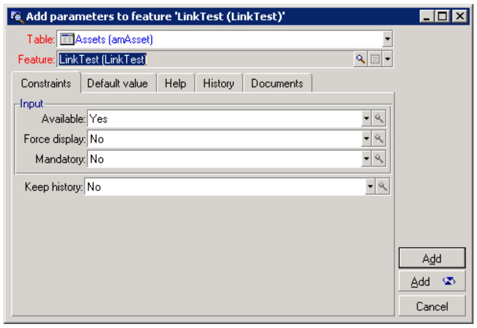
LinkTest fields

Field name	Field location	Field value
Title (SQL name: TextLabel)	Feature detail screen	"LinkTest"
Name (SQL name: Name)	Top of the detail window of a feature	"LinkTest" (This field is mandatory, but it is automatically populated when you place yourself in another field.)
Input type (SQL name: seDataType)	**Input** tab in the feature detail	**Link**
To table (SQL name: TargetTableName)	**Input** tab in the feature detail	Table of models

Press **Create** to create the feature.

Now select the **Parameters** tab and click the + button next to the list of parameters. You must inform Asset Manager that this feature is associated and available for the table of assets, as illustrated in the following screen:

Available and associated feature (the Available (SQL name: seAvailable) field is set to "Yes") for the table of assets



Add parameters to feature 'LinkTest (LinkTest)'

Table: Assets (amAsset)

Feature: LinkTest (LinkTest)

Input

Available: Yes

Force display: No

Mandatory: No

Keep history: No

Editing the filter for the source table of the link

We previously created a feature linked to the table of models and defined it as available for the table of assets. Therefore the link has been created "physically". All that remains is to specify the information to be transferred by this link, that is, which values from the table of models may be selected to populate the feature. These values may be:

- Absolute values. For example, you can select for the value of the feature all the models of a certain type. The following example selects all the models of the "Compaq" brand.

  ```
  Brand.Name='Compaq'
  ```

- Contextual values. You can select a value that depends on a field from the table of assets, such as all models associated with an asset of a certain brand. To define the context, use the `AmDbVal()` function.
  This function has the following parameters:

  ```
  AmDbVal(<Destination table of the link>:<Field or link of the destination
  table>)
  ```

  The following example defines the **Brand** field in the table of models as the context:

  ```
  AmDbVal(amAsset:Model.Brand)
  ```

Select the **Filter** tab in the detail of the parameters for the "LinkTest" feature and populate the tab as follows:

```
Brand.Name = AmDbVal(amAsset:Model.Brand.Name)
```

> **Note:** The value returned by `AmDbVal()` corresponds to the current values of the detail window, even if these values have not been saved in the database.
>
> In the previous example, lets suppose that at the database level, the asset is linked to a **DBModel** model of brand **DBBrand**. If at the detail window level, you select a model **WindowModel** of brand **WindowBrand** without having saved this modification with the **Modify** button, the **WindowBrand** brand is returned by the `AmDbVal()` function.

> **Note:** The **Filtering condition** field (SQL name: LinkFilter) is a traditional query for a table, made using the expression builder. The fields used to build the query all belong to the source table of the link.

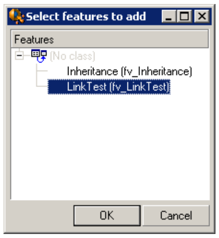
Verification

To check that the link we created performs its function, we will create a new asset, then add the "LinkTest" feature to it.

1. Open the screen for creating an asset by selecting the **Asset lifecycle/ Infrastructure management/ Assets** menu item or by clicking the button.

2. Click **New** to create a new asset.
3. Select the **Features** tab. No features are displayed in the tab.
4. Next populate the asset's **Model** field (SQL name: Model).
5. Click the + button in the features tab and choose the "LinkTest" feature in the selection screen displayed by Asset Manager:

   Selecting the "Link" type feature in the feature tree

   Asset Manager adds the "LinkTest" feature to the **Features** tab in the new asset's detail.
6. Now click the ▾ button next to the "LinkTest" feature.Choose a value from the table of models to populate this field. The itemized list enables you to choose from the models.

