

Peregrine

Connect-It



HP OpenView Service Desk and OpenView AssetCenter Integration

Avis juridiques

Garanties

Les seules garanties qui s'appliquent aux produits et services HP figurent dans les déclarations de garanties formelles qui accompagnent ces produits et services.

Rien de ce qui figure dans cette documentation ne peut être interprété comme constituant une garantie supplémentaire.

HP n'est pas responsable des erreurs ou omissions techniques ou éditoriales qui pourraient figurer dans cette documentation.

Les informations contenues dans cette documentation sont sujettes à des modifications sans préavis.

Mention relative à la restriction des droits

Ce logiciel est confidentiel.

Vous devez disposer d'une licence HP valide pour détenir, utiliser ou copier ce logiciel.

Conformément aux articles FAR 12.211 et 12.212, les logiciels commerciaux, les documentations logicielles et les données techniques des articles commerciaux sont autorisés au Gouvernement Fédéral des Etats-Unis d'Amérique selon les termes du contrat de licence commercial standard.

Copyrights

© Copyright 1994-2006 Hewlett-Packard Development Company, L.P.

Marques

- Adobe®, Adobe Photoshop® and Acrobat® are trademarks of Adobe Systems Incorporated.
- Corel® and Corel logo® are trademarks or registered trademarks of Corel Corporation or Corel Corporation Limited.
- Java™ is a US trademark of Sun Microsystems, Inc.
- Linux is a U.S. registered trademark of Linus Torvalds
- Microsoft®, Windows®, Windows NT® and Windows® XP are U.S. registered trademarks of Microsoft Corporation.
- Oracle® is a registered US trademark of Oracle Corporation, Redwood City, California.
- UNIX® is a registered trademark of The Open Group.

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



Table des matières

Chapitre 1. Introduction	9
Audience	9
Prerequisite	10
Chapitre 2. Overview	11
Use case	11
Architecture	13
Chapitre 3. Installation and configuration	17
Required configuration	18
Configuration of OpenView Service Desk	18
Configuration of OpenView AssetCenter	19
Configuration of Connect-It scenarios	19
Installation of the cross-launch functionality	21
Chapitre 4. Preparing the integration	23
Adapting of scenarios	24
Configuration Item/Asset synchronization	24
Customizing date format	25
Configuration of categories (codes, itemized lists, etc.)	27
Configuration of map tables	28
Configuration of cross-launches	34
Constraints to be respected	38

Chapitre 5. Performing initial propagation	41
Chapitre 6. Configuring scheduled services	43
Chapitre 7. Implementing choices	45
Person and Employees	45
Organization and Company/Department	46
Configuration Item and Asset	46
Change/Work Order and Work Order	48
Chapitre 8. Business process example: Cost tracking with request fulfillment process	49
Chapitre 9. Unsupported features	51
Cross launches	51
Conflict with existing items	52
Location deletion	52
Employees/Workgroups relationship deletion	52
Maintenance contracts synchronization	52
Person synchronization	53
Error Management	53
Chapitre 10. Scenario Documentation File SD2AC_Init.scn	55
Chapitre 11. Scenario Documentation File SD2AC_Sync.scn	63
Chapitre 12. Scenario Documentation File AC2SD_Sync.scn	69
A. Glossary	81
AssetCenter	81
Connect-It	81
Scenario	82
Service	82
Service Desk	82
Index	83



Liste des illustrations

2.1. ServiceDesk to AssetCenter call flow	13
2.2. AssetCenter to ServiceDesk call flow	14

12.5. Connector configuration : Command line (SD import)	73
12.6. WHERE clauses by Document type	73
12.7. Mapping: Locations -> Locations	73
12.8. Mapping: Groups -> Workgroups	73
12.9. Mapping: Departments -> Organizations	74
12.10. Mapping: Departments Phones -> Telephones	74
12.11. Mapping: Departments Supervisor -> Organizations	74
12.12. Mapping: Employees -> Persons	74
12.13. Mapping: Employees Phones -> Telephones	75
12.14. Mapping: Employees Address -> Persons	75
12.15. Mapping: Employees Workgroup -> Persons	76
12.16. Mapping: Companies -> Organizations	76
12.17. Mapping: Companies Phones -> Telephones	76
12.18. Mapping: Maintenance Contracts -> Contracts	76
12.19. Mapping: Assets -> Cls	77
12.20. Mapping: Assets Users -> Cls Users	77
12.21. Mapping: Assets Parent -> Cls Parent	78
12.22. Mapping: Work orders -> Changes	78
12.23. Mapping: Work orders -> Work orders	79
12.24. Mapping: Sub Work orders CI -> Work orders Cls	79
12.25. Mapping: Reconciliation Proposals -> Incidents	79
12.26. Mapping: Relationship Employee/Group -> Members of Group	80



1 Introduction

CHAPITRE

The OpenView Service Desk and AssetCenter Integration allows to synchronize data between both applications in customizable way, depending on customer Asset and IT management process.

It also provides some direct links to both applications, to either access to some Service Desk specific information from AssetCenter interface or to AssetCenter ones from Service Desk.

This guide covers the main points of integration customization between Service Desk and AssetCenter, to be completed after installation.

For a more detailed description of the two products, please refer to the HP OpenView Service Desk and/or HP OpenView AssetCenter User's Guides.

Please also refer to Connect-It documentation set for a detailed description of scenario customization and available configuration parameter to optimize performance.

Audience

This guide and all the associated documents were originally written for internal use at HP Systems, in particular for the Research and Development Marketing departments to improve production extension and integration.

For this reason, it does not fulfill the requirements of a proper user's guide and is provided with Connect-It for technical support for the integration process.

Prerequisite

The configuration and use of the SDAC Integration imply that :

- Service Desk is installed and its database configured under a MSDE or Oracle instance.
- AssetCenter is installed and its database is empty but configured under a MSDE or Oracle instance. If the database is not empty, please contact support to update it in order to respect the required configuration and constraints listed in chapter 4.
- As this document refers to Service Desk, AssetCenter and Connect-It features, the reader should know their mechanisms. Moreover, he should be familiar with IT management processes.



2 | Overview

CHAPITRE

This chapter gives an overview of the implemented use cases and presents the architecture of the Service Desk and AssetCenter Integration. It is essential to have a good understanding of the different components to be able to adapt solution to local needs.

Use case

The use case covered here starts from an existing Service Desk database and aims at exporting data into a newly installed AssetCenter.

Then, after an initialization phase that exports existing data from Service Desk to AssetCenter, operations could be performed on one product and exported to the other.

The allowed operation type (creation, update, etc.) on one product depends on the data type (Employee, CI, etc.) and the master/slave model defined for this data type.

- AssetCenter is master for:
 - Employee
 - Department
 - Location

- CI
- Company
- Maintenance contract
- Incident
- All categories (itemized lists, codes, etc.)
- Service Desk is master for:
 - Change
 - Work Order

In consequence, following operations are authorized in both applications after initialization step:

Tableau 2.1.

	OpenView Service Desk	OpenView AssetCenter
Employee		Create/Update/Delete
Department		Create/Update/Delete
Location		Create/Update
CI	Update	Create/Update/Delete
Change/Work Order	Create/Update/Delete	Create
Company/Department (organisation)		Create/Update/Delete
Maintenance contract		Create/Update/Delete
Incident	Update	Create
Categories, itemized lists, codes		Create/Update

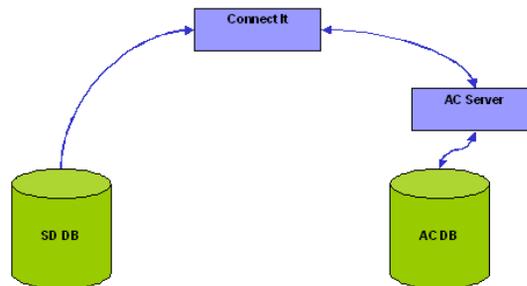
Scenarios related to the different steps and data type are described into the following sections.

Any change to the master/slave model would imply to update scenarios accordingly.

Architecture

Service Desk to AssetCenter data synchronization

Figure 2.1. ServiceDesk to AssetCenter call flow



Insert and Update scenario

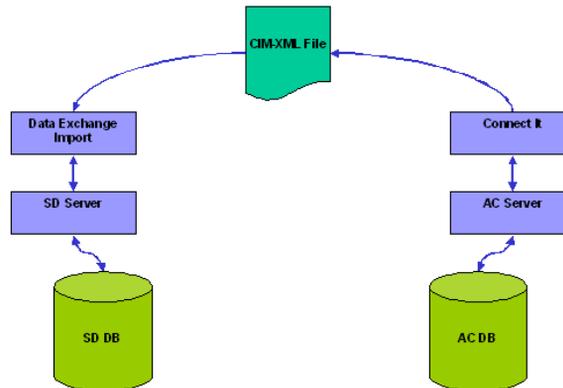
- The Connect-It uses a database connector (either Oracle or SQL Server) to access the SD database (Directly access to tables).
- Using particular columns, it detects which records have been created or updated since the last synchronization and propagates all that changes into the AC database applying the appropriate mapping.

Delete scenario

- Only possible for Changes and Work Orders
- They are not physically removed from SD DB but moved to a special folder dedicated to deleted items.
- Related AC items are physically removed from AC DB

AssetCenter to Service Desk Data synchronization

Figure 2.2. AssetCenter to ServiceDesk call flow



Insert and Update scenario

- A Connect-It scenario regularly detects OpenView AssetCenter records that have been created or updated (since the last synchronization) and that need to be replicated into the SD data store. The scenario applies the mapping as expected by OpenView Service Desk.
- The Connect-It scenario produces an XML CIM document into a given directory.
- The OpenView Service Desk Data Exchange Importer is activated to load that XML file and to apply the changes detected by Connect-It.

