

# **HP OpenView Service Desk**

Version 4.0

## **Migration Guide**

Document version 1.3

Technical information in this document is subject to change without notice.

© Copyright 2001 Hewlett-Packard Company

### **Trademarks**

Oracle® is a registered U.S. trademark of Oracle Corporation, Redwood City, California.  
SQL\*Net® and SQL\*Plus® are registered U.S. trademarks of Oracle Corporation, Redwood City, California.

UNIX® is a registered trademark of the Open Group.

### **Restricted Rights Legend**

Use, duplication or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause in DFARS 252.227-7013.

Hewlett-Packard Company  
3000 Hanover Street  
Palo Alto, CA 94304 U.S.A.

Rights for non-DOD U.S. Government Departments and Agencies are as set forth in FAR 52.227-19 (c)(1,2).

## Table of contents

<b>1. Preface</b>	<b>5</b>
1.1 Related Publications	5
<b>2. Differences between ITSM 5.7 and Service Desk 4.0</b>	<b>7</b>
2.1 General	7
2.2 Authorization	9
2.3 System Administration	9
2.4 Organization	9
2.5 Configuration Management	10
2.6 Help Desk Management	11
2.7 Problem Management	13
2.8 Change Management	14
2.9 Work Orders	15
2.10 Service Level Management	16
2.11 Web Connect (replaced by Service Pages in Service Desk)	17
2.12 Software Control and Distribution	19
2.13 Reporting	19
<b>3. Data Migration</b>	<b>21</b>
<b>4. Migration Tools</b>	<b>25</b>
4.1 Data Exchange Process	25
4.2 Migration Task List	26
4.3 Installing Service Desk and Migration Tools	27
4.4 Preparing Service Desk for Migration	28
4.4.1 <i>Checking the Number of User Accounts</i>	28
4.4.2 <i>Adjusting the Regional Settings</i>	28
4.4.3 <i>Adjusting the Time Zone Settings</i>	29
4.4.4 <i>Running the Pre-migration action</i>	30
4.5 Adding the Contact Organization Field	32
4.5.1 <i>Modifying the Export Configuration Files</i>	32
4.5.2 <i>Adapting the Export Configuration Files for Language</i>	35
4.5.3 <i>Migrating ITSM Variable Fields</i>	36
4.6 Creating an ODBC Connection to the ITSM Database	36
4.7 Running pre_check_itsm.sql to prepare ITSM	36
4.7.1 <i>Configuration Item Administrators without a Workgroup</i>	37
4.7.2 <i>Overview priority and impact texts</i>	37
4.7.3 <i>Truncated Fields</i>	37
4.7.4 <i>ITSM Category descriptions</i>	38
4.8 Modifying the Import Mapping	38
4.8.1 <i>Checking the Value Mapping for Locale Codes</i>	38
4.8.2 <i>Mapping Impact and Priority Code Values</i>	39

4.8.3 Migrating Variable Fields .....	40
4.9 Order of Migration .....	44
4.10 Exporting ITSM Data.....	45
4.10.1 Checking for Errors After Exporting .....	46
4.11 Importing ITSM Data.....	46
4.11.1 Verifying the Data Migration.....	47
4.12 Running the Post- Migration Action.....	47
4.13 Parallel import option .....	48
4.13.1 About the Parallel Import Process.....	48
4.13.2 Using The parallel option .....	48
4.13.3 Configuring Script Files.....	49
The Client Session.....	51
4.14 Tips and Troubleshooting.....	51
4.14.1 Memory Problems.....	52
<b>Appendix A - Mapping Overview with Field Lengths .....</b>	<b>53</b>
<b>Appendix B - Detailed Mapping .....</b>	<b>63</b>
<b>Appendix B - .1 Pools .....</b>	<b>64</b>
<b>Appendix B - .2 Accounts.....</b>	<b>64</b>
<b>Appendix B - .3 Organization .....</b>	<b>65</b>
<b>Appendix B - .4 Contacts.....</b>	<b>71</b>
<b>Appendix B - .5 Employees .....</b>	<b>73</b>
<b>Appendix B - .6 CI Relation Types .....</b>	<b>77</b>
<b>Appendix B - .7 Configuration Items .....</b>	<b>77</b>
<b>Appendix B - .8 Services .....</b>	<b>84</b>
<b>Appendix B - .9 Service Calls.....</b>	<b>86</b>
<b>Appendix B - .10 Problems.....</b>	<b>91</b>
<b>Appendix B - .11 Changes.....</b>	<b>96</b>
<b>Appendix B - .12 Work Orders .....</b>	<b>101</b>
<b>Appendix B - .13 Variable Fields Example .....</b>	<b>107</b>

## 1. Preface

This guide explains the migration from IT Service Manager 5.6/5.7 to Service Desk 4.0. If you are using an earlier version of ITSM you will first need to upgrade to 5.6/5.7. With the information in this guide, you can install, configure and perform all tasks to migrate ITSM data.

This guide is intended for IT administrators who will be conducting the data migration. You must have an understanding of the Data Exchange features in Service Desk prior to performing the migration.

This guide is organized as follows:

- Chapter 1, “Preface”, explains the structure of this document and what it is about.
- Chapter 2, “Differences between ITSM 5.7 and Service Desk 4.0”, on page 7, provides an overview of what can be migrated and the differences between ITSM 5.7 and Service Desk 4.0.
- Chapter 3, “Data Migration”, on page 21 contains tables showing an overview of items and the intended migration process for those items.
- Chapter 4, “Migration Tools” on page 25, describes how to use the migration tools and scripts for migrating your data, to include the installation configuration tasks.
- Appendix A, “Mapping Overview with Field Lengths”, on page 53 provides an overview of the mapping from ITSM Tables and fields to the XML classes and attributes, and finally into Service Desk entities(items) and attributes.
- Appendix B, “Detailed Mapping” on page 63 explains the mapping in greater detail to include relations, filters and joined tables.

### 1.1 Related Publications

This section lists publications you may need to refer to when using this migration guide:

- The *Readme.htm* file supplied with the migration software contains information that will help you get started with the migration. It also contains any last-minute information that became available after this document went to manufacturing.
- The *HP OpenView Service Desk: Installation Guide* covers all aspects of installing Service Desk. It also includes information on system requirements. The file name is *Installation\_Guide.pdf*.
- The *HP OpenView Service Desk: Administrator's Guide* contains Service Desk configuration information. The file name is *Administrator's\_Guide.pdf*.
- The *HP OpenView Service Desk: Data Exchange Administrator's Guide* explains how you can use data from other application in Service Desk. It explains the underlying Data Exchange tools used during migration. The file name is *Data\_Exchange.pdf*.
- The *HP OpenView Service Desk: Data Dictionary* contains helpful information about the structure of the Service Desk application. The file name is *Data\_Dictionary.htm*.



## 2. Differences between ITSM 5.7 and Service Desk 4.0

This chapter contains a series of tables organized by ITSM feature. The tables with the heading “Alternative solutions” depict features in ITSM that are not directly migrated into Service Desk. In most instances this is due to differences in system architecture. The Service Desk alternative column in these tables explains how the feature can be implemented in Service Desk, when applicable. Additional tables with the heading “New Features in Service Desk 4.0”, show features that were added after Service Desk 3.0 to improve the migration path for ITSM users:

### 2.1 General

Alternative solutions	
ITSM Description	Service Desk alternative
1. User interface on Unix (Motif UI).	NA
2. BusinessObjects integration for flexible reporting (see also section 2.13).	NA
3. Message of the day.	NA
4. Use of proportional elapsed time (% of open time to target) for progress checks → as in ITSM progress monitor). Use of absolute time is supported.	NA
5. In ITSM the ‘Closed Status’ default setting indicates what the first closed status is. It is used as a filter in all features that automatically display a list of open items (e.g. Open service call help desk). It is also used to automatically enter system date and time in the ‘close date’ attribute of a service call, problem, change, or work order when this status or any status with a higher number is assigned.	In Service Desk a selection of ‘open’ items can be configured in the definition of view filters. The Rule manager can be used to configure a rule that enters the system date/time in the ‘Actual finish’ attribute upon assignment of any particular status to an item..
6. Printing of applied selection values in a report (see also section 2.13).	NA
7. Archiving.	NA
8. Relating multiple items to another item (e.g. service calls to problem) in a single action.	NA
9. Check on remaining open work orders when item (call, incident, problem, change) is closed.	Define a “UI rule” that starts an overview action “Open ...” when item status is changed to ‘Closed’.