The feature performs its role as a link between the table of models and the table of assets.

Transforming a feature into a field

Asset Manager Application Designer enables you to transform a feature in a database into a field or a link. This operation can be performed using the wizard available via the **Database/ Transform a feature into a field** menu.

Analyzing the impact of the transformation

Before proceeding such a transformation, make sure you are taking into account the impact of the transformation, in particular:

- The properties of the features, fields and links. The following table summarizes these properties and details the transformation that is made.

  Correspondence of properties

Feature property	Equivalent field or link property
Available (*)	Opposite of the Irrelevant property

Correspondence of properties, continued

Feature property	Equivalent field or link property
Force display	No equivalent
Mandatory	Mandatory
Historization	Historized
Default value	Default
Consolidation	No equivalent
Inheritance	No equivalent
Link type feature	The feature is transformed into a link
Help on the feature	Help on the field or link

> **Caution:** (*) a non available feature has its value emptied when the database is updated. The value of an irrelevant field remains unchanged.

When you transform a feature into a field, all the properties and mechanisms specific to the features are no longer available. Make sure you have fully taken into consideration your needs and technical constraints before proceeding with the transformation.

- Amount of records for which the value is populated. If the value that you would like to associate with the records is meaningful for at least 75% of the records, then use a field instead of a feature. If the value is meaningful for less than 25% of all records, then use a feature instead of a field. If the value is meaningful for 25 to 75% of all records, then it makes no difference whether a field or a feature is used.

  > **Note:** The given percentages may vary according to the database engine you are using.

- The potential impact on performance: on databases with a high number of records, it is advised not to use a feature to store information that is frequently displayed in a list or that is subject to access restriction.

Transformation example

Once the initial analysis has been performed, you can start the transformation. In the following example, the feature named **Work Unit** whose type is **Link** is transformed into a link.

This feature is available in the Asset Manager line-of-business data. If this data has not been imported into your database, create it using the following information (**Administration/ Features/ Features** navigation menu):

Field	Value
Label	Work unit
SQL name	WorkUnit
Description	Work unit
Input type	Link
To table	Products (amCatProduct)

Click the + button in the **Parameters** tab and populate the fields as follows:

Parameters tab

<table>
<tr><th>Field</th><th>Value</th></tr>
<tr><td>Table</td><td>Departments and employees (amEmplDept)</td></tr>
<tr><td>Available</td><td>(Script)
<pre>
If [bDepartment] = 0 Then
  RetVal = 1
Else
  RetVal = 0
End If
</pre></td></tr>
<tr><td>Force display</td><td>(Script)
<pre>
If [bDepartment] = 0 Then
  Dim lRc As Long
  Dim hq As Long
  hq = AmQueryCreate()
  lRc = AmQueryExec(hq, "SELECT EmplGroups FROM amEmplDept
  WHERE lEmplDeptId = " & [lEmplDeptId] & " AND
  EmplGroups.bAssignable = 1")
  If lRc = 0 Then
    'The employee belongs to an assignable group
    RetVal = 1
  Else
    RetVal = 0
  End If
Else
  RetVal = 0
End If
</pre></td></tr>
</table>

To transform this feature into a field:

1. Launch Asset Manager Application Designer and then connect to your database (**File/ Open/ Open existing database** menu.)

2. Select the **Database/ Transform a feature into a field** menu.
3. In the first page of the wizard, select the feature in the displayed list.

   > **Note:** In the wizard, selecting the option **Only features in table** enables you to select and sort the features according to a table.