Delete scenario

- AC Master repository: the data is deleted.
- SD Slave repository: the data is moved to a special folder dedicated to deleted items.
- The AC workflow is configured in a way that deleted records are logged in a specific AC DB table.
- A Connect-It scenario reads regularly that table and produces a list of records which deletion has to be propagated on SD side.

- A script parses the file and applies the mark for deletion in SD by marking the object instances needing to be deleted.



3 Installation and configuration

CHAPITRE

This chapter describes the different subsets corresponding to the SDAC Integration solution and packaged into Connect-It.

These components are divided into two categories :

- scenarios files : all files related to the Connect-It scenarios that deal with the synchronization of data between AssetCenter and Service Desk.
- configuration subsets : packages containing configuration files needed for the functioning of SDAC Integration tool. These subsets are :
 - OpenView Service Desk configuration files
This component contains the files required for the configuration of the HP OpenView Service Desk product to support the data synchronization with AssetCenter.
 - AssetCenter configuration files
This component contains the files required for the configuration of the AssetCenter to support the data synchronization with Service Desk.
 - Cross Launch configuration files
This component contains the DDE package and the necessary configuration files required for the cross-launch feature of SDAC Integration.

 Note :

As the integration tool is packaged with the Connect-It distribution, this application must be installed. If you need details about how to install it, you can refer to the Connect-It User's Guide documentation.

Required configuration

This integration project is compatible with the following software versions:

Software	Version
HP OpenView Service Desk	4.5 Service Pack 16
HP OpenView AssetCenter	4.4
Connect-It	3.6 or higher

Make sure these software are correctly installed and configured on your system before going any further.

Configuration of OpenView Service Desk

The subset containing the SD configuration is composed of one XML file and should be available in the "SD Configuration" subdirectory, under the Connect-It distribution :

```
<Connect-It root path>\datakit\hpovsdac\<Integration version>\SD Configuration\ACES.
```

This file must be imported into Service Desk using the ACES importing facilities available from Service Desk Administration Console (refer to Service Desk Administrator's Guide for more information).

 Note :

When importing the XML file into Service Desk with ACES tool, the overwrite existing items option should be enabled.

It contains import mapping configuration for Service Desk Data Exchange as well as required views and templates

Importing this file will also create some categories required to be able to run provided scenarios with Service Desk and AssetCenter demo databases. These categories have been spelled in English and could be updated as well as related map tables, which is explained in the chapter 4.

Configuration of OpenView AssetCenter

All the configuration files will be available into the Connect-It distribution :

```
<Connect-It root path>\datakit\hpovsdac\<Integration version>\AC Configuration
```

It contains several workflows to install into AssetCenter application. Each one begins with the WF_ prefix.

Each workflow must be imported into AssetCenter using from menu **Tools/Workflow/Workflow Schemes**, then click on Import button.

The Import wizard need to have a .lst file to proceed, so for each workflow to import select the .lst file present in its directory :

Once all workflows are imported, the deletion is operational since these workflows deal with preparing the propagation into Service Desk of the deletion of certain AssetCenter objects.

 **Avertissement :**

The database connection must be restarted in order to activate the installed workflows. To do this just select menu File/Disconnect from database, and once disconnected, reconnect to it selecting File /Connect to database.

Configuration of Connect-It scenarios

The Connect-It Scenarios for the AssetCenter and Service Desk Integration must be configured to be able to connect to the AssetCenter and Service Desk databases and then manage the data to synchronize.

Connectors configuration

The Connect-It Scenarios for the AssetCenter and Service Desk Integration must be configured to be able to connect to the AssetCenter and Service Desk databases and then manage the data to synchronize.

From Connect-It, configure connectors according to local database installations (Database/ODBC names, user and password, etc.) Please refer to Connect-It - Connector guide for a full description of connector configuration steps.

Execute the following main steps:

- 1 Right-click the connector.
- 2 Select "Configure connector". Connect-It start the connector configuration wizard.
- 3 Click Next twice and configure the connection to the database; Populate the ODBC data source, Login and Password fields.
- 4 Click Finish.

Command files configuration

Four command files are used by the scenario synchronizing the data from AC to SD, and are available under scenario\hpovsdac\

- 1 **XML_Transformation.bat**: Called once XML files containing data to synchronize have been produced by scenario. It performs some reformatting and calls the tool to import these datas into SD.
- 2 **XML_cleanup.bat**: Called by scenario before execution to clean if needed working files resulting from previous execution.
- 3 **acsd_import.bat**: Called by XML_Transformation.bat script and deals with the call to sd_import tool which extracts datas from XML files and stores them into SD DB.
- 4 **setenv.bat**: Called by XML_Transformation.bat script to set some environment variables. This script at least should be updated to adapt the variables to your environment.

 Note :

acsd_import.bat script must be updated in case of a distributed environment, like some read/write access permissions given to remote host on working directories (scenario\hpoovsdac\\acsd and subdirectories), or the SD server hostname in the call of sd_import command.

In this framework, the following guidelines must be respected:

- The SD2AC_Sync scenario produces XML files containing data to import into Service Desk. XML_Transformation.bat processes this files and copy the results under res subdirectory. This directory must be accessible from remote host as data_exchange needs to have a read access to these files.
- Logs from the sd_import command are redirected into a text file, which is analysed by the acsd_import.bat script. In consequence these logs must be available (i.e. acsd_import.bat must have an access to it) even if the command has been remotely executed.
- When the log file does not contain any error (i.e. XML file has been processed by Service Desk sd_import tool), XML source file is moved under the Archives subdirectory. If a fatal error has occurred (Service Desk server unreachable), the process is stopped and the file stays in the res subdirectory. Then, it will try to load again the non processed files during the next launch of the script. So files in these directories must not be deleted by any external process and must be available to both hosts.

Installation of the cross-launch functionality

The subset containing the Cross-Launches configuration files should be installed in the "Cross-Launches" subdirectory available into the Connect-It distribution:

```
<Connect-It root path>\datakit\hpoovsdac\\AC Configuration
```

All these files must be installed on a Windows server where both AssetCenter and Service Desk client GUIs have been installed.

- Service Desk configuration file has been provided with the form of an ACES XML import file (See Service Desk Administrator's Guide for installation instruction), stored into "sd\ACES" subdirectory.

- AssetCenter configuration files have been provided with script file and must be imported using File/Import menu and then Execute a script action. They have been stored under "ac\Actions" subdirectory
 - DDE TestCenter, which can be found under ac\DDE subdirectory. It is the engine that permits to launch actions of an application from another one. In the framework of the SDAC Integration, it is used by Service Desk to launch AssetCenter actions. To install it just launch the Setup.exe command and follow the instructions.
-

 Note :

This installation makes some predefined cross-launches available, but according to your needs you can easily add new ones in both applications. For more explanations, please refer to the chapter 4, section Configuration of cross-launches.



4 Preparing the integration

CHAPITRE

This chapter lists a set of mandatory preparative steps to be performed after Service Desk and AssetCenter Integration has been installed and configured. Then, it lists some rules to apply in order to respect integrity constraints implemented by both applications.

 **Avertissement :**

Delivered scenarios have been based on demonstration databases provided with AssetCenter and Service Desk products, and may need to be adapted to your needs. Chapter 7 highlights some implementation choices linked to these databases which should be reviewed before using the Service Desk and AssetCenter Integration solution.

Adapting of scenarios

Customizing the scenario

All the scenarios can be adapted to customer needs directly editing scenarios by using Connect-It. Please refer to Connect-It documentation for more explanation

For example, a SD field not exported today could be mapped to an AC Feature created for this purpose, and mapping added to scenario.

To view a mapping:

- 1 Start Connect-It.
- 2 Open the scenario (**File/ Open menu**).
- 3 In the **Mapping scenario** window, **Global** tab, right-click the Mapping box and select **Edit a mapping** from the shortcut menu.
- 4 Double-click the mapping to be viewed in the **Select a mapping** window
- 5 Examine the **Mapping** window.

Configuration Item/Asset synchronization

Management of Configuration Items is a complex process differently implemented from a company to another.

In consequence, it cannot be mapped in a standard master/slave model applicable for Service Desk and AssetCenter installations

For this reason, all fields have been mapped in both senses into delivered scenarios (from SD to AC and from AC to SD). It must be modified after installation depending on customer process by opening scenario files and removing unexpected mapping

 **Note :**

Please also refer to Connect-It User's Guide for important information about how to modify a scenario, performance optimization, getting report on scenario results, etc.

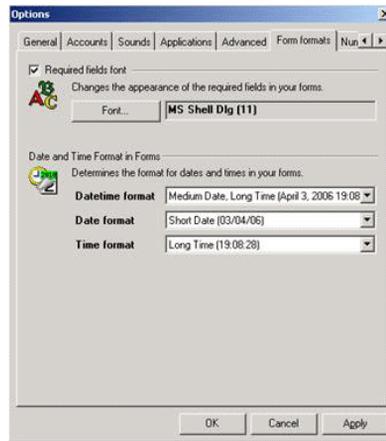
Customizing date format

In Service Desk, the date format can be customized according to the needs thanks to the tab **Form formats** in menu **Tools/Options** of the client. This has an impact on the data propagation between AssetCenter and Service Desk. Consequently, the mapping of this scenario should be adapted to the customization of the date format done in Service Desk.