Alternative solutions	
ITSM Description	Service Desk alternative
10. Spent time field in progress lines.	NA
11. Updateable who-field for manually entered progress lines (Service Desk automatically generates the value for 'who' from the logged on account for automatic progress lines as well as for manually entered progress lines).	Actions performed by external people or organizations can be registered as workorders.
12. Pop-up messages based on user entries in the interface (open calls, changes, SLA applies etc.).	Define a "UI rule" that starts an overview action when Caller name is entered, for example.
13. Holiday schedule by time zone (Service Desk has global holiday schedule).	NA
14. Synchronization between workgroup and specialist is optional through system setting (making it possible to choose a specialist that doesn't belong to the selected workgroup).	NA
15. Database and OPS\$ user.	NA
16. In Service Desk the functional IDs of the following entities have a length of 9 instead of 10 in ITSM: Servicecall, problem, change, workorder, configuration item	

New features in Service Desk 4.0
Description
1. Use of time zones in target calculations.
2. Global update prices.
3. Second currency (dependant on primary currency).
4. Password settings.

**NOTE:** The implementation of other defaults and automated actions is supported, but handled differently in Service Desk. For example, setting defaults for initial values (like the initial status for service calls) is handled within Service Desk Templates. The Service Desk Rule Manager supports automatic entry of the close date based on entry of a status.



## 2.2 Authorization

Alternative solutions	
Description	Service Desk alternative
1. Menu access.	NA
2. Distinction between “select pools” and “store pools”.	NA

New features in Service Desk 4.0	
Description	
1. Data access authorization based on pools (called folders in SD 4.0).	
2. Data access authorization based on assignment (group & specialist level).	
3. Data access authorization based on status range.	
4. Authorization for assignment of statuses within a particular range.	

## 2.3 System Administration

New features in Service Desk 4.0	
Description	
Session monitor.	
Standard reports: Overview of code tables by module (see also section 2.13).	

## 2.4 Organization

Alternative solutions	
Description	Service Desk alternative
1. Rank field (“position”) for specialist.	Add as custom field.

## 2.5 Configuration Management

Alternative solutions	
Description	Service Desk alternative
1. Bar code inventory processing.	NA
2. Articles not planned.	Templates can be used.
3. Settings for unique numbering.	NA
4. Stock features.	NA
5. Consolidation mode in data exchange (as available in IRM).	NA
6. Type ('Asset', 'Include', 'Logical').	Add as custom field.
7. Default Status (default system setting). Value is copied as default in new, empty configuration items.	Define in template.
8. Service Navigator service tree configuration.	-
9. Standard reports: Quantity by CI classification (see also section 2.13).	Reporting is possible, but does not aggregate to a total for higher category.
10. Standard reports: Overview configuration breakdown (see also section 2.13).	NA

New features in Service Desk 4.0	
Description	
1. Manipulating search codes when generating CIs.	
2. Unique search code.	
3. Functional ID.	
4. Standard reports: CI detail (see also section 2.13).	
5. Standard reports: CI multi-record (see also section 2.13).	
6. Standard reports: Quantity 1. by brand, 2. by name (see also section 2.13).	
7. Standard reports: Overview configuration value (see also section 2.13).	
8. Standard reports: Overview configuration purchase (see also section 2.13).	

## 2.6 Help Desk Management

Alternative solutions	
Description	Service Desk alternative
1. Customer satisfaction.	NA
2. Service call registration timer.	NA
3. Automatic generation of service call number after entry of caller.	Generation of service call number after creation of service call or after saving service call (system setting) is supported.
4. Relating service calls to service calls (relating service calls to incidents is supported).	NA
5. Service call replication.	Depending on the use, alternative solution is possible with service events.
6. View planned progress monitor actions from a service call.	-
7. Desktop beeper.	Service Desk 4.0 has various alternatives for simple notification of newly assigned calls (e.g. using database rules or you can set up Service Desk so that a flashing icon appears in your Windows tray bar with a sound alerting you that you have new unopened items in Service Desk that need your attention).
8. Selection for LOVs and search screens for CIs and services on personal or organizational level (caller field compared to CI user field, etc).	NA
9. Caller type (employee, contact, organization, telephone, location).	Can be added as custom field but without any additional functionality (like adapted selection for search screens or LOVs for caller).
10. Default Status, Category, Medium, and Priority values are copied as defaults in new, empty service calls.	Define in template.
11. Standard call.	Templates can be used.

<b>Alternative solutions</b>	
<b>Description</b>	<b>Service Desk alternative</b>
12. Standard reports: Quantity by CI classification (see also section 2.13).	Reporting is possible, but does not aggregate to a total for a higher category.
13. Standard reports: Progress time by group (see also section 2.13).	NA
14. Standard reports: Open time by priority (see also section 2.13).	NA
15. Standard reports: Open time by category (see also section 2.13).	NA
16. Standard reports: Client Impact (see also section 2.13).	Reporting is possible, but without the minimum number of calls the caller must have made in order to be listed in the report.
17. Standard reports: Open and closed calls (see also section 2.13).	Reporting is possible, but only per status. Does not aggregate to open and closed statuses.
18. Standard reports: SLA impact analysis.	NA

<b>New features in Service Desk 4.0</b>
<b>Description</b>
1. Checklist (general, services).
2. Find matching service calls using the entered description as search criterion.
3. Indicate which service call fields should be copied when a new service call is entered and the contents of an existing service call must be copied into it.
4. Caller and CI details in service call screen.
5. Standard reports: Overview service call detail (see also section 2.13).
6. Standard reports: Overview service call multi-record (see also section 2.13).
7. Standard reports: Overview service calls: by specialist, by support group, and by external provider (see also section 2.13).
8. Standard reports: Overview service calls by CI breakdown (see also section 2.13).
9. Standard reports: Unassigned service calls (see also section 2.13).
10. Standard reports: Service call registration peaks (see also section 2.13).
11. Standard reports: Quantity: by category, by status, by incident code, and by closure code, by impact, and by priority (see also section 2.13).
12. Standard reports: Quantity: by CI supplier, and by CI name (see also section 2.13).

## 2.7 Problem Management

Alternative solutions	
Description	Service Desk alternative
1. Priority – maximum duration.	The objective of problem management is to identify and analyze quality issues in IT services. This is a tactical process to which a fixed priority-duration setting does not apply very well. The deadline set for problem completion, to the point of a solution proposal, will be a result of planning rather than being system generated.
2. Default Status, Category and Priority values are copied as defaults into new, empty problems.	Define in template.
3. Standard reports: Quantity by CI classification (see also section 2.13).	Reporting is possible, but does not aggregate to a total for a higher category.
4. Standard reports: Overview problem with related service calls (see also section 2.13).	NA
5. Standard reports: Open time by priority (see also section 2.13).	NA
6. Standard reports: Open and closed problems (see also section 2.13).	Reporting is possible, but only per status. Does not aggregate to open and closed statuses.