   Click **Next**.
4. The wizard gives a SQL name for the object that will be created by the transformation. You can modify this name, paying attention to its uniqueness within the table (a test is automatically run by the wizard.)
   Click **Next**.
5. The next page summarizes the transformations that will be performed. The wizard displays a summary page for each table having a feature. You navigate through these pages using the **Next** and **Previous** buttons.

   > **Note:** The **Available** script of a feature is transformed into an **Irrelevant** script for a field or a link. As these two scripts are the exact opposite of each other, the transformation systematically adds the following reversal line:
   >
   > RetVal = Not RetVal.

   Click **Next**.
6. The last page of the wizard enables you to choose a target folder for the transformation log. Click the **Finish** button to start the operation.

The transformation performs the following operations:

- The new field or link is created in the database
- Every feature property is transformed into the corresponding property for the field or link. The specific properties of the feature (consolidation, inheritance, and so on) are lost. The field type is deduced from the feature type thanks to the following correspondence:

Types correspondence

Feature type	Object created by the transformation	Type	User type
Text	Field	Text	Default
Numeric	Field	Double-precision number	Default
Boolean	Field	Integer (16 bit)	System itemized list

Types correspondence, continued

Feature type	Object created by the transformation	Type	User type
Date	Field	Date	Default
Itemized list	Field	Text	Custom itemized list
Duration	Field	Integer (32-bit)	Default
Date+Time	Field	Date and Time	Default
Link	1-N link (made of a half 1-1 link in the table to which the feature was associated and a half 1-N link in the target table of the link)	Normal (the foreign key of the link is reset to 0 if the target record is destroyed)	Normal

- The feature values are deleted as well as the feature itself if there are no more values that exist for the feature.

The new field or link is now available in the table to which the feature was previously associated (in our example, it is the **amEmplDept** table where the new link with SQL name **WorkUnit** has been created). To view the field or link in a detail screen, you must add it manually to an existing screen.

Chapter 6: Associating documents with records

Definition of a document

A document is an external file that you link to a record in the Asset Manager database.

Most of the Asset Manager tables can be associated with a document.

A document can be linked to a record in the Asset Manager database using several different methods:

- Without storing the document, but rather storing its full path.
  The document is only accessible to users who see the external document with the same path.
- By storing the document in the database.

All Asset Manager users can access the document if they possess the necessary Asset Manager user rights.

> An action can use a file dynamically using its full path. For example, an action can open the file by launching the application associated with the file.

Managing documents associated with a record

Documents associated with a record appear in the **Documents** tab of the record detail. Each individual document is described in the **Reference** frame of its sub-tab. To add, duplicate or delete a document, right-click within the tab and select the **Add linked record** or **Delete link** options from the shortcut menu.

You can create as many documents as you want, each one having its own corresponding sub-tab.

Documents not stored in the Asset Manager database

We recommend that you indicate the full path to the file on the network in order that all Asset Manager users can find it.

The **File** field (FileName) enables you to enter the full path to a file.

Click ▣ to locate this file.

> **Tip:** The specified file can be used in **Executable Type** actions. This is the main advantage of the **File** field.

Documents stored in the Asset Manager database

> **Caution:** If you store a large number of documents in the database, you need to be careful not to overload it.

Windows client

To store a document in the database:

1. Click **Import**.
2. Select the file to store.
3. Click **Modify**.

To view the document:

1. Click **Run**.

To copy the contents of the document:

1. Click **Export**.
2. Specify the full path of the file.
3. Click **Save**.

To remove the contents of the document in the Asset Manager database:

1. Click **Detach**.
2. Click **Modify**.

Web client

To store a document in the database:

1. Click **Choose File**.
2. Select the file to store.
3. Click **Open**.
4. Click **Attach**.

To view the document:

1. Click **View**.

   > **Note:** You can only view a document with one of the following file extensions:
   >
   > - JPG
   > - JPEG
   > - PNG
   > - GIF
   > - PDF
   > - TXT

To download the contents of the document:

1. Click **Download**.

To remove the contents of the document in the Asset Manager database:

1. Click **Detach**.
2. Click **Save**.

Modify a document stored in the database

Windows client

1. Display the detail of the record to which the document is attached.
2. Display the **Documents** tab.
3. Display the detail of the document to modify.
4. Click **Execute**.
5. When the **Would you like to integrate the changes that you are going to make to the document into the database?** message appears, click **Yes**.