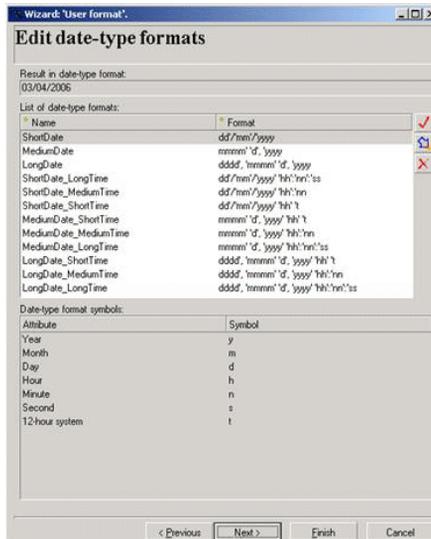
The following steps explains how to adapt the mapping:

- Open the scenario SD2AC_Sync.scn with Connect-It
- Go to menu Scenario/User formats
- Select the global.usr user formats file (the one of the SDAC Integration solution), then click on Next
- Some predefined strings are stored in this file. They defines different date and datetime formats that Service desk supports. Remember of the name of the date and datetime formats that corresponds to these one chosen in the Service Desk configuration. If another predefined format must be added to this list, just click on  button, enter a name and its predefined format. For more details about the user formats, please refer to the Connect-It User's Guide.
- Then, click on Cancel (or Ok if modifications have been done) button
- Go to menu Scenario/Global functions
- Select the global.bas functions file (the one of the SDAC Integration solution), then click on Next
- Look for the AC2SDFormatDate function, and replace the second argument of the PifUserFmtVarToStr function with the value of the date predefined format available in the user formats and corresponding to the format defined in Service Desk for Dates
- Do the same for the the AC2SDFormatDateTime function according to the chosen datetime format in Service Desk
- Finally, click on Modify button then Close button.

Let's take an example. Here is the current configuration of Service Desk Form formats :



The following screen shot presents the available predefined date and datetime formats through the Connect-It GUI :



Then just edit the global.bas file to customize the concerned functions, that is to say adapt these two following functions with the name of the corresponding predefined date and datetime formats (in red) :

Configuration of categories (codes, itemized lists, etc.)

Creation of categories into Service Desk

It is not possible to create new categories into Service Desk by a simple process 'on-the-fly'.

Considering that such items do not need to be regularly updated (as they reflect ITSM process implementation which should not be regularly updated), it is expected to manually create missing ones into SD to reflect AssetCenter configuration.

The correspondence between SD list of values and AC ones is given by map table files. (see the next section for map table configuration). These files can also be used to handle different localizations between the two applications).

Completing these map tables is a prerequisite before any synchronization.

Creation of categories into AssetCenter

Missing categories in AssetCenter can be automatically created by the Integration.

In order to handle different values into both applications (or different languages), related map tables must be filled (see next section for map table configuration).

Completing these map tables is a prerequisite before any synchronization.

Avertissement :

Please note that SD and AC categories (itemized lists) cannot have same names, even located on different places into hierarchy (as name is the key for map table).

Configuration of map tables

Customizing a map table



Map tables are fully described into Connect-It User's Guide. Please refer for more information.

Map tables have been used to in the context of Service Desk / AssetCenter Liaison to allow to have different values between both applications, or different languages.

By this way, a field that can take a value among a limited possible choices in one application can be mapped to a similar field into other application, even though list of authorized values is slightly different.

It also applies when a string value on one side must be mapped to a numbered one on the other side.

In a map table, the first column contains a key and the following columns contain values. Each key corresponds to a value in a given column.

Two map tables files have been created for the Liaison and have been stored under mpt subdirectory:

- **sdac.mpt** is used when getting a value from Service Desk to convert it into AssetCenter corresponding one.
- **acsd.mpt** is used when getting a value from AssetCenter to convert it into Service Desk corresponding one.

In order to make internationalization easier, these files contains only key references whose corresponding string value has been defined into "string files", stored into strings subdirectory and included into map table file with the directive `#include_str` at the beginning of file.

A special DEFAULT entry has been often added into map tables to return a default value when none of the other entries has matched with searched value.

 Note :

Any map table defined into either **hpovsd45ac44.mpt** or **ac44hpovsd45.mpt** file should have a corresponding one into the other file so that mapping correctly applies in both senses.

As example, map table TelephoneTypes in file hpovsd45ac44.mpt gives the list of types a telephone could have in Service Desk:

```
{ MapTable TelephoneTypes
BUSINESS | $(IDS_TELEPHONESTYPES_BUSINESS)
HOMEPHONE | $(IDS_TELEPHONESTYPES_HOME)
FAX | $(IDS_TELEPHONESTYPES_FAX)
MOBILE | $(IDS_TELEPHONESTYPES_MOBILE)
}
```

First column is the fixed key that gives the corresponding value stored in the second column. Actually, the second column only stores keys whose values as string have been defined into hpovsd45ac44.str

Following values have been defined:

```
TELEPHONESTYPES_BUSINESS, "Business"
TELEPHONESTYPES_HOME, "Home"
TELEPHONESTYPES_FAX, "Fax"
TELEPHONESTYPES_MOBILE, "Mobile"
```

This map table is used to internationalize the telephones types retrieved from Service Desk during the mapping of the phone numbers into AssetCenter. So, if Service Desk has been installed with another language or if a telephone type value has been changed, it is easy to propagate this modification in order to keep a correct mapping of phone numbers.

Let's suppose that the telephone type Home has been replaced with Personal into Service Desk, then just redefine the value of the key TELEPHONESTYPES_HOME into hpovsd45ac44.str file as following:

```
TELEPHONESTYPES_HOME, "Personal"
```

Created map tables with their format has been listed hereunder.

Service Desk to AssetCenter map table formats

Related files are **sdac.mpt** and **sdac.str**

- MapTable Currency
 - Gives the symbol of currency used in Service Desk, to be added to costs when created into AssetCenter. It has a unique entry point.

- Column 1 is CURRENCY and cannot be modified.
- Column 2 is the currency symbol as expected by AssetCenter and can be modified.
- MapTable CICategoryName
 - Gives identifier of codes related to CI Categories into Service Desk. This value may be change in the case of internationalization. It must have a single entry point.
 - Column 1 is DEFAULT and cannot be changed.
 - Column 2 is the code of CI category in Service Desk as seen in Administration Console.
- MapTable TelephoneTypes
 - Gives the different telephone types used in Service Desk.
 - Column 1 is the identifier key, and should not be changed
 - Column 2 is Service Desk value
- MapTable EmployeeGender
 - Gives the correspondence between Service Desk integer values and AsserCenter string ones.
 - Column 1 is Service Desk integer values.
 - Column 2 is AssetCenter strings.
- MapTable WorkgroupStatus
 - Gives the correspondence between Service Desk string values and AsserCenter integer ones for the mapping of workgroup status.
 - Column 1 is Service Desk values (Active or Inactive)
 - Column 2 is AsserCenter values as integer
- MapTable MaintContract
 - This map table gives the common tag used to fill the SD SourceID field. of a contract
 - Column 1 is the identifier key, and should not be changed
 - Column 2 is the key for the common tag to used.
- MapTable ContractModel
 - Service Desk only manage Maintenance Contracts. This table gives corresponding contract model to be used in AssetCenter.
 - Column 1 is MAINTENANCE and cannot be modified.
 - Column 2 is string value as expected by AssetCenter
- MapTable ContractStatus

- Maps Service Desk available statuses for contracts with AssetCenter expected values
- Column 1 is Service Desk possible values.
- Column 2 is AssetCenter corresponding value as integer.
- MapTable CICategory
 - Gives mapping between Service Desk CI category and related nature into AssetCenter.
 - Column 1 is Service Desk possible values.
 - Column 2 is AssetCenter code values of corresponding nature.
- MapTable CompanyQualif1
 - This map table gives the "Qualif 1" value to be used into AssetCenter when a Service Desk Organization is exported to an AsserCenter Company. It must have a unique entry point
 - Column 1 is DEFAULT and cannot be modified.
 - Column 2 is Qualif 1 value to export to AssetCenter.
- MapTable SD2ACWOGGeneral
 - This map table gives the name of AssetCenter work orders parent to all work orders created from Service Desk Changes. It must have a unique entry point.
 - Column 1 is COMMON_PARENT_WONO and cannot be changed.
 - Column 2 is the Work order Number of common work order parent in AssetCenter.
- MapTable FulfillmentGeneral
 - This map table gives the name of Service Desk changes and work orders categories to use when a work order created in AssetCenter is exported into Service Desk.
 - Column 1 is the identifier key, and should not be changed
 - Column 2 is the name of the category in Service Desk as seen in Administration Console.
- MapTable IncidentClosure
 - This map table gives the name of a Service Desk Incident closure code.
 - Column 1 is the identifier key, and should not be changed
 - Column 2 is the name of the closure code in Service Desk as seen in Administration Console.
- MapTable IncidentClass
 - This map table gives the name of a Service Desk Incident classification.

- Column 1 is the identifier key, and should not be changed
- Column 2 is the name of the classification in Service Desk as seen in Administration Console.
- MapTable Folder
 - This map table gives the name of Service Desk Folder used to mark an object as deleted.
 - Column 1 is the identifier key, and should not be changed
 - Column 2 is the name of this special folder in Service Desk as seen in Administration Console.