New features in Service Desk 4.0	
Description	
1. Copying field values from service call to problem.	
2. Standard reports: Overview problem detail (see also section 2.13).	
3. Standard reports: Overview problems multi-record (see also section 2.13).	
4. Standard reports: Overview problems: by specialist, by support group, and by external provider (see also section 2.13).	
5. Standard reports: Overview problems by CI breakdown (see also section 2.13).	
6. Standard reports: Unassigned problems (see also section 2.13).	
7. Standard reports: Problem registration peaks (see also section 2.13).	
8. Standard reports: Quantity: by category, by status, by problem code, by closure code, by impact, and by priority (see also section 2.13).	

**New features in Service Desk 4.0**

**Description**

9. Standard reports: Quantity: by CI supplier, and by CI name (see also section 2.13).

**2.8 Change Management**

**Alternative solutions**

**Description**

**Service Desk alternative**

1. Transferred status (system setting) to indicate which status must be transferred to related work orders when the change reaches the status indicated as the closed status for changes.	Define database rule.
2. Reason.	Add as custom field
3. Standard change.	Templates can be used.
4. Default Status, Category, and Priority (default system settings) values are copied as defaults in new, empty changes.	Define in template.
5. Standard reports: Overview change with related service calls (see also section 2.13).	NA
6. Standard reports: Overview change with related problems (see also section 2.13).	NA

**New features in Service Desk 4.0**

**Description**

1. Planned CI location, status, relations and corresponding automatic CMDB update.
2. Outage planning and the related OVO integration.
3. Standard reports: Overview change details (see also section 2.13).
4. Standard reports: Overview changes multi-record (see also section 2.13).
5. Standard reports: Overview changes: by specialist, by support group, and by external provider (see also section 2.13).
6. Standard reports: Quantity: by closure code, by category, by change code, by priority, by impact, and by priority (see also section 2.13).
7. Standard reports: Implementation rate (see also section 2.13).

## 2.9 Work Orders

Alternative solutions	
Description	Service Desk alternative
1. Classification ('Appointments', 'Risk and Impact analysis' and 'Work order').	Use Category field.
2. Standard work order.	Templates can be used.
3. Priority – maximum duration	Work orders are more likely to be scheduled based on: the overall planning of the item, (such as the change, or problem) to which they belong, and their place in the order of execution when there are multiple work orders related to that item. The planning fields 'planned start' and 'planned duration' are therefore more suitable for managing the timely execution of work orders rather than priority.
4. Default Status, Classification, Impact, and Priority (default system settings) values are copied as defaults in new, empty work orders.	Define in template.
5. Sequence number.	Add as custom field.

New features in Service Desk 4.0	
Description	
1. Standard reports: Overview work order detail (see also section 2.13).	
2. Standard reports: Overview work order multi-record (see also section 2.13).	
3. Standard reports: Overview work order 1. by specialist, 2. by group, 3. by company, 4. by contact, 5. by CI (see also section 2.13).	
4. Standard reports: ToDo overview specialist (see also section 2.13).	
5. Standard reports: Implementation rate (see also section 2.13).	

## 2.10 Service Level Management

Alternative solutions	
Description	Service Desk alternative
1. Multiple services per Service Level Agreement (SLA).	Define a new SLA for each service.
2. SLA type (customer or provider).	Add a custom field.
3. Scheduled action (through Progress Monitor) that takes service hours into account.	Rules are used in Service Desk for scheduled actions.
4. Scheduled action (through Progress Monitor) is carried out after a specific amount of time measured in <u>proportional</u> terms has passed (e.g. 10% of the solution time).	Rules are used in Service Desk for scheduled actions but they only allow for <u>absolute</u> time
5. SLA Wizard.	NA
6. SLM Evaluation report.	NA
7. Checklist for combination of receiver and service.	NA
8. Service hours and impact-priority rules associated to an SLA separately. More than one impact-priority rule can be combined with the same service hours.	In Service Desk service hours and impact-priority rules are combined in one association. Migrate only the service hours and add impact-priority rules manually, extend the set of service levels when necessary and update SLAs that need to reference a newly added service level. Manually add cost for each service.
9. Days notice (system setting). Indicates how many days before the expiration of an SLA the system should start displaying a message.	Define database rule.

New features in Service Desk 4.0	
Description	
1. Checklist for services.	
2. Assignment fields (company, contact, group, specialist)	



## 2.11 Web Connect (replaced by Service Pages in Service Desk)

Before migrating, we advise you to investigate your Web Connect usage pattern in order to assess whether Service Pages or a full client installation covers your requirements when migrating to Service Desk.

Web Connect was designed to accommodate both performance over low bandwidth, as well as thin client requirements for users with limited functional requirements and limited application usage frequency.

The Service Desk full client, as the table below shows, covers most Web Connect functions. Service Desk performs well over low bandwidth, providing a fully functional Service Desk application and thus providing an excellent migration path for Web Connect users. For customers who need all of the features in Web Connect, we advise using the full client of Service Desk.

For specialists who use Web Connect predominantly to log calls, modify assigned calls, or incidents on remote sites on an infrequent basis, Service Pages might be a good solution. However, not all Web Connect features are available in Service Pages:

Alternative solutions	
Description	Service Desk alternative
1. Search service calls (End user option).	Search features allow you to query Frequently Asked Questions (FAQs) for specific text (in the Description and Information fields). In Service Pages, FAQ's are used as a knowledge base.
2. Extended search, service calls (Specialist option).	Suggest migrating to full client
3. ConText search option (Specialist option).	NA Service Desk does not support this feature because of its platform independent architecture.
4. View personal CIs.	When logging on to Service Pages, the CI field will show the end user CI names without details. Verification of CIs will be possible to this extent.
5. Help option.	Links to help content can be defined with URLs in Service Pages.
6. About option.	NA
7. Beeper screen.	NA

<b>Alternative solutions</b>	
<b>Description</b>	<b>Service Desk alternative</b>
8. Message option	The definable URL's in Service Pages can be used to create messages for users.
9. Print option	NA
10. Create, view and modify work orders in the service call screen.	To view and modify work orders, select the work order menu.
11. User can set maximum number of records displayed on one page.	NA
12. Use of Variable fields	Use full client

<b>New features in Service Desk 4.0</b>	
<b>Description</b>	
1. 'Service request'. (End user option). The following fields in Web Connect are not available in SP: status and caller + caller details.	
2. 'Service call entered by...' (Specialist option). The following fields in Web Connect are not available in SP: status, caller + caller details, caller location, caller telephone, CI details, incident code, pool and close code or the overview of CIs.	
3. 'View service calls'. (End user option). No search criteria available: the system administrator defines the underlying query within Service Desk.	
4. 'View/modify service call'. (End user option). The following fields in Web Connect are not available though in SP: caller + details.	
5. 'Open service calls'. (Specialist option). No search criteria available: the system administrator defines the underlying query within Service Desk.	
6. 'View/modify service call'. (Specialist option). The following fields in Web Connect are not available in SP: caller + details, caller location, caller telephone, CI details, incident code, pool and close code, or the overview of CIs.	
7. 'Open changes'. No search criteria available, the system administrator defines the underlying query within Service Desk.	
8. 'Viewing open changes'.	
9. 'Open work orders'. No search criteria available, the system administrator defines the underlying query within Service Desk	
10. 'Viewing and modifying work order details'. The following fields in Web Connect are not available in SP: classification, pool, and closure code.	
11. Subject and a text search options extend the FAQ feature, already available with 3.0, to include explorer navigation. Free text searching in the database, as was possible with Web Connect, is not a good solution for performance reasons.	
12. Password and time zone settings.	

New features in Service Desk 4.0
Description
13. View and add progress lines.

## 2.12 Software Control and Distribution

This module will not be implemented. Most of the features in this module will eventually be replaced by extending change management, configuration management and integrations to include software distribution tools.

## 2.13 Reporting

Different customers have different reporting needs. Service Desk 4.0 offers flexible reporting to meet those needs. In contrast to ITSM, a standard reporting tool containing examples and a report viewer is not provided with Service Desk 4.0. Instead two types of views are supplied for creating reports: Console views making it possible to create basic reports, containing the data from one item per view and its relations (Person – workgroup for example); and database views for creating complex reports that combine information from different items.