6. Modify the document.
7. Save your changes made to the document.
8. Close the document.
9. Go back to the **Documents** tab of the record to which the document is attached.
10. Click **Modify**.

Web client

1. Display the detail of the record to which the document is attached.
2. Display the **Documents** tab.
3. Display the detail of the document to modify.
4. Click **Download**.
5. Save the document to your local disk (somewhere outside of the Asset Manager database).
6. Edit and modify the document you just saved.
7. Save your changes made to the document.
8. Close the document.
9. Go back to the **Documents** tab of the record to which the document is attached.
10. Click **Detach**.
11. Click **Choose File**, select the modified document.
12. Click **Open**.
13. Click **Attach**.
14. Click **OK**.

The Full file name field

We recommend that every document is associated with an attachment. You can follow these steps to make the **Full file name** field mandatory.

> **Note:** You can configure the field with the Windows client or the Application Designer. It takes effect on both Windows and Web clients.

1. On the **Documents** tab, right-click the **Full file name** field.
2. Click **Configure object...**.

3. Select **Yes** from the **Mandatory** drop-down list.
4. Click **OK**.

After you configure the **Full file name** field, when you try to save a document without an attachment, the following error message is returned:

```
Field 'Full file name (FileName)' in table 'Documents (amDocument)' cannot be
empty.
```

Opening a document using an action

When a document gives the full path of a file, it is possible to use this file dynamically by using actions.

Creating an appropriate action

As an example, we are going to create an action that enables you to open the file specified in the **File** field (SQL name: ActionFile) of a document. The program that opens the file is the one associated with the filename extension by the Windows file manager.

Display the list of actions using the **Tools/ Actions/ Edit** menu item. Create a new action. In particular, populate the following fields:

Action creation

Field	Value
Name (SQL name: Name)	Name of your choice
Context (SQL name: ContextTable)	**Documents** (**amDocument**)
Type (SQL name: seActionType)	**Executable**
Executable or DDE tab	
File (SQL name: ActionFile)	[FileName]

Executing the action to open the file

1. Display the detail of a record (of an asset, for example).
2. Select the **Documents** tab.
3. Describe a document in a sub-tab of the **Documents** tab.

4. Populate the **File** field (SQL name: FileName). Select a file whose filename extension is associated with a Windows file manager program.
5. Place the cursor on any field in the description sub-tab of the document.
6. Right-click to activate the shortcut menu. The **Actions** command gives the list of actions associated with the table of documents. You will find the action you have created.

> **Note:** Actions that concern links cannot be triggered with the **Tools/ Actions** menu item or the ⚙.

Chapter 7: Bar-code inventory

Overview

One quick and reliable way to inventory your company's assets and monitor their movement is by using a bar-code reader.

The bar-code reader records the IDs of each asset (normally, via a bar-code label placed on the asset) and some of the asset's properties (location, user, brand, model) available in a printed bar-code catalog.

Asset Manager lets you generate a bar-code catalog in RTF format.

You will need to do the following to make a complete bar-code inventory:

- Select and acquire the appropriate bar-code readers
- Import and reconcile data inventoried in the Asset Manager database (using the Asset Manager or HPE Connect-It import module).

Installing the bar-code font

Bar-code catalogs use a special True Type font that can transcode alphanumeric codes entered in Asset Manager into a "Code 39" series of bars. Asset Manager automatically installs this font named **C39T30Lfz** with other Windows fonts which can be viewed in the Windows Control Panel. Use this font when you create or print your labels.

Refer to your Windows guide for more information on installing fonts.