AssetCenter to Service Desk map table formats

Related files are **acsd.mpt** and **acsd.str**

- MapTable Gender
 - Gives the correspondence between AsserCenter string values and Service Desk integer ones
 - Column 1 is AssetCenter strings.
 - Column 2 is Service Desk integer values.
- MapTable EmployeeCategory
 - Gives the value of the category to use in Service Desk for an Employee mapping
 - Column 1 is the identifier key, and should not be changed.
 - Column 2 is Service Desk string values.
- MapTable CIStatus
 - Maps assets statuses from AssetCenter to Service Desk CI statuses.
 - Column 1 is AssetCenter integer value.
 - Column 2 is Service Desk related value as string.
- MapTable ChangeState
 - Maps AssetCenter Work Order status to Service Desk Change state.
 - Column 1 is AssetCenter integer value.
 - Column 2 is Service Desk related value as string.
- MapTable MaintContract
 - This map table gives the common tag used to fill the SD SourceID field of a contract
 - Column 1 is the identifier key, and should not be changed

- Column 2 is the key for the common tag to used.
- MapTable ContractStatus
 - Maps AssetCenter available statuses for contracts with Service Desk expect values
 - Column 1 is AssetCenter possible values as integer.
 - Column 2 is Service Desk corresponding value as string.
- MapTable LocationCategory
 - Maps the category of location between AssetCenter and Service Desk
 - Column 1 is AssetCenter value as string.
 - Column 2 is Service Desk value as string.
- MapTable CICategory
 - Maps AssetCenter Models to Service Desk CI categories.
 - Column 1 is AssetCenter Asset model
 - Column 2 is Service Desk CI category.
- MapTable CIBrand
 - If no brand, gives the default value to be used in AsserCenter (as it is a mandatory field in AC)
 - Column 1 is DEFAULT and cannot be changed.
 - Column 2 is default value as string.
- MapTable SD2ACWOGGeneral
 - This map table gives the name of AssetCenter work orders parent to all work orders created from Service Desk Changes. It must have a unique entry point.
 - Column 1 is COMMON_PARENT_WONO and cannot be changed.
 - Column 2 is the Work order Number of common work order parent in AssetCenter.
- MapTable FulfillmentGeneral
 - This map table gives the name of Service Desk changes and work orders categories to use when a work order created in AssetCenter is exported into Service Desk.
 - Column 1 is the identifier key, and should not be changed
 - Column 2 is the name of the category in Service Desk as seen in Administration Console.
- MapTable FWOClass

- This map table gives the correspondence between the AssetCenter models of work orders and the Service Desk Change classifications for the mapping of the request fulfillments from AssetCenter to Service Desk.
- Column 1 is the name the work order model in AssetCenter.
- Column 2 is the name of the change classification in Service Desk.
- MapTable Priority
 - This map table gives the correspondence between the AssetCenter priorities and Service Desk ones. The values of AssetCenter priorities can be changed. This map table is used for the mapping of the request fulfillments from AssetCenter to Service Desk.
 - Column 1 is the priority value as string in AssetCenter.
 - Column 2 is the priority value as integer in Service Desk.
- MapTable FWOSTatus
 - This map table gives the correspondence between the AssetCenter work order status and the Service Desk change one when a work order created in AssetCenter is exported into Service Desk.
 - Column 1 is the work order status value as integer in AssetCenter.
 - Column 1 is the change status value as integer in Service Desk.
- MapTable FSWOSTatus
 - This map table gives the correspondence between the AssetCenter sub work order status and the Service Desk work order one when a sub work order created in AssetCenter is exported into Service Desk.
 - Column 1 is the sub work order status value as integer in AssetCenter.
 - Column 1 is the workorder status value as integer in Service Desk.
- MapTable IncidentClass
 - This map table gives the name of a Service Desk Incident classification.
 - Column 1 is the identifier key, and should not be changed
 - Column 2 is the name of the classification in Service Desk as seen in Administration Console.

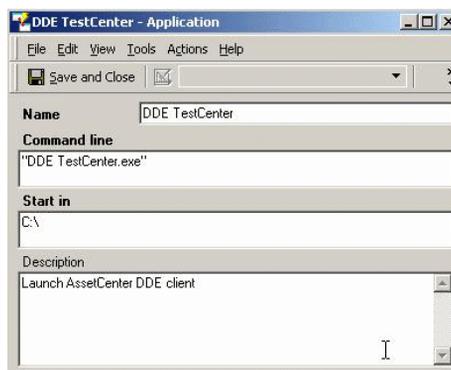
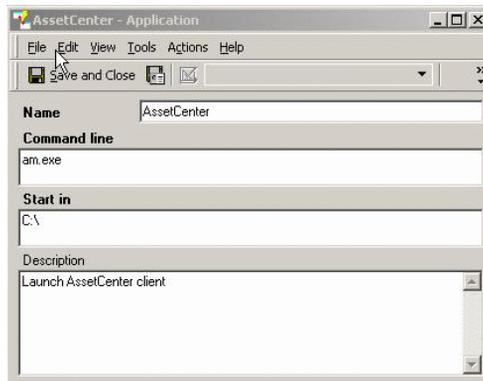
Configuration of cross-launches

Cross-launches allow the user to perform an action on one application from the other one.

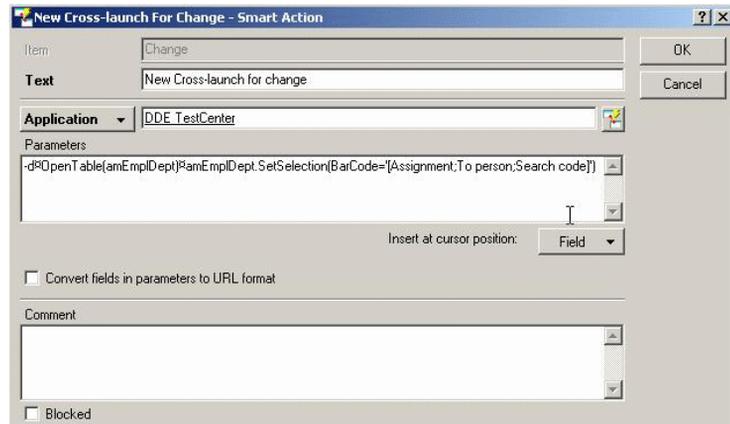
Some predefined cross-launches have been created, and can be installed with the SDAC Integration tool.

New cross-launches can be added to each application if needed, just follow these steps :

- Service Desk cross-launch creation:
 - 1 open the Administrator console (Tools/System)
 - 2 go to Business Logic/Application. AssetCenter and DDE TestCenter should be present if the predefined cross-launches have been installed. There are two types of cross-launches: the contextual ones and the non-contextual ones. The contextual cross-launches use the DDE TestCenter application and the other one use the AssetCenter application. If these applications must be redefined, refer to the following screenshots :



- then the new cross-launch can be defined. Go to Business Logic/Actions/Smart Actions. Choose the object on which a new cross-launch will be performed. Just fill the different fields as following



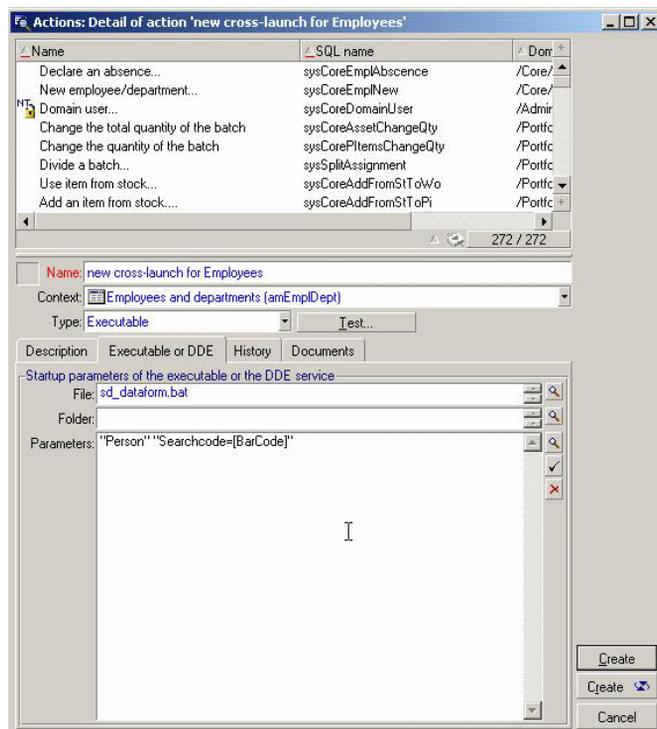
- choose the application AssetCenter or DDE TestCenter according to the kind of the cross-launch.
- specify the parameters referring to the AssetCenter Administration document, chapter 15 Using AssetCenter as a DDE server. Each parameter is delimited thanks to the ` character. The -d parameter allows to display a popup window if a problem occurs. You can also launch an action with conditions, that is to say simulate a conditional loop. First precise the -iif option as parameter, then the next parameter gathers the condition, the action to execute if true, the action to execute otherwise, each of them separated by `§ character. (Warning : The condition can only be an equality test). For example, the previous cross-launch parameters could be :



Where the action `amWorkOrder.SetSelection(WONo='[Source ID]')` is launched if the test `[Category]=AC RF Ch` is true, otherwise `amWorkOrder.SetSelection(WONo='[ID]')` action is launched

- click on ok to save this new cross-launch.
 - AssetCenter cross-launch creation :

- 1 go to Tools/Actions/Edit menu
- 2 click on New button to create a new cross-launch
- 3 fill the Name. For a contextual cross-launch, choose the object that will be concerned by this cross-launch in Context field. Select Executable as value of Type field.
- 4 in Executable or DDE tab, specify sd_dataform.bat as file. No absolute path is necessary.
- 5 in Parameters field, specify as first parameter the Service Desk targeted view, and as second one the research criteria. Each parameter must be enclosed with double-quote character. Refer to the Service Desk Administrator's Guide, chapter 5 Opening Forms from the Command Line
- 6 click on Create button once defined the cross-launch.



Constraints to be respected

Maintaining consistent data in two different systems brings about a series of constraints.

This chapter describes a list of recommendations to apply to both softwares.

Recommendations on Service Desk

Search code constraints

Search Code must be unique in Service Desk as it will be mapped to Bar Code in AssetCenter, which cannot be duplicated.