The console view is a powerful tool to create basic reports. These reports can be used to group, and filter information. Table views and explorer views offer different ways of grouping the information. Chart views and card views offer different ways of presenting the information. Not all standard reports supplied with ITSM can be duplicated with console views.

Database views are created with ITSM's standard reports – and more – in mind. Database table views can be used to simplify your table structure without the complex task of joining tables. Analyzed table views can be used to quickly summarize data from history lines, shortening the otherwise lengthy analysis process. The information gathered by these views can then be formulated into a report by a reporting tool, such as BusinessObjects or Crystal reports.

Database views in Service Desk 4.0 can be localized. After implementing Service Desk, you can generate database views and build reports based on those views. The views are generated in the language that was used on the Service Desk client. As a result, you can choose the language of your report by choosing the language on the client.

Because BusinessObjects is no longer supplied as the standard reporting tool, there are no universes. Database views must be accessed in the Service Desk database. Security, and authorization features are not supplied with the views. Security must be set at the database level. Authorization must be set using the report definitions in your third-party reporting tool.

Specifically for BusinessObjects users:

For standard reports a supplier providing migration solutions for BusinessObjects reports has been identified. Please contact your BusinessObjects supplier for details. Also, contact your BusinessObjects supplier for details on migrating BusinessObjects licenses from ITSM to Service Desk.



### 3. Data Migration

The following criteria were checked to determine if automatic migration of data from ITSM to Service Desk is possible and sensible:

- The data items to be migrated from ITSM must exist in Service Desk.
- A clean migration must be possible (no or little need for post-migration restructuring of data).
- The automatic migration must not have a negative impact on the usability of Service Desk features.

The tables on the following pages provide an overview of items in ITSM and the intended migration path of those items. A Yes or No in the Migr column is used to identify items that can or cannot be directly migrated. Comments and advice for the migration of each item are provided whenever possible:

**Table 1 - System Data**

System data	Migr.	Comment	Advice
Profiles	No	Profile (called roles in Service Desk) structure in Service Desk is hierarchical. Migrating profiles from ITSM would force degraded use of profiles in Service Desk.	Consider the possibilities of the new structure and manually enter new roles in Service Desk after migration.
Profile settings (authorization and defaults)	No	Many settings are specific to Service Desk and they are administered in relation to the new “roles” structure. Partly migrating and then adjusting would be more work to the customer than adding roles and settings manually.	Add roles and settings manually after migration.
Codes	Yes	Service Desk has several hierarchical code lists which are flat lists in ITSM (e.g. service call category). These code lists will be migrated to flat code lists in Service Desk, and can be reorganized to hierarchical code lists after migration (without deleting codes). This will not impact the references made to these codes from service calls, changes, etc.) In Service Desk, impact and priority codes use the same value list for service calls, incidents, problems and changes. Therefore, a 1:1 migration of these codes is not advised.	For impact and priority codes; new, general codes should be entered in Service Desk before migration and then the respective lists in ITSM should be mapped to these new values.
Default values	No	New “Template” capabilities.	Manual entry in templates after migration.

System data	Migr.	Comment	Advice
Service Hours and Free Days	Yes	In Service Desk 4.0, service levels can be defined for multiple 'service hour windows' within the same SLA. This addresses the fact that the importance of services may vary over time.	-
Variable Field definition	No	In Service Desk, variable field definitions are category-based for all items, not just CIs. Some variable field formats are not supported yet; like upper, lower and money.	When ITSM variable fields do not have an equivalent standard field in Service Desk, new fields must be created in Service Desk before migration, to which the ITSM fields can be mapped.
Customer Satisfaction	No	Not included in Service Desk 4.0.	-
Checklist	Yes	-	-
Languages, messages and window titles	No	ITSM specific text, like messages, window and field titles are not migrated, because they are not relevant to Service Desk. System administrators can perform Code translations in Service Desk 4.0.	-
Pools	Yes	Pools are called Folders in Service Desk 4.0	In Service Desk 4.0, folders are applicable to people and organization data. Customers should consider whether there are attributes (like organization or location) that can be used to map against a folder value.
Query Restriction	No	New architecture with default query restriction. In ITSM this was possible per item for each user profile.	Set query restriction in Service Desk manually.

**Table 2 - Operational Data**

Operational data	Migr.	Comment	Advice
Persons	Yes	-	-
Organizations	Yes	-	-
Work orders	Yes	Spent time, added in ITSM 5.7.3, is not migratable.	-
Changes	Yes	-	-
Problems	Yes	-	-
Service calls	Yes	Service call to service call relations cannot be migrated. This relationship is not supported by Service Desk 4.0.	-
CMDB	Yes	-	-
CI Barcodes	No	Bar code inventory taking is not supported in service Desk 4.0.	-
CI stock	No	The ITSM features supporting issue, intake and replacement of CIs are not supported in service Desk 4.0.	-
Services	Yes	-	-

Service Level Agreements	Yes	In ITSM an SLA can contain multiple services. In Service Desk an SLA contains one service. For example, an SLA with 5 services will generate 5 SLAs, each with one service within Service Desk.	-
Software Control and Distribution	No	SC&D features are not implemented in Service Desk 4.0	-
Variable Field values	Yes	Need to be mapped to Service Desk (standard or custom) fields.	-
Attachments	No	New architecture	Potential workaround: place attachments in central directory and link using smart link.
Archived data	No	New architecture	Assess the feasibility of retaining a small ITSM environment to enable access to archived ITSM data.

**Table 3 - Standard Entities**

Standard entities	Migr.	Comment	Advice
Standard Service Call	No	Not supported.	After migration use templates in Service Desk.
Standard Change	No	See previous.	-
Standard Work Order	No	See previous.	-
Configuration Item Template	No	See previous.	-
Configuration Item Article	No	Service Desk does not include an article item.	After migration use templates in Service Desk.

**Table 4 - Profile Security**

Profile Security	Migr.	Comment	Advice
User profiles (incl. module, mode, menu, item access)	No	New security architecture based on roles. There are too many differences between user profiles and roles to enable a timesaving conversion. Adjusting afterwards is likely to be a more tedious task than setting up new roles.	Set up user profiles and settings in Service Desk manually.

**Table 5 - Reports**

Reports	Migr.	Comment	Advice
Standard reports	Yes/ partly	New architecture based on views. Service Desk 4.0 covers about 70 % of the standard reports available in ITSM.	-

Reports	Migr.	Comment	Advice
Flexible reports	No	The BusinessObjects integration with specifically designed universes for ITSM will not be continued, because it does not fit the new architecture of Service Desk. Support of multiple databases and XML for data exchange are the primary factors in discontinuation.	The alternative offered is extended internal reporting capability, which offers more extensive query features and an object model – database table relationship description to support users of any reporting tool that can operate on relational databases (Oracle and SQL Server). The capability for statistical reporting has improved in Service Desk 4.0, compared to Service Desk 3.0.

**Table 6 - Integrations**

Integrations	Migr.	Comment	Advice
User applications	No	-	Set up links to external applications after migration by defining 'Smart actions'.
IRM	No	New architecture.	Set up data exchange after migration.
EIDK	No	New architecture.	Set up service events after migration.
Sites	No	ITSM Service Call replication.	Re-configure with service events.

**Table 7 - Actions**

Actions	Migr.	Comment	Advice
Progress Monitor	No	New architecture.	Set up database rules after migration.
Escalation	No	New architecture.	Set up actions with database rules after migration.
Task Manager events, actions and conditions	No	New architecture.	Set up rules with Rule Manager manually after migration.