Identifying records to include in the bar-code catalog

Before generating the bar-code catalog in RTF format, you must populate the following tables. We recommend you do this in the following order:

Locations

1. Display the locations (**Organization management/ Organization/ Locations** navigation menu).
2. Create or modify the locations to add to the bar-code catalog.
   In particular, populate the **Name** (Name) and the **Bar-code** (BarCode) fields.

Departments and employees

1. Display the departments and employees (**Organization management/ Organization/ Employees and departments** navigation menu).
2. Create or modify the departments and employees to add to the bar-code catalog.
   In particular, populate the **Name** (Name) and the **Bar-code** (BarCode) fields.

Brands

1. Display the brands (**Portfolio management/ Catalog/ Brands**).
2. Create or modify the brands to add to the bar-code catalog.
   In particular, populate the **Name** (Name) and the **Bar-code** (BarCode) fields.
   Select the **Include in bar-code catalog** (bInvent) option to have the brand listed in the bar-code catalog.

Models

1. Display the models (**Portfolio management/ Asset configurations/ Models** navigation menu).
2. Create or modify the models to add to the bar-code catalog.
   In particular, populate the **Name** (Name) and the **Bar-code** (BarCode) fields.

Select the **Include in bar-code catalog** (bInvent) option to have the model listed in the bar-code catalog.

Features

1. Display the features (**Administration/ Features/ Features** navigation menu).
2. Clear the **Include in bar-code catalog** (bInvent) option for each of the features. Features are not of interest for inclusion in the bar-code catalog for this version of Asset Manager.

Itemized list **amStatus**

1. Display the itemized lists (**Administration/ System/ Itemized lists**).
2. Define the list of values for the itemized list.
   This itemized list is used for each of the asset's detail, to populate the **Status** (Status) field.

Generate the bar-code catalog

1. Display the bar-code catalog generation screen (**Portfolio/ Bar-code inventory** menu).
2. Enter the folder in which the RTF file of the bar-code catalog will be generated (**Inventory folder** field).
3. Click the **Generate the catalog in RTF format in the inventory folder** button.

Asset Manager creates a file named **cat.rtf** in the inventory folder. This file is in standard **RTF** (Rich Text Format) format and should be recognized by most word processors. It will be used to print the bar-code catalog.

The **cat.rtf** file already contains certain layout options: font, size, styles, tabs, page and section breaks. This file, in the standard (Rich Text Format) format recognized by most word processors, will be used as the basis for the data-entry books for the inventory technicians.

Here are a few tips to guarantee optimal reading:

- Do not modify the size of the bar-codes. They use a special font named C39T30Lfz delivered with the bar-code module, and which should ideally be used with a size of 30 points. If you vary this recommended size by more than 10% you may have problems reading bar-codes.

- Print your catalogs on a good quality laser printer. Make sure the toner cartridge is full and that the paper feed mechanism is clean and does not leave any traces.
- To facilitate reading and avoid errors, bar codes must be spaced approximately 2 cm (3/4 of an inch) apart.

> **Note:** Another file, **car.dat**,is created in the same folder as **cat.rtf**. **car.dat** does not need to be kept. You can leave it in the folder or delete it.

Glossary

Bar-code Catalog

A rather hefty document that resembles a catalog of the bar codes associated with the values that describe your company's assets. Reading these values directly facilitates the identification of these assets during an inventory.

Synonyms
Data-entry book.

Bar-code reader

An optical device used to read bar codes.

Synonyms
Reader, scanner, handset, data-entry terminal.

Code 39

Self-checking alphanumeric barcode. It is the most widely used barcode because of its ease of use in printing and reading.

Code 39 specifications

This section provides the full specifications for the Code 39 type bar code.

Bar code construction

Code 39 is an alphanumeric coding system that encodes decimal numbers, alphabetic letters (in upper-case only) and the following series of special characters:

```
<space> -. * $ / % +
```

> **Note:** This bar-code type can theoretically encode an unlimited number of characters.