Moreover, AssetCenter Bar Code field is limited to 40 chars (Service Desk accepts up to 50 chars).

In consequence it is expected to ensure that these rules have been respected before exporting any data from Service Desk to AssetCenter.

Management of deletion

It is not allowed within provided liaison implementation to delete items from Service Desk, except Changes and Work Orders within conditions.

Change and work orders should be moved to folder "Deleted" instead of using "Delete" action. By this way, the deletion could then be propagated to AssetCenter.

It is advised to prevent Service Desk users from deleting other items by configuring user profiles (see Service Desk documentation for more information).

When an item has been deleted from AssetCenter, corresponding record is marked as deleted in Service Desk by moving it to a special dedicated folder.

This folder has been named by default "Deleted" and is created when importing ACES configuration file after installation. It must not be removed from Service Desk and changing its name would imply to adapt scenarios.

Moreover, a good practice would be to adapt view in Service Desk to show only items not belonging to "Deleted" Folder so that such items would not be visible to users anymore.

Recommendations on AssetCenter

Bar Code constraints

AssetCenter Bar Code field has been mapped to Service Desk Search Code, which cannot contain spaces nor special characters ("?", "*", "_", "%") nor start with a number.

In consequence, it is expected to ensure that these rules have been respected before exporting any data from AssetCenter to Service Desk.

 **Note :**

In case this constraint could not be respected, it may be possible to map AC Bar Code to SD Source Id if this field has not been used yet.

This would imply to modify scenarios in accordance.

Management of deletion

Workflows are created into AssetCenter during installation to manage Deletion. These call flows should not be removed nor modified, otherwise the deletion will not any more be propagated into Service Desk.

Fields configuration

It is possible to configure each field in AssetCenter, and some of them have a default configuration fixed during the installation of the application

In order to avoid conflicts when mapping objects from Service Desk to AssetCenter, it is recommended to disable the default configuration of some of these scripts :

- The Notified On field present in the Work orders view,
 - The IP address field, in the Network tab of some of Portfolio Items.
- To disable their configuration:
- Click on right button mouse on the concerned field, and Configure object.
 - Then remove script in Default section, and validate the modification.



5 Performing initial propagation

CHAPITRE

Open the **SD2AC_Init.scn** scenario and run it once by clicking **Produce now**. After running it, review the Connect-It log for any errors that may have occurred.

 **Avertissement :**

The best practices mentioned in this guide only deal with implementing AssetCenter when Service Desk is already being used. They do not deal with the case of implementing both applications at the same time.



6 Configuring scheduled services

CHAPITRE

Connect-It enables to associate a Windows service to a scenario. This service allows your Connect-It server to start the data-processing procedure as a background task, depending on the scheduler associated to the scenario. Then these services regularly perform data synchronization between both applications.

Creating a Windows Service is composed of the following steps :

- Create the service
- Configure the service
- Install the service
- Start the service

Connect-It provides a Service Console which is a graphical interface that enables to manage the scenarios and create services to associate to them

To create a new service for a scenario, please refer to Connect-It User's Guide, chapter 4, sections Define a Connect-It service (Windows environment) and Creating a schedule.

Then to manage a scheduled service, you can also refer to Connect-It User's Guide, chapter 5, section Managing the tracking of a scenario using the Service console.



7 | Implementing choices

CHAPITRE

This section focuses on particular choices of implementation that may need to be reviewed and checked against customer IT and Asset management process. The full list of delivered mappings is given in Appendix.

Person and Employees

Telephone

Service Desk 4.5 allows to associate several telephones for each category, while AssetCenter 4.4 handles only one per category.

In consequence, only first phone for each category is exported to AssetCenter.

Address

Service Desk allows to associate several addresses for an employee, including personal one.

In AssetCenter, professional address depends on location and is not tied to employee.

In consequence, the address of the employee is not exported to AssetCenter.

Organization and Company/Department

AssetCenter keeps separated lists of Supplier and Company while Service Desk merges all in Organization.

Target table during export from SD to AC depends on the context: an Organization referenced from a CI will be mapped to a Company categorized as supplier in AC, while it is mapped to Department when referenced from Person.

As a result, it is not possible to have a Department and a Company with the same name in AssetCenter as it would result in conflicts during an export from AC to SD: both would be targeted as Organization, which is the same table in database view.

A second result is that not all Organizations are exported from SD to AC, but only those referenced by CIs or Persons.

Configuration Item and Asset

Initialization step

During first initialization, new Asset Models could be created into AC to reflect SD CI Categories.

SD to AC synchronization

Parent relationship

Service Desk allows to have several parents on a CI while it is not possible on AssetCenter 4.4.

In consequence, synchronization process from SD to AC only exports first parent of the CI.

Mapping of category to model

Service Desk classic data organization is to limit CI categories to a generic naming (ex: Business PC) while AssetCenter classic approach is to include also the model of the asset (ex: "Deskpro EN - DT - PIII 933") to Model field.

So category from Service Desk would likely correspond more to the rank before the last into AC model hierarchy in classic usage of these products, and last name in model full name should comes from another field in SD.

This could be illustrated having a look on AC and SD demo databases:

- In AC demo database, full name for "Deskpro EN - DT - PIII 933" model is "/IT/Workstation/Desktop computer/Deskpro EN - DT - PIII 933".
- In SD demo database, some CIs have a category "Business PC" and "Name 1" field set to "PCKAYAK P3 900 Mhz".

Provided scenario has been designed in this sense: AssetCenter Model is built from Service Desk category (using a map table) and "Name 1" field, according to data organization in SD demo database.

This may need to be adapted if installed Service Desk repository has applied a different approach.

ACto SD synchronization

Mapping of model to category

In order to be consistent with the mapping of Category to Model performed in the context of SD to AC data synchronization (see SD to AC Synchronization above), asset models are exported to CI categories as follows:

- "Name 1" field is filled with last qualifier in AC model full name ("Deskpro EN - DT - PIII 933" in previous example).
- CI category is built according to AC model, taking the hierarchical path of the full name ("/IT/Workstation/Desktop computer" into previous example).

This may need to be adapted if installed Service Desk repository has applied a different approach.

Change/Work Order and Work Order

Service Desk to AssetCenter synchronization

Changes and Work Orders are only created into AssetCenter for TCO calculation purpose, so only a few set of mandatory fields are mapped.

They are located in AssetCenter under a common parent work order which must be created into AssetCenter before performing a synchronization. This common parent has a default name (SD#Changes#Workorders), but it can be modified when it is created in AssetCenter. In this case, do not forget to propagate this modification into the SD2ACWOGeneral map table (see both the AssetCenter to Service Desk and Service Desk to AssetCenter map tables formats section of this document)

formats section of this document) This may be useful to be able to easily make the difference from AssetCenter interface between work orders created from AssetCenter and the ones coming from Service Desk.

AssetCenter to Service Desk synchronization

In the opposite sense, AC work orders are only mapped to Service Desk changes and work orders for fulfillment process. So only items not belonging to common parent (see SD to AC synchronization) are exported with a few set of mandatory fields.

Moreover, only first two levels of AC work orders can be mapped on the Service Desk change / work order model as SD work orders do not implement parent/child relation.

When AC work order creator is a Department, the supervisor of this department is mapped to SD (as SD expects a person to be creator).

As Status is a mandatory field in Service Desk for the Changes and Work orders, a default value taken from the corresponding map table is mapped during export.



8 Business process example: Cost tracking with request fulfillment process

CHAPITRE

It is important to track cost associated with assets. This information can be used to help make appropriate decisions regarding the asset. In situations where a charge back system or billing system is used, the creation of cost records are necessary to track the cost of providing service and to charge the cost to the appropriate business unit.

AssetCenter has the ability to collect expense lines to calculate the Total Cost of Ownership (TCO) of an asset while Service Desk has the ability to collect costs tied to any incident, problem or change. These costs collected by Service Desk can be integrated to AssetCenter in order to create corresponding expense lines.

The following scenario is representative of a business process starting from fulfillment process and integrating SD costs. For more convenience, it is described here from the Demo database provided with AssetCenter.

- In AssetCenter, go to Portfolio/Standard Request , look at the request DEMO-REQ5 (High-end graphics workstation), and in the composition tab, you'll see that this is a Goods and Services request, with one change to be created (installation).
- Go to Procurement/Purchase Request, click New, Fill the purpose field with 'New employee setup', click on the next input, then click on Ok, then choose the 'High-End graphics workstation' in the Service Catalog and click on finish.
- The request should appear in the screen. Change the Req. status to Validated and click execute then Finish.

- This should have created a new work order. In the work order Acquisition tab, you can see the relationship between work order and request.
Note that this is not a full procurement cycle but just a shortcut for the test.
- Once synchronization has been performed from AssetCenter to Service Desk, this work order will have been created on Service Desk.
- Fill information into Service Desk (actual duration, cost, assignment, etc.) and close Change.
- These information will be reported into AssetCenter and will lead to the creation of an expense line, resulting from the actual duration/cost and hourly price.

Another classic business process would be to create an incident with Service Desk, create associated Change and Work Order with all cost related information. These information will be propagated to AssetCenter and would impact the Total Cost of Ownership of the related Asset.



9 | Unsupported features

CHAPITRE

Cross launches

Cross-Launch functionality expects both clients to be installed on same Windows server. Service Desk server must be manually started.

When AssetCenter client is started from Service Desk via a cross-launch, the connection properties to AC database must be manually entered.

If the clients are not running, trying to call a screen of an application from another one will result in an error message.

The locations can't dispose of a cross-launch to SD from AC because Locations are not considered as objects (like CIs for example) and so no action can be associated to them.

As SD Organizations are mapped into AC in a complex way (organizations referenced by CIs or maintenance contracts are mapped as Companies, the ones referenced by Persons are mapped as Departments, and others are not mapped), the corresponding cross-launch has not been added into Service Desk.