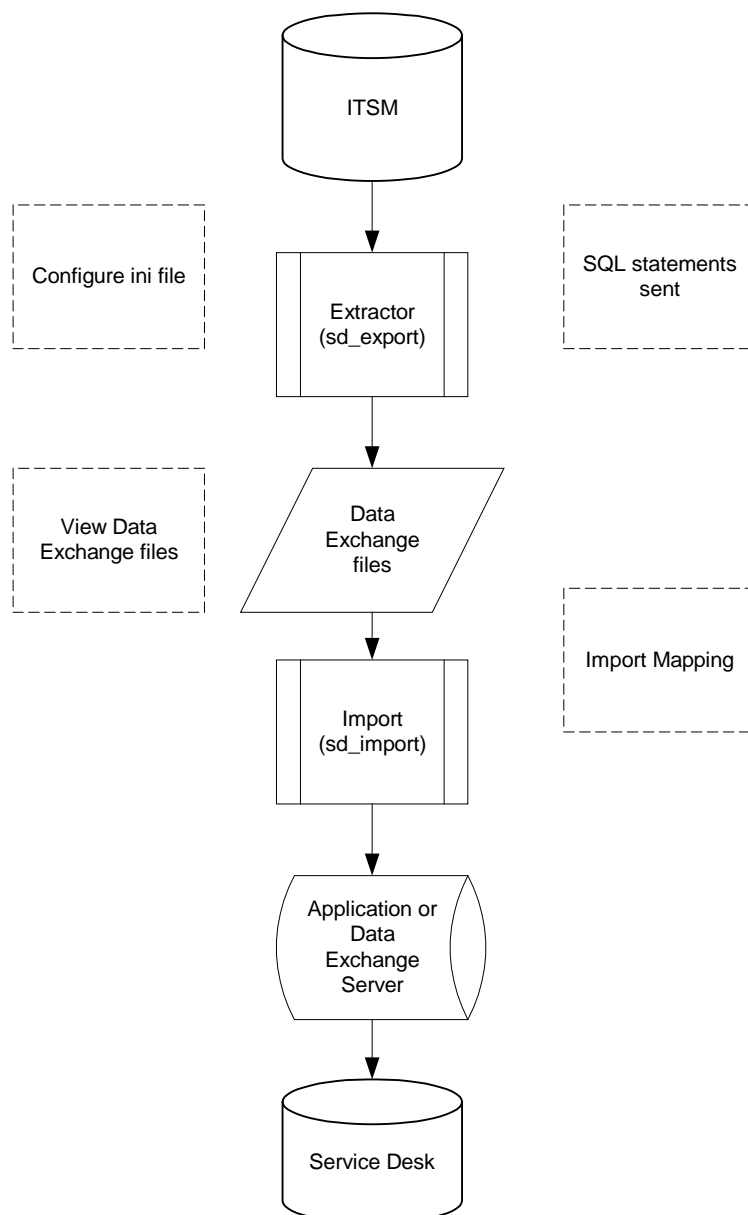


## 4. Migration Tools

The Data Exchange features in Service Desk make it possible to export your ITSM data and import it into Service Desk. The process and tools used for migration of your ITSM data are explained in the following sections. This chapter also describes the configuration tasks you will need to perform in ITSM and Service Desk.

### 4.1 Data Exchange Process

Data Exchange is the process of exporting information from a data source, formatting it and then importing it into Service Desk. A configurable extractor is used to export the data into a data exchange file in extensible markup language (CIM-XML).



The exchange file can be viewed in an object-tree format. This makes it possible for you to verify that the data was exported correctly before you import it into Service Desk. If the data was not exported correctly you can adjust the configurable extractor and export the data again.

When the command is given to import the data, the import mapping is applied and the data in the exchange file is reorganized then imported through the server into the Service Desk database.

The migration process is divided into a series of tasks. Each task has its own configurable extractor used to export the data and its own import mapping for importing the data. For example, one task is to export then import configuration items, while another task is to export and then import service call data. This makes it easier to manage the data and the order in which the data is imported.

## 4.2 Migration Task List

The following task list shows the major steps required when migrating from ITSM to Service Desk:

1. Identify what you will migrate, see “Differences between ITSM 5.7 and Service Desk 4.0” on page 7, and “Data Migration” on page 21 for additional information
2. Upgrade ITSM to the correct version, including any necessary service packs.
3. Install Service Desk with an empty database.
4. Install Integrations, with Data Exchange and Migration options selected.
5. Install other Service Desk items as needed, for example: Service Pages, and agents.
6. Install Service Desk license key.
7. Identify the number of user accounts.
8. Set the Time Zone settings.
9. Add the Contact Organization field to the service call form (optional).
10. Modify the export configuration files to reflect the correct username and password for each migration task.
11. Modify the export configuration files for language (optional)

12. Configure the export configuration file ITSM\_Varfields.ini to export your variable fields (optional)
13. Set up SQL\*Net on the application server so it can find the ITSM database.
14. Adjust the regional setting for dates.
15. Establish an ODBC connection for the ITSM database.
16. Update your Service Desk data.
17. Update your ITSM data using the pre\_check\_itsm.sql script.
18. Add configuration item administrators to workgroups in ITSM.
19. Run the script auth.sql on the ITSM database to determine mode access (new, modify, view, delete) for each profile, a file called auth.lst will be generated.
20. Run the Pre\_migration action on the Service Desk database and check dbconfconsole.log for errors.
21. Modify the import mapping for each migration task
22. Export your ITSM data.
23. Examine the XML file and the log files.
24. Import your ITSM data in the order specified.
25. Run the Post\_migration action on the Service Desk database
26. Complete configuring Service Desk. See the *HP OpenView Service Desk: System Administrator's Guide* for more information.

### **4.3 Installing Service Desk and Migration Tools**

To install the migration:

1. Install the Service Desk application server. For installation instructions refer to the *HP OpenView Service Desk 4.0 Installation Guide*.
2. Install an “empty” Service Desk database.
3. Install Integrations. Perform a custom installation and select Data Exchange and Migration from the list of optional integration components

The export configuration files (ITSM\_xxx.ini files) will be automatically installed in: *Service Desk Path\Server\data\_exchange\config*. The default location is: *Program Files\Hewlett-Packard\OpenView\ServiceDesk 4.0\Server\data\_exchange\config*

## 4.4 Preparing Service Desk for Migration

This section explains the configuration tasks you need to perform in Service Desk to prepare the application for migration.

**NOTE:** To speed up the migration, turn off the auditing rules and the analyzed data options in Service Desk. If you do not, unnecessary information will be put in the history lines, slowing down the import considerably.

To turn off auditing rules: from the Administrator Console in your Service Desk application, select Security, then Audit, and then Audit Rules. Open each item, select the **Do Not Audit** option, and then **OK**.

To turn off the analyzed data option: from the Administrator Console open the system panel, select Report Settings, open the Enable ‘Generate Analyzed Data’ for items tab, clear all item check boxes and click **OK**.

### 4.4.1 Checking the Number of User Accounts

With Service Desk 4.0, all accounts are migrated as concurrent users. You only need 1 concurrent user license for migration in addition to the correct module licenses. For more information on licensing, refer to the *Service Desk Installation Guide*.