> **Caution:** The * character is reserved for framing encoded sequence in the bar code. It thus cannot be a part of the sequence itself.

Each encoded character is composed of nine elements: five black bars and four spaces. Out of the nine elements, three are wider than the others: two black bars and a space. The name of this type of bar code is based on this particularity. The wider elements are represented by 1s and the narrower elements are represented by 0s.

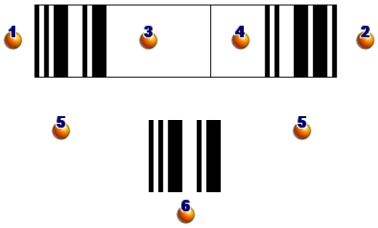
For a bar-code reader to be able to distinguish the wide and narrow elements, a ratio between the size of these two types of elements is necessary. According to the resolution used for printing, the size of the wider elements must be at least two times greater than the narrower elements. A 1 to 3 ratio is usually used. All elements of the same type must have the same size when printed: For example, the size of a narrow bar must be equal to the size of a narrow space.

Code 39 is a discrete-type bar code. Each encoded character is separated by a space (named a space character) that does not contain any information. The size of the space character is generally equal to the size of a narrow element.

Due to its structure, Code 39 can check its own characters. Nevertheless, a check character option is available.

Code structure

The following diagram presents the typical structure of a Code 39 bar code.



1 : Beginning margin.

2 : Ending margin.

3 : Encoded characters.

4 : Optional check character.

5 : Encoded "*" character, reserved to mark the beginning and the end of the bar code.

6 : Example of an encoded character.

Character set

The following table presents the list of available characters, the binary word associated with each character, and the optional check character used for each character.

ASCII character	Associated binary word	Control character value
0	000110100	0
1	100100001	1
2	001100001	2
3	101100000	3
4	000110001	4
5	100110000	5
6	001110000	6
7	000100101	7
8	100100100	8
9	001100100	9
A	100001001	10
B	001001001	11
C	101001000	12
D	000011001	13
E	100011000	14
F	001011000	15
G	000001101	16

ASCII character	Associated binary word	Control character value
H	100001100	17
I	001001100	18
J	000011100	19
K	100000011	20
L	001000011	21
M	101000010	22
N	000010011	23
O	100010010	24
P	001010010	25
Q	000000111	26
R	100000110	27
S	001000110	28
T	000010110	29
U	110000001	30
V	011000001	31
W	111000000	32
X	010010001	33
Y	110010000	34
Z	011010000	35
-	010000101	36
.	110000100	37
Space	011000100	38
*	010010100	None
$	010101000	39
/	010100010	40
+	010001010	41
%	000101010	42

Control character

The check character for a Code 39 bar code is determined by the total of the values of the character check modulo 43 (divide the total by 43). This operation's result is a value between 0 and 42 that enables you to determine the check character using the character table. For example:

Characters	C	O	D	E		3	9
Control value	12	24	13	14	38	3	9
Total of the values	113						
Modulo 43	27						
Control character	R						
Encoded sequence	CODE 39R						

Dimensions of the code

The length of a printed, Code 39 bar code is determined by the following formula:

```
Length = N*16x + 31x + (16x) + M1 + M2
```

In this formula:

- M1 and M2 represent the starting and ending margins of the bar code. Their size must be between 0.25 inches (6 mm) or 10 times the size of a narrow element if this dimension is superior to 0.25 inches.
- N represents the number of encoded characters.
- X represents the size of a narrow element. For an optimal reading of the bar codes by the bar-code reader, x must be greater or equal to 0.075 inches (0.19 mm).
- 16x represents the size of the encoded character, including the size of the space character (using a wide-narrow ratio of 3: a large element is three times the size of a narrow element).
- 31x represents the size of the starting and ending characters (*) in the sequence. This value includes the space character between the starting character and the first character of the sequence.
- (16x) represents the size of the optional check character.