Conflict with existing items

Conflicts may appear during an export if data with same key already exists in target database. These conflicts may result either in an error (due to some integrity constraints) or to override items.

As example, CI categories may have a parent which is exported when synchronizing it with AC CI models. If this parent already exists in AC, new CI category may be relocated as a child of this parent instead of a separated one.

Location deletion

A location in Service Desk is only represented by a Code, and consequently it doesn't have a Folder attribute. So if a location is deleted in AssetCenter, this deletion can't be propagated into Service Desk.

Employees/Workgroups relationship deletion

As the deletion can't be propagated from AC to SD, if the relationship between an employee and a group is removed from AC DB, it will not be removed from SD DB.

Maintenance contracts synchronization

AssetCenter and Service Desk don't manage same way the maintenance contracts. In Service Desk, the Name of the contract is used as primary key where as in AssetCenter it's a Reference, and so several contracts can have the same name. Consequently, when a SD maintenance contract is exported in AC, a reference is built thanks to information about it. In the opposite sense, the contract name is composed of its original name plus its reference as it must be unique in SD (the original reference is stocked in the SourceID field).

As a result maintenance contracts can be duplicated in Service Desk since a contract synchronized in the two ways have a different name.

Person synchronization

As the field Name of an Employee is mandatory in AssetCenter, the Last Name field in FullName properties of a Service Desk Person must be filled. Otherwise, the object will be rejected during the initial synchronization from SD to AC.

Error Management

Unprocessed XML files are left into directory when Service Desk server is unreachable (server stopped).

They will be automatically reprocessed prior to next resynchronization if connection to server has been restored.

When an XML file has been processed, it is moved into "Archives" subdirectory, but error(s) occurring during process are not automatically handled (rejected or duplicated record, unexpected status/enumerated value, etc).

In consequence Administrator must check log messages and results after synchronization (available in Service Desk Data Exchange logs for the 4.5 version) and may have to take actions.



10 Scenario Documentation File

SD2AC_Init.scn

CHAPITRE

This appendix presents information about the mapping realized by the scenario **SD2AC_Init.scn**, which corresponds to the initialization phase between Service Desk and AssetCenter.

Tableau 10.1. Information about the views

View Name	Associated Mappings
Global	<ul style="list-style-type: none">■ SD_LOCATIONS / AC_LOCATIONS■ SD_PERSONS / AC_PERSONS■ SD_WORKGROUPS / AC_WORKGROUPS■ SD_DEPARTMENT / AC_DEPARTMENT■ SD_CI_SUPPLIER / AC_COMPANIES■ SD_CATEGORY / AC_MODEL■ SD_TELEPHONES / AC_EMPLOYEES■ SD_MAINTENANCE_CONTRACTS / AC_CONTRACTS■ SD_CONFIGURATION_ITEMS / AC_ASSETS

Tableau 10.2. Connector configuration : Service Desk 4.5

Connector name	Produced document types
Service Desk	<ul style="list-style-type: none"> ■ SD_LOCATIONS ■ SD_PERSONS ■ SD_DEPARTMENT ■ SD_TELEPHONES ■ SD_CI_SUPPLIER ■ SD_WORKGROUPS ■ SD_CATEGORY ■ SD_MAINTENANCE_CONTRACTS ■ SD_CONFIGURATION_ITEMS

Tableau 10.3. Connector configuration : Asset Management 4.4

Connector name	Consumed document types
Asset Management	<ul style="list-style-type: none"> ■ AC_LOCATIONS ■ AC_PERSONS ■ AC_WORKGROUPS ■ AC_DEPARTMENT ■ AC_COMPANIES ■ AC_MODEL ■ AC_EMPLOYEES ■ AC_CONTRACTS ■ AC_ASSETS

Tableau 10.4. Mapping: Locations -> Locations

AC fields	SD fields	Comment
Comment	Remark	
Name	Searchcode	
BarCode	Searchcode	
LocationType	Category	
Parent.BarCode	Parent.Searchcode	
Parent.Name	Parent.Searchcode	

Tableau 10.5. Mapping: Persons -> Employees

AC fields	SD fields	Comment
BarCode	Searchcode	
Name	Last Name	
First	First Name	
FirstName2	Middle Name	
MrMrs	Gender	
bDepartment	0	Boolean defining the type of object to map. For an employee the value is 0, whereas it's 1 for a department.
Comment	Remark	
EMail	E-Mail	
ID #	Identification Number	Boolean defining the type of object to map. for an employee the value is 0, whereas it's 1 for a department.
Field2	Birthdate	
Field3	Status	
Title	Category	
Telephone	Business Phone	
Fax	Fax	
Mobile tel	Mobile	
Home tel	Home	
Icon	images\img16x16\personne .bmp	Chosen icon among these provided in AssetCenter
Parent.Name	Organization.Name1	the parent of an employee is a department
Parent.BarCode	Organization.Searchcode	the parent of an employee is a department
Parent. bDepartment	1	the parent of an employee is a department
Location.BarCode	Location. Searchcode	

Tableau 10.6. Mapping: Workgroups -> Groups

AC fields	SD fields	Comment
BarCode	Searchcode	

AC fields	SD fields	Comment
Name	Name	
bAssignable	Status	Boolean defining the status of a workgroup.
Parent.Name	Parent.Name1	
Parent.BarCode	Parent.Searchcode	
Members.BarCode	Members.Searchcode	Relation between employees and workgroup

Tableau 10.7. Mapping: Organizations -> Departments

AC fields	SD fields	Comment
BarCode	Searchcode	
Name	Name1	
bDepartment	1	Boolean defining the type of object to map. For an employee the value is 0, whereas it's 1 for a department.
Comment	Remark	
Telephone	Business Phone	
Fax	Fax	
E-Mail	E-Mail	
Icon	images\img16x16\service.bmp	Chosen icon among these provided in AssetCenter
Parent.Name	Parent.Name1	
Parent.BarCode	Parent.Searchcode	
Supervisor.BarCode	Manager.Searchcode	
Location.BarCode	Location. Searchcode	

An SD Organization is mapped in AC Departments only if it is referenced by a Person.

Tableau 10.8. Mapping: Organizations -> Companies

AC fields	SD fields	Comment
Code	Searchcode	
Name	Name1	
Qualif1	Category	
Comment	Remark	
Telephone	Business Phone	

AC fields	SD fields	Comment
Fax	Fax	
EMail	E-Mail	
URL	Web page	
Mainsite.BarCode	Location. Searchcode	

An SD Organization is mapped in AC Company only if it is referenced by a CI.

Tableau 10.9. Mapping: CI Categories -> Models

AC fields	SD fields	Comment
Name	Text	
blInvent	0	Determines if BarCode is mandatory or not.
Nature.Code	Text	Association defined in a map table
Parent.Name	Parent.Text	
Parent. blInvent	0	Determines if BarCode is mandatory or not.
Parent.Nature.Code	Parent.Text	Association defined in a map table

Tableau 10.10. Mapping: Maintenance Contracts -> Contracts

AC fields	SD fields	Comment
Reference	OID	built with a common tag and the OID database of the contract
Comment	Description	
End	End date	
Start	Start date	
Init Payment	Price	
IntPayCur	Currency	gives by a map table
Purpose	Name	
Type	4	Always a contract of type 'maintenance'
Status	Status	
Model.Name	'maintenance'	Always the 'maintenance' model used

AC fields	SD fields	Comment
Company.Code	Organization.Searchcode	If a company is referenced by the contract, it is created 'on the fly' as Company into AssetCenter
Company.Name	Organization.Name1	
Company.Qualif1	Organization.Category	
Company.Comment	Organization.Remark	
Company.EMail	Organization.E-Mail	
Company.URL	Organization.Web page	
Company.Mainsite.BarCode	Organization.Location. Searchcode	

Tableau 10.11. Mapping: CI -> Assets

AC fields	SD fields	Comment
BarCode	Searchcode	
AssetTag	Searchcode	
Serial #	Serial Number	
External ID	Name2	
Comment	Remark	
Icon	'images\img16x16\pcs-tatn.bmp'	Chosen icon among these provided in AssetCenter
Acquis.Purchase date	Purchase date	
Acquis.Purchas Price	Price	
Acquis.Procurement.Supplier.BarCode	Supplier.Searchcode	
Maint.Expiration	Warranty date	
Maint.dInstall	Purchase date	
Maint.Maint. Contract.Ref	MaintenanceContract.OID	built with a common tag and the OID database of the contract
Model.Name	Name1	
Model.blInvent	0	Determines if BarCode is mandatory or not.
Model.Brand.Name	Brand.Name	
Model.Nature.Code	Category	Association defined in a map table
Model.Parent.Name	Category.Text	
Portfolio.AssetTag	Searchcode	
Portfolio.Comment	Remark	
Portfolio.Unit	Value Price	
Portfolio.AvgPriceCur	Currency	gives by a map table
Portfolio.Model.Name	Name1	

AC fields	SD fields	Comment
Portfolio.Model.blInvent	0	Determines if BarCode is mandatory or not.
Portfolio.Model.Nature.Code	Category	Association defined in a map table
Portfolio.Model.Parent.Name	Category.Text	
Portfolio.Location.BarCode	Location.Searchcode	
Portfolio.Location.Name	Location.Searchcode	
Portfolio.Icon	'images\img16x16\pcstata.png'	Chosen icon among these provided in AssetCenter
Portfolio.Parent.AssetTag	ParentCI.Searchcode	
Portfolio.Parent.Model.Name	ParentCI.Name1	
Portfolio.Parent.Model.blInvent	0	Determines if BarCode is mandatory or not.
Portfolio.Parent.Model.Nature.Code	ParentCI.Category	Association defined in a map table
Portfolio.User.BarCode	Person	
Portfolio.Supervisor.BarCode	Owner.Searchcode	owner person if defined else supervisor of owner organization
Network.IP Address	IP Address	



11 | Scenario Documentation File

SD2AC_Sync.scn

CHAPITRE

This appendix presents information about the mapping realized by the scenario SD2AC_Sync.scn, which corresponds to the synchronization phase between Service Desk and AssetCenter.