To see the number accounts you have, `pre_check_itsm.sql` generates a report containing information about your licensed accounts. The “xx” in the following lines represents the number of concurrent users:

```
“Check number of Service Desk user accounts”  
“Mininum number of Licenses needed”  
“xx”
```

**NOTE:** All migrated accounts will be given the password “*ITSM*”.

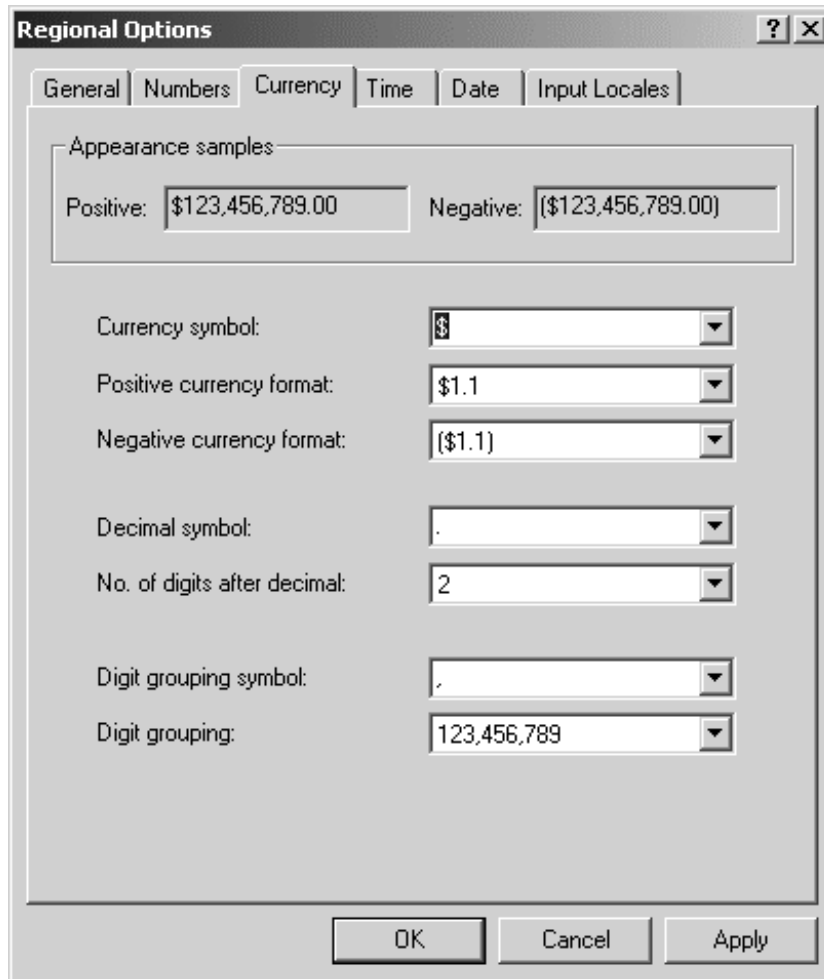
### 4.4.2 Adjusting the Regional Settings

To ensure that your dates are imported into Service Desk in the correct format, you need to set your Windows NT regional settings to *English* so that dates can be migrated correctly. The date settings should be set to MM/dd/yyyy. You will also need to set your currency settings to the US standard. After migration you can return the settings to their original configuration. To adjust your regional settings:

1. From your Windows Start button, select Settings, then Control Panel.

2. From the Control Panel, double-click the Regional Settings icon.
3. Set the regional setting to **English(United States)**, (for Windows 2000 users this is set in the General tab.).
4. Select the Currency tab and adjust the currency to the following (US English) standard format:

**Figure 1 - Currency Format**



5. From the Date tab use the drop-down arrow to select **MM/dd/yyyy** in the Short date style field.
6. Click **OK** to finish.

**NOTE:** No currency is specified at database level. If you have specified all values in one currency in ITSM you must use the same currency in Service Desk for migration.

#### 4.4.3 Adjusting the Time Zone Settings

Set the time zone setting on the Service Desk application server you are using for migration, to the same time zone as your ITSM database. To change the time zone on your application server:

1. Double-click the clock in your task bar and select the **time zone** tab.
2. Select the same time zone you are using for your ITSM database.

The import process will use local time for all time fields in the XML files. It will then use the time zone settings of the computer you are using to adapt the data to database GMT values. The time zone of your ITSM database and your PC need to be the same before you perform the import process.

#### 4.4.4 Running the Pre-migration action

1. The `pre_migration` action must be run on the Service Desk database before migration can be started. The script inserts a number of necessary elements and configures settings..

**NOTE:** All errors will be logged in the `dbconfconsole.log`. This file will be created in the location you run the script from.

The script performs the following tasks:

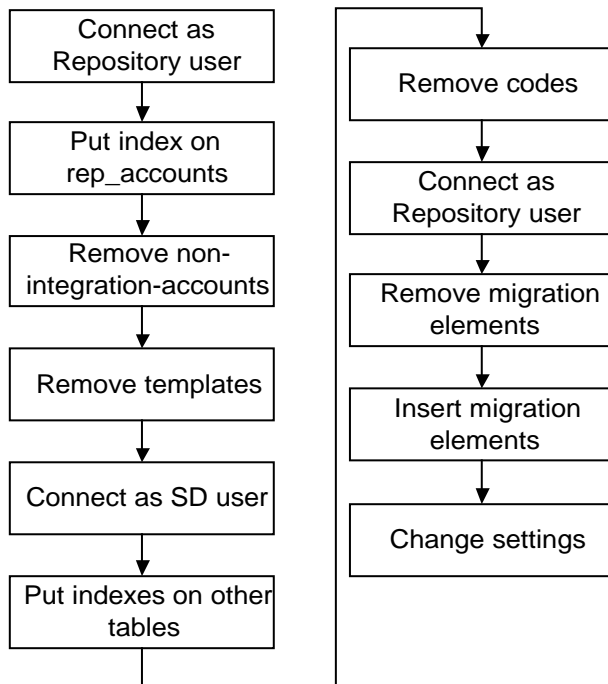
- Creates indexes to improve the migration's performance on the following tables:
  - `rep_accounts` (on `acc_loginname`)
  - `itsm_persons` (on `per_sourceid`)
  - `itsm_configuration_items` (on `cit_sourceid`)
  - `tsm_services` (on `srv_sourceid`)
  - `itsm_organizations` (on `org_sourceid`)
  - `itsm_workgroups` (on `wog_sourceid`)
- Removes all non-integration accounts except "system" and "mailmanager"
- Removes all templates for:
  - Changes
  - Service calls
  - Work orders
  - Problems
  - Services and Configuration items except default templates and templates for DTA, NNM, PSP or SMS.
- Removes all codes for the following entities:
  - Pool,
  - SE Relation Type
  - Brand
  - CI Category, CI relation type, CI status
  - Location
  - StatusService
  - Sc. Category, Sc. Closure code, Service Call classification, Servicecall status

- Medium,
- Problem category, Problem classification, Problem status,
- Pro. Closure code
- Change category, Change status, ChangeClosureCode, ClassificationCha
- Category workorder, Wor. Closure code, and Workorder status
- Free Days records
- Removes previous inserted migration elements (all elements starting with ITSM), in case this is an update:
  - migration account
  - migration Templates
  - migration Import mappings
  - migration Data exchange tasks
  - migration External entities
  - migration External attributes
  - migration External attribute mappings
  - migration Value mappings
- Inserts the new migration elements:
  - migration account
  - migration templates
  - migration Import mappings
  - migration Data exchange tasks
  - migration task group
  - migration external entities
  - migration external attributes
  - migration external attribute mappings
  - migration value mappings
  - an UNDEFINED ci relation type with a reverse relation to itself
  - a Done assignment status
- Changes the following settings:
  - Switches the Caller field of Service Calls to “not mandatory”.
  - Switches the Organization field of Persons to “not mandatory”.
  - Switches the City of an address to “not mandatory
  - Switches the Password field of Accounts to “not mandatory
  - Switches the ID After Creation setting to “off”.

- Activates the servicecall custom field “Ser.Shorttext1” and call it “Contact Organization”,
- Activates the CI user organization custom field.

The following diagram shows the flow of the pre-migration sql script:

**Figure 2 - Pre-migration Script**



## 4.5 Adding the Contact Organization Field

If you import servicecall.callername2 data and want to view it in Service Desk, you will need to add the field called Contact Organization to the service call form in Service Desk. For migration the ITSM servicecall.callername2 field is mapped the **Contact Organization** field of a service call item in Service Desk. This field is available in Service Desk but needs to be manually added to the service call form in Service Desk 4.0

Information on creating custom fields is available in the Service Desk Online help.