When the narrow-wide ratio is not 3 (in this case, R represents the ratio used), the formula is written:

```
Length = N(3Rx+7x) + 6Rx + 13x +(3Rx+7x) + M1 + M2
```

The height of a Code 39 bar code when printed is 0.25 inches (6mm) or 0.15 times the length of the symbol (without counting its margins) if this dimension is superior to 0.25 inches.

Chapter 8: ID generation

Overview

Almost every table in an Asset Manager database has a primary key, which stores a unique global ID for every single record in these tables. The only exceptions are the **LastId** table and the **independentLastId** table. Asset Manager uses the **LastId** table and the **independentLastId** table to store and maintain so-called ID seeds. Each ID seed reserves the space for 32 global IDs.

> **Note:** The primary keys (IDs) of the **LastId** table and the **independentLastId** table are independent from the global ID system.

This chapter introduces how Asset Manager uses ID seeds to generate and allocate global IDs.

Tables used for ID generations

The following tables are involved in particular for generating global IDs:

- amWfInstance
- amLoginSlot
- LastId
- independentLastId

When a record is created in a table that is not **amWfInstance** or **amLoginSlot**, Asset Manager searches for an available ID seed in the **LastId** table. If no existing ID seed is available, Asset Manager creates a new ID seed in the **LastId** table. Similarly, when a record is created in the **amWfInstance** table or the **amLoginSlot** table, Asset Manager searches for an available ID seed in the **independentLastId** table. If no existing ID seed is available, Asset Manager creates a new ID seed in the **independentLastId** table.

> **Note:** The **amWfInstance** table and the **amLoginSlot** table have the most frequent operations among all tables in Asset Manager. For performance considerations, the ID seeds designated to these two tables are stored in an independent table (**independentLastId**). The **LastId** table and the **independentLastId** table have exactly the same data structure.

Mechanism of ID generation

The **LastId** table and the **independentLastId** table both have two columns to manage the ID seeds, **lInUse** and **lRemain**.

- The **lInUse** column is used to identify whether an ID seed is in use. When an ID seed is assigned to an active connection, the value of **lInUse** is set to 1. The ID seed cannot be shared with other connections.
- The **lRemain** column is used to identify the number of available IDs in this ID seed. Notice that this column is only updated when the user logs out. Therefore, as long as the connection is active, the value of this column does not change.

When a connection is made to the database and a record is to be created, Asset Manager follows certain rules to get an available seed ID for this connection in the **LastId** table or the **independentLastId** table.

1. Asset Manager searches for an ID seed record that meets the following conditions.
   - The value of the **lInUse** column is 0.
   - The value of the **lRemain** column is greater than 0.
2. Asset Manager assigns the ID seed with the smallest "private" ID (the primary key of the **LastId** table or the **independentLastId** table) to the connection.

   > **Note:** The primary keys ("private" IDs) of the **LastId** table and the **independentLastId** table are independent from the global ID system.

3. If there is not any ID seed available, Asset Manager creates a new ID seed record and assign it to the connection.
4. When an in-use ID seed reaches its carrying capacity (all 32 global IDs are used) and the connection is still active, Asset Manager repeats step 1 to step 3 to find a new ID seed for the connection.

   > **Note:** Because the ID seed is not released until the user logs out, the values of the **lInUse** column and the **lRemain** column of the old ID seed record are not updated.

When the user logs out, Asset Manager releases the ID seed(s) assigned to the connection by setting the value of the **lInUse** column to 0 and updating the **lRemain** column.

Send documentation feedback

If you have comments about this document, you can contact the documentation team by email. If an email client is configured on this system, click the link above and an email window opens with the following information in the subject line:

Feedback on Common Tables (Asset Manager 9.61)

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to ovdoc-ITSM@hpe.com.

We appreciate your feedback!