Tableau 11.1. Information about the views

View Name	Associated Mappings
Global	<ul style="list-style-type: none">■ SD_CONFIGURATION_ITEMS / AC_ASSETS■ SD_CHANGES / AC_WORKORDERS■ SD_CHANGES / AC_WORKORDERS2■ SD_WORKORDERS / AC_SUBWORKORDERS■ SD_WORKORDERS / AC_SUBWORKORDERS2■ SD_WORKORDERS / AC_SUBWORKORDERS3■ SD_INCIDENTS / AC_RECONC_PROPOSAL

Tableau 11.2. Connector configuration : Service Desk 4.5

Connector name	Produced document types
Service Desk	<ul style="list-style-type: none"> ■ SD_CONFIGURATION_ITEMS ■ SD_CHANGES ■ SD_WORKORDERS ■ SD_INCIDENTS

Tableau 11.3. Connector configuration : Asset Management 4.4

Connector name	Consumed document types
Asset Management	<ul style="list-style-type: none"> ■ AC_ASSETS ■ AC_WORKORDERS ■ AC_WORKORDERS2 ■ AC_SUBWORKORDERS ■ AC_SUBWORKORDERS2 ■ AC_SUBWORKORDERS3 ■ AC_RECONC_PROPOSAL

Tableau 11.4. ORDER BY clauses by Document type

Document type	WHERE clause	Comment
SD_WORKORDERS	Status	In order to closed the subworkorders and their parent at the appropriate moment

Tableau 11.5. Mapping: CI -> Assets

AC fields	SD fields	Comment
BarCode	Searchcode	
AssetTag	Searchcode	
Serial #	Serial Numbe	
External ID	Name2	
Comment	Remark	
Icon	images\img16x16\pcs-tatn.bmp	Chosen icon among these provided in AssetCenter
Acquis.Purchase date	Purchase date	

AC fields	SD fields	Comment
Acquis.Purchas Price	Price	
Acquis.Procurement.Sup- plier.BarCode	Supplier.Searchcode	
Maint.Expiration	Warranty date	
Maint.dlInstall	Purchase date	
Maint.Maint. Contract.Ref	MaintenanceContract.Sour- celD or MaintenanceCon- tract.OID	if SourceID is filled and begins with the correct tag, the original AC reference is extracted. otherwise the reference is rebuilt with a common tag and the OID database of the contract
Model.Name	Name1	
Model.Brand.Name	Brand.Name	
Model.Nature.Code	Category	Association defined in a map table
Model.Parent.Name	Category.Text	
Portfolio.AssetTag	Portfolio.AssetTag Search- code	
Portfolio.Comment	Remark	
Portfolio.Unit Value	Portfolio.Unit Value Price	
Portfolio.AvgPriceCur	Currency	gives by a map table
Portfolio.Model.Name	Name1	
Port- folio.Model.Nature.Code	Category	Association defined in a map table
Port- folio.Model.Parent.Name	Category.Text	
Portfolio.Location.BarCode	Location.Searchcode	
Portfolio.Location.Name	Location.Searchcode	
Portfolio.Icon	images\img16x16\oc- cas.bmp	Chosen icon among these provided in AssetCenter
Portfolio.Parent.AssetTag	ParentCI.Searchcode	
Port- folio.Parent.Model.Name	ParentCI.Name1	
Portfolio.User.BarCode	Person	
Portfolio.Supervisor.Bar- Code	Owner.Searchcode	owner person if defined else supervisor of owner organization
Network.IP Adress	IP Address	

Tableau 11.6. Mapping: Changes -> Work orders

AC fields	SD fields	Comment
Work order #	SourcelD or ID	if the change originally comes from AC, the SourcelD is mapped, otherwise it's the ID
Expected Resol	Deadline	
Type	0	always 'internal maintenance' type
Title	Description	
Problem	Description	
Icon	images\img16x16\pcstn.bmp	Chosen icon among these provided in AssetCenter
Priority	Priority	
Estimated Cost	Planned Cost	
Init. Cost	Actual Cost	
Planned end date	Planned Finish	
Planned start	Planned Start	
Actual end date	Actual Finish	
Actual start	Actual Start	
Notified On	Creation date	
Requester.BarCode	Requestor.Searchcode	
Tracking.Technician.BarCode	ToPerson.Searchcode	
Tracking.Group.BarCode	ToWorkgroup.Searchcode	
Asset.BarCode	CI.Searchcode	
Asset.Model.Name	CI.Name1	
Parent work order	Default value	each change originally created in SD are mapped under a common parent into AC

Tableau 11.7. Mapping: Work orders -> Sub Work orders

AC fields	SD fields	Comment
Work order #	SourcelD or ID	if the work order originally comes from AC, the SourcelD is mapped, otherwise it's the ID
Expected Resol	Deadline	

AC fields	SD fields	Comment
Type	0	always 'internal maintenance' type
Title	Description	
Problem	Description	
Icon	images\img16x16\ocas.bmp	Chosen icon among these provided in AssetCenter
Priority	Priority	
Estimated Cost	Planned Cost	
Init. Cost	Actual Cost	
Planned end date	Planned Finish	
Planned start	Planned Start	
Actual end date	Actual Finish	
Actual start	Actual Start	
Notified On	Creation date	
Requester.BarCode	Requestor.Searchcode	

Tableau 11.8. Mapping: Parent link between Work orders and Sub Work orders

AC fields	SD fields	Comment
Work order #	SourceID or ID	if the work order originally comes from AC, the SourceID is mapped, otherwise it's the ID
Parent.Work order #	Change.SourceID or Change.ID	if the parent change originally comes from AC, the SourceID is mapped, otherwise it's the ID. If the work order hasn't any parent, it is mapped under the common parent into AC.

Tableau 11.9. Mapping: Incidents -> Reconciliation Proposals

AC fields	SD fields	Comment
Code	SourceID	
Selected Value	Solution	only if the Incident was closed with the code solved

AC fields	SD fields	Comment
Status	Validated or Document rejected	Validated if closed with the code solved, Document rejected if closed with the code Unsolved
Comment	ID	

Two other mappings deals with the deletion of items :

- SD_CHANGES / AC_WORKORDERS2 : manages the deletion of Changes from SD to AC
- SD_WORKORDERS / AC_SUBWORKORDERS2 : manages the deletion of Workorders from SD to AC

To propagate the deletion of these objects, they must not be physically removed from SD, but must be moved into the Deleted Folder. Then these mappings will physically remove the corresponding AC objects from database.



12 | Scenario Documentation File

CHAPITRE | AC2SD_Sync.scn

This appendix presents information about the mapping realized by the scenario **AC2SD_Sync.scn**, which corresponds to the synchronization phase between AssetCenter and Service Desk.

Tableau 12.1. Information about the views

View Name	Associated Mappings
Global	<ul style="list-style-type: none"> ■ Deletions / SD_Deletions ■ Process-end / Events-deletion ■ Locations / SD_Locations ■ WorkGroups / SD_Workgroups ■ Departments / SD_Departments ■ Departments / SD_Departments-phone ■ Departments / SD_Departments-fax ■ Departments / SD_Departments-supervisor ■ Employees / SD_Employees ■ Employees / SD_Employees-phone ■ Employees / SD_Employees-address ■ Employees / SD_Employees-group ■ Employees / SD_Employees-fax ■ Employees / SD_Employees-mobilephone ■ Employees / SD_Employees-homephone ■ Company / SD_Companies ■ Company / SD_Companies-phone ■ Company / SD_Companies-fax ■ Contracts / SD_Contracts ■ Asset / SD_Cls-users ■ Asset / SD_Cls-parents ■ Asset / SD_Cls ■ Workorders / SD_Changes ■ Workorders / SD_Workorders ■ Workorders / SD_Workorders_Cis ■ Reconciliation Proposal / SD_Incidents
Exception	◆ Process-end / Events-deletion

Tableau 12.2. Connector configuration : Service Desk 4.5

Connector name	Produced document types	Consumed document types
Asset Management	<ul style="list-style-type: none">■ Deletions■ Locations■ WorkGroups■ Departments■ Company■ Employees■ Contracts■ Asset■ Workorders■ Reconciliation Proposal■ Employees-Groups	<ul style="list-style-type: none">◆ Process-end / Events-deletion

Tableau 12.3. Connector configuration : Asset Management 4.4

Connector name	Consumed document types
Service Desk	<ul style="list-style-type: none"> ■ SD_Deletions ■ SD_Locations ■ SD_Workgroups ■ SD_Departments ■ SD_Departments-phone ■ SD_Departments-fax ■ SD_Departments-supervisor ■ SD_Employees ■ SD_Employees-phone ■ SD_Employees-address ■ SD_Employees-group ■ SD_Employees-fax ■ SD_Employees-mobilephone ■ SD_Employees-homephone ■ SD_Companies ■ SD_Companies-phone ■ SD_Companies-fax ■ SD_Contracts ■ SD_Cls-users ■ SD_Cls-parents ■ SD_Cls ■ SD_Changes ■ SD_Workorders ■ SD_Workorders_Cis ■ SD_Incidents ■ SD_Employee-Group

Tableau 12.4. Connector configuration : Command line (cleanup)

Connector name	Consumed document types
Service Desk	Command cleanup

Tableau 12.5. Connector configuration : Command line (SD import)

Connector name	Consumed document types
Service Desk	Command SD import

Tableau 12.6. WHERE clauses by Document type

Document type	WHERE clause	Comment
Departments	bDepartment = 1	Selects only the departments as AC stores in the same table the departments and the employees
Employees	bDepartment = 0	Selects only the employees as AC stores in the same table the departments and the employees
Contracts	seType = 4	Selects only the contracts whose type is 'maintenance', since SD manages only this type of contracts.