### 4.5.1 Modifying the Export Configuration Files

A series of configuration files are provided for exporting data from ITSM. The configuration files need to be adapted to fit your environment. The configuration files contain information necessary to extract the data from the data source. The configuration files map the ITSM entities and attributes into a format meeting CIM-XML standards that can be imported into Service Desk.

All of the migration configuration files follow the naming convention: ITSM\_\*.ini and are installed by default in:

*Program Files\Hewlett-Packard\OpenView\ServiceDesk 4.0\Server\data\_exchange\config*



To open the configuration files:

1. From the **Tools** menu, select **System**.
2. From the Administrator Console, open the **Data** folder and then **Data Exchange**.
3. Select the migration task you want to use, and click **Edit** in the Data Exchange dialog box.

The names between [] brackets in the configuration file are the entities you will be extracting. ATT and PARENT\_RELATION\_NAME in the configuration file determine the fields that will be exported. The field names are mapped to attributes of Service Desk items in the import mapping.

The following example will be used to explain the different parts of the configuration file:

```
[DSN]
NAME=migration
USR=itsm_user
PWD=itsm_pwd

[SYSTEM]
LOG=TRUE
XML=TRUE
TXT=FALSE
LOG_FILE=C:\ITSM_Organization.log
OUTPUT_FILE=C:\ITSM_Organization.txt
XML_OUTPUT_FILE=C:\ITSM_Organization.xml
APPLICATION_NAME=ITSM

[CLASSES]
NAME= CL_CODE_WORKGROUP, CL_CODE_ORG_LOC, CL_EXT_ORGANIZATION,
      CL_EXT_ORGANIZATION_RELATION,

-----
-- WORKGROUP CODES --
-----
[CL_CODE_WORKGROUP]
SOURCE=      [APP_CODE_TEXTS]
ATT=         [TEXT], [SEARCHCODE], [SOURCE_ID]
COLUMNS=    [APP_CODE_TEXTS].[COX_TEXT] AS [TEXT],
              REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(
                DECODE(LTRIM(SUBSTR([COX_SEARCHCODE],1,1),
                '0123456789'),NULL,DECODE([COX_SEARCHCODE],NULL,
                NULL,CONCAT('#',[COX_SEARCHCODE])),[COX_SEARCHCODE]),
                '*','-' ),'?','-' ),'_','-' ),'%','-' ),' ','-' ) AS [SEARCHCODE],
              [APP_CODE_TEXTS].[COX_COD_ID] AS [SOURCE_ID]
CONDITION=   [COX_COD_ID] BETWEEN 900000 AND 999999 AND [COX_LNG_ID] =
              'GB'
ORDERBY=     [COX_COD_ID]
LOADTABLE=   TRUE

-----
-- ORGANIZATION LOCATION CODES --
-----
```

```

[CL_CODE_ORG_LOC]
SOURCE=      [APP_CODE_TEXTS]
ATT=        [TEXT]
COLUMNS=
    REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(
        DECODE(LTRIM(SUBSTR([COX_TEXT],1,1),
        '0123456789'),NULL,DECODE([COX_TEXT],NULL,NULL,
        CONCAT('#',[COX_TEXT])),[COX_TEXT]),
        '*','-' ), '?','-' ), '_','-' ), '%','-' ),
        ' ','-') AS [TEXT]
CONDITION= [COX_COD_ID] BETWEEN 800000 AND 899999 AND [COX_LNG_ID] =
    'GB'
ORDERBY=    [COX_COD_ID]
LOADTABLE=  TRUE

```

```

-----
-- EXTERNAL ORGANIZATIONS --
-----

```

```

[CL_EXT_ORGANIZATION]
SOURCE=      [ADDRESS]
ATT=        [EMAIL], [NAME1], [NAME2], [REMARK], [SEARCHCODE],
            [ORG_ID], [CATEGORY], [STATUS]
COLUMNS=    [ADDRESS].[EMAIL] AS [EMAIL], [ADDRESS].[NAME1] AS
            [NAME1], [ADDRESS].[NAME2] AS [NAME2], [ADDRESS].[REMARK] AS
            [REMARK],
            REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(
                DECODE(LTRIM(SUBSTR([SEARCHCODE],1,1),
                '0123456789'),NULL,DECODE([SEARCHCODE],NULL,NULL,
                CONCAT('#',[SEARCHCODE])),[SEARCHCODE]),
                '*','-' ), '?','-' ), '_','-' ), '%','-' ),
                ' ','-') AS [SEARCHCODE],
            CONCAT('EO',[ADDRESS].[ID]) AS [ORG_ID], 'EXTERNAL' AS
            [CATEGORY], 'ACTIVE' AS [STATUS]
CONDITION= [ADDRESS].[SUB_TYPE]=1
LOADTABLE=  TRUE

```

```

-----
-- EXTERNAL ORGANIZATIONS RELATIONS --
-----

```

```

[CL_EXT_ORGANIZATION_RELATION]
SOURCE=      [ORGANIZATION]
ATT=        [ORG_ID], [PARENT_ID]
COLUMNS=    CONCAT('EO',[ORGANIZATION].[ADDRESS]) AS [ORG_ID],
            CONCAT('EO',[ORGANIZATION].[PARENT]) AS [PARENT_ID]
CONDITION= [ORGANIZATION].[PARENT] IS NOT NULL
LOADTABLE=  TRUE

```

#### [DSN]

DSN is the data name source. Under this header the data source will be defined. The NAME is used to specify what ODBC data source will be used and, depending on the source a username and password are needed for the owner of the database. The ODBC data source name used in the configuration file must be the same as the one used when you make your ODBC connection.

#### [SYSTEM]

Under the system header the settings and data files are defined. The first four headers are used to specify if a log file will be created (LOG=TRUE), if the output file is of the XML type (XML=TRUE) or text (TXT=FALSE) and if a dump file should be created (DUMP=TRUE).

**[CLASSES]**

The classes section is where the entities are defined that will be exported. The class name is important later on for the import mapping. In the example the class names are CL\_CODE\_WORKGROUP, and CL\_CODE\_ORG\_LOC, CL\_EXT\_ORGANIZATION, and CL\_EXT\_ORGANIZATION\_RELATION.

For each class defined in the section [CLASSES] a definition of the content must be provided. The destination of the data must be given via the attributes (ATT) that will be stored in the XML output file. The attributes can literally be the same as the columns or aliases. The attributes will be captured from the columns that are written in the COLUMNS section.

For additional information on configuring the extractor, see the section “Configuring the Extractor” in the *HP OpenView Service Desk: Data Exchange Administrator’s Guide*.

**4.5.2 Adapting the Export Configuration Files for Language**

If you want to migrate code tables from ITSM that contain another language ID, you will need to modify the export configuration files by replacing **GB** wherever it occurs in the configuration file with the appropriate language ID. In the export configuration files, code tables are retrieved with conditions that include a language identifier, for example [X].[Y]=GB. The default code identifier is **GB** for English. Only the code tables with that language identifier will be retrieved. The following example shows a small portion of the Service Call export configuration file with the language identifier in bold:

```
-----
-- SERVICE CALL CATEGORY CODES --
-----
[CL_CODE_SC_CAT]
SOURCE=      [APP_CODE_TEXTS]
ATT=         [TEXT]
COLUMNS=    [APP_CODE_TEXTS].[COX_TEXT] AS [TEXT]
CONDITION=   [COX_COD_ID] BETWEEN 13900000 AND 13999999 AND
             [COX_LNG_ID] = 'GB'
ORDERBY=     [COX_COD_ID]
LOADTABLE=   TRUE

-----
-- SERVICE CALL CLOSURE CODES --
-----
[CL_CODE_SC_CLO]
SOURCE=      [APP_CODE_TEXTS]
ATT=         [TEXT]
COLUMNS=    [APP_CODE_TEXTS].[COX_TEXT] AS [TEXT]
CONDITION=   [COX_COD_ID] BETWEEN 14000000 AND 14099999 AND
             [COX_LNG_ID] = 'GB'
ORDERBY=     [COX_COD_ID]
LOADTABLE=   TRUE
```

**NOTE:** It is possible to import one language. The migration will not function properly if you try to import more than one language, both English and Dutch, for example.