Tableau 12.7. Mapping: Locations -> Locations

SD fields	AC fields	Comment
Remark	Comment	
Searchcode	Name	
Category	LocationType	
Parent.Searchcode	Parent.BarCode	

Tableau 12.8. Mapping: Groups -> Workgroups

SD fields	AC fields	Comment
Searchcode	BarCode	
Name	Name	
Status	bAssignable	Boolean defining the status of a workgroup.
Parent.Searchcode	Parent.BarCode	

Tableau 12.9. Mapping: Departments -> Organizations

SD fields	AC fields	Comment
Searchcode	BarCode	
Name1		
Remark	Comment	
E-Mail	EMail	
Parent.Searchcode	Parent.BarCode	
Location.Searchcode	Location.BarCode	

Tableau 12.10. Mapping: Departments Phones -> Telephones

SD fields	AC fields	Comment
Organization.Searchcode	Department.BarCode	
Number	Number	
Category	'value of Category'	the value of the category currently mapped
Primary		

Subsidiary mapping to associates the different phone numbers to the departments (same mapping is duplicated two times for each category)

Tableau 12.11. Mapping: Departments Supervisor -> Organizations

SD fields	AC fields	Comment
Department.Searchcode	Department.BarCode	
Supervisor.Searchcode	Supervisor.BarCode	

This mapping defines the relation between the departments and their supervisor.

Tableau 12.12. Mapping: Employees -> Persons

SD fields	AC fields	Comment
Searchcode	BarCode	
Last Name	Name	
FirstName	FirstName	
Middle Name	FirstName2	
Gender	MrMrs	
FullName.Title	MrMrs	

SD fields	AC fields	Comment
Remark	Comment	
Identification Number	ID #	Boolean defining the type of object to map. for an employee the value is 0, whereas it's 1 for a department.
Job title	Title	
Category	'Default category'	value in a map table
Organization.Searchcode	Parent.BarCode	
Location. Searchcode	Location.BarCode	

Tableau 12.13. Mapping: Employees Phones -> Telephones

SD fields	AC fields	Comment
Person.Searchcode	Employee.BarCode	
Number	Number	
Category	'value of Category'	the value of the category currently mapped
Primary		

Subsidiary mapping to associates the different phone numbers to the persons (same mapping is duplicated four times for each category)

Tableau 12.14. Mapping: Employees Address -> Persons

SD fields	AC fields	Comment
Person.Searchcode	Employee.BarCode	
Address type	'Business'	default value
Street1	Employee.Location.Address1	
Street2	Employee.Location.Address2	
City	Employee.Location.City	
ZIP	Employee.Location.ZIP	
State	Employee.Location.State	
Country	Employee.Location.Country.Name	

Subsidiary mapping to associates the business address to the persons

Tableau 12.15. Mapping: Employees Workgroup -> Persons

SD fields	
Person.Searchcode	Employee.BarCode
Workgroup.Searchcode	Employee.Group.BarCode

Subsidiary mapping to associates the workgroup to the persons

Tableau 12.16. Mapping: Companies -> Organizations

SD fields	AC fields	Comment
Searchcode	Code	
Name1	Name	
Remark	Comment	
E-Mail	EMail	
Web page	URL	
Location. Searchcode	Mainsite.BarCode	

Tableau 12.17. Mapping: Companies Phones -> Telephones

SD fields	AC fields	Comment
Organization.Searchcode	Company.BarCode	
Number	Number	
Category	'value of Category'	the value of the category currently mapped
Primary		

Subsidiary mapping to associates the different phone numbers to the companies (same mapping is duplicated two times for each category)

Tableau 12.18. Mapping: Maintenance Contracts -> Contracts

SD fields	AC fields	Comment
Name	Purpose + Reference	to respect unicity of name
Description	Comment	
End date	End	
Start date	Start	
Price	Init Payment	
Status	Status	

SD fields	AC fields	Comment
SourcelD	Reference	common tag + Reference
Organization.Searchcode	Company.Code	If a company is referenced by the contract, it is created 'on the fly' as Company into AssetCenter

Tableau 12.19. Mapping: Assets -> Cls

SD fields	AC fields	Comment
Searchcode	AssetTag	
Name1	Model.Name	
Serial Number	Serial #	
Name2	BarCode + External ID	
Max Installations	Quantity	
Price	Acquis.Purchas Price	
Category	Model.Parent.Name	
Status	Assignement	
Remark	Comment	
IP Address	Network.IP Adress	
Purchase date	Acquis.Purchase date	
Warranty date	Maint.Expiration	
Location.Searchcode	Portfolio.Location.BarCode	
Brand.Name	Model.Brand.Name	
Owner.Searchcode	Portfolio.Supervisor.Bar-Code	Owner Organization if Supervisor is a Department, Owner Person if Supervisor is an Employee.
Supplier.Searchcode	Acquis.Procurement.Supplier.BarCode	
MaintenanceContract.OID Maint.Maint. Contract.Ref	Maint.Maint. Contract.Ref	built with a common tag and the OID database of the contract

Tableau 12.20. Mapping: Assets Users -> Cls Users

SD fields	AC fields	Comment
Cl.Searchcode	Asset.AssetTag	
User.Searchcode	Asset.Portfolio.User.Bar-Code	

this mapping defines the link between the CIs and their users

Tableau 12.21. Mapping: Assets Parent -> CIs Parent

SD fields	AC fields	Comment
CI.Searchcode	Asset.AssetTag	
Parent.Searchcode	Asset.Portfolio.Parent.Asset-Tag	

this mapping defines the relationship between the CIs

Tableau 12.22. Mapping: Work orders -> Changes

SD fields	AC fields	Comment
SourceID	Work order #	
Description	Title	
Priority	Priority	
Status	'Default Value'	Since the status field is mandatory in SD, it is filled with a default value stored in a map table
Classification	Model.Name	Association between Model and Classification in a map table
Category	'Common Category'	The category is a common one, specially created for these kind of workorder to distinguish them
Planned Duration	Planned duration	
Requestor.Searchcode	Requester.BarCode or Requester.Supervisor.Bar-Code	The BarCode of Requester if it's an Employee or Supervisor BarCode if it's a Department
CI.Searchcode	Asset.BarCode	

Only workorders created in AC via a request fulfillment are mapped into SD

Tableau 12.23. Mapping: Work orders -> Work orders

SD fields	AC fields	Comment
SourcelD	Work order #	
Description	Title	
Priority	Priority	
Status	'Default Value'	Since the status field is mandatory in SD, it is filled with a default value stored in a map table
Category	'Common Category'	The category is a common one, specially created for these kind of workorder to distinguish them
Planned Duration	Planned duration	
Requestor.Searchcode	Requester.BarCode or Requester.Supervisor.BarCode	The BarCode of Requester if it's an Employee or Supervisor BarCode if it's a Department
CI.Searchcode	Asset.BarCode	

Only sub workorders created in AC via a request fulfillment are mapped into SD

Tableau 12.24. Mapping: Sub Work orders CI -> Work orders CIs

SD fields	AC fields	Comment
Workorder.SourcelD	Workorder.Work order #	
CI.Searchcode	Workorder.Asset.BarCode	

This mapping defines the link between Work orders and their CI

Tableau 12.25. Mapping: Reconciliation Proposals -> Incidents

SD fields	AC fields	Comment
SourcelD	Code	
Classification	'Reconciliation'	A new Classification must be created into SD
Status	'Registered'	
Description	Name	

SD fields	AC fields	Comment
Information	built string	<ul style="list-style-type: none"> ■ Code ■ Path ■ New Value ■ Previous Value ■ Comment a string including following fields :
Configuration Item		The linked asset (in Computer table)

Tableau 12.26. Mapping: Relationship Employee/Group -> Members of Group

SD fields	AC fields	Comment
Workgroup.Searchcode	Group.BarCode	
Person.Searchcode	Employee.BarCode	

This mapping is used to define the relationship between employees and workgroups.

The mappings :

- Deletions / SD_Deletions
- Process-end / Events-deletion

aims at propagating the deletion of Employees, Departments, Companies, Groups, Assets and Maintenance Contracts, from AC to SD. Let's consider an example to explain its functioning : if an employee is physically deleted from AC, the corresponding SD Person will have 'Deleted' as Folder and will no longer be synchronized.



A | Glossary

ANNEXE

AssetCenter

HP Openview AssetCenter serves as the consolidated asset information hub, providing a complete lifecycle view of the assets. It allows the organizations to manage its assets portfolio and associated fiscal and operational characteristics, in line with business objectives. Sometimes its short name (AC) is used in this documentation.

Connect-It

Connect-It is an EAI (Enterprise Application Integration) type integration platform. An EAI solution enables a company to integrate the different applications from which it can obtain or to which it can provide internal data (Internal support, equipment management software, etc.) or external data (ERP, B2B, B2C).

Scenario

A scenario is the Connect-It concept that enables information to be passed between different external applications. It defines the way to propagate the information.

Service

A service under Windows provides an application that runs as a background task. Connect-It lets the user create and launch a different service for each of his scenarios.

Service Desk

HP OpenView Service Desk automates IT infrastructure management processes to control the quality and delivery of business critical IT services. Sometimes its short name (SD) is used in this documentation.