### 4.5.3 Migrating ITSM Variable Fields

A configuration file called ITSM\_Varfields.ini is available for migrating variable fields. Refer to section 4.8.3 “Migrating Variable Fields” on page 40 for an example of how you can modify the configuration file and modify the import mapping to import your particular variable fields.

The script is designed to identify all category descriptions that need to be adapted because they occur more than once in ITSM or because they already exist as codes in Service Desk

## 4.6 Creating an ODBC Connection to the ITSM Database

The ODBC Connection needs to be set on your Service Desk application server prior to exporting or importing data into Service Desk.

1. From the Windows **Start** button, select **Settings**, then **Control Panel**, then select **ODBC Data Sources**.
2. Open the **System DSN** tab and click **Add**.
3. Select the **Microsoft ODBC for Oracle** option and enter the following:  
Data source name: **migration**  
Description: migration **ITSM**  
User name: **ITSM database user**.  
Server: connect string used to connect to the ITSM database user.
4. Click **Save**.

The ODBC data source name entered needs to match the name used in the DSN section of each export configuration file. Check the data source name section of all of the configuration files you will use to make sure they contain the correct information. The following example shows the default settings in quotation marks:

```
[DSN]
NAME=name of the ODBC data source, "migration"
USR=login name for database user "itsm_user"
PWD=password "itsm_user"
```

## 4.7 Running pre\_check\_itsm.sql to prepare ITSM

This section contains information about tasks you will want to perform in ITSM application prior to migrating. The *pre\_check\_item.sql* script is provided to help identify areas in your ITSM environment they may need modification.

Pre\_check\_itsm.sql is provided with the migration tools and needs to be run on your ITSM server prior to migration. After modifying the category descriptions identified you can run the script again to verify that all of the corrections were made. To run the script:

1. Start **Oracle SQL\*Plus®** from your **Start** menu.
2. Log on to your ITSM database.
3. Start the script with the following command: *SQL>start c:\pre\_check\_itsm.sql*

**Figure 3- pre\_check\_itsm.sql script**

```

Oracle SQL*Plus
File Edit Search Options Help
SQL*Plus: Release 8.1.5.0.0 - Production on Fri Jul 20 09:41:04 2001
(c) Copyright 1999 Oracle Corporation. All rights reserved.

Connected to:
Oracle7 Server Release 7.3.4.3.0 - Production
With the distributed, replication and parallel query options
PL/SQL Release 2.3.4.3.0 - Production

SQL> start c:\pre_check_itsm.sql

```

#### 4.7.1 Configuration Item Administrators without a Workgroup

To migrate information about Configuration Item administrators to Service Desk the administrators must first be added to a workgroup in ITSM. If a person is registered as an administrator for a configuration item in ITSM and they are not registered to a workgroup, the person will not be imported as an administrator.

#### 4.7.2 Overview priority and impact texts.

Can be used to add value mappings for Priority and impact codes.

#### 4.7.3 Truncated Fields

The following fields are truncated because the ITSM field is longer than the corresponding Service Desk 4.0 field. During migration the truncated data will be lost:

**Table 8- Truncated with Data Loss During Migration**

ITSM	Char Length	Service Desk	Char Length
Contact.Attn	60	Person.Name	50

Employee.Attn	60	Person.Name	50
Service.Srv_description	2000	Service.Description	80

The following table shows truncated fields where the truncated portion of the data will be placed in the Service Desk history lines:

**Table 9 -Truncated with No Data Loss During Migration**

ITSM	Char Length	Service Desk	Char Length
Progress.Action	2000	History Line Servicecall.Subject	255
Problem_Progress.Action	2000	History Line Problem.Subject	255
WO_progress.Wop_Action	2000	History Line Workorder.Subject	255
Change_Progress.Action	2000	History Line Change.Subject	255

#### 4.7.4 ITSM Category descriptions

If you import a CI category description that matches the description of any of the codes already present in Service Desk, the error *more than one item found* will occur and the import will fail. For the same reason importing ITSM categories that have the same description will also fail. All CI categories imported into Service Desk can have a parent category. During the import process a search is conducted for the parent category based on the category description. In Service Desk 4.0 this search is conducted on all codes, including incident classification codes already present in Service Desk.

## 4.8 Modifying the Import Mapping

Import mapping is used to tell Service Desk where to store data that is imported. Default import mappings of ITSM entities and fields to Service Desk items and attributes are provided. You may need to change the import mapping so that all of the data you export is mapped in Service Desk. Review the import mapping carefully, some data from ITSM cannot be imported into Service Desk because of differences in the two applications, and in some cases information from one ITSM entity may be mapped to multiple items in Service Desk. The appendices in this guide include an overview of the import mapping, and a detail description of the import mapping. Chapters 2 and 3 of this manual provide an overview of the differences in ITSM compared to Service Desk.

For detailed information on how to conduct import mapping in Service Desk, see “*Configuring the Import Settings*” in the *HP OpenView Service Desk 4.0: Data Exchange Guide*.

**CAUTION:** Typographical errors in your import mapping will result in a loss of data during migration.

### 4.8.1 Checking the Value Mapping for Locale Codes

It is important to check the ITSM values that are mapped to locale codes in Service Desk. If the codes have been changed, the import mapping provided for those values by default will fail. For example, the ITSM value MAIL is mapped to Mail in Service Desk, if Mail has been changed to Post the mapping will not work. If differences exist in ITSM or Service Desk for any of the values in the following list, you will need to modify the value mapping:

**Table 10 - Value Mapping for Locale Codes**

Import Mapping	Ext. Class	Ext. Attribute	ITSM Value	Service Desk Value
ITSM_Organization	CL_ADDRESS_CONTACT_POSTAL	TYPE	MAIL	Mail
ITSM_Organization	CL_ADDRESS_CONTACT_VISITING	TYPE	BUSINESS	Business
ITSM_Organization	CL_ADDRESS_EMP	TYPE	HOME	Home
ITSM_Organization	CL_ADDRESS_EXTORG_POSTAL	TYPE	MAIL	Mail
ITSM_Organization	CL_ADDRESS_EXTORG_VISITING	TYPE	BUSINESS	Business
ITSM_Organization	CL_CONTACT	CATEGORY	CONTACT	Contact
ITSM_Organization	CL_CONTACT	GENDER	Female	Female
ITSM_Organization	CL_CONTACT	GENDER	Male	Male
ITSM_Organization	CL_CONTACT	STATUS	ACTIVE	Active
ITSM_Organization	CL_EMPLOYEE	CATEGORY	EMPLOYEE	Employee
ITSM_Organization	CL_EMPLOYEE	GENDER	Female	Female
ITSM_Organization	CL_EMPLOYEE	GENDER	Male	Male
ITSM_Organization	CL_EMPLOYEE	STATUS	ACTIVE	Active
ITSM_Organization	CL_EMPLOYEE_ACTIVE	STATUS	ACTIVE	Active
ITSM_Organization	CL_EMPLOYEE_INACTIVE	STATUS	INACTIVE	Inactive/retired
ITSM_Organization	CL_EXT_ORGANIZATION	CATEGORY	EXTERNAL	Company
ITSM_Organization	CL_EXT_ORGANIZATION	STATUS	ACTIVE	Active
ITSM_Organization	CL_INT_ORGANIZATION	CATEGORY	INTERNAL	Organization
ITSM_Organization	CL_INT_ORGANIZATION	STATUS	ACTIVE	Active
ITSM_Organization	CL_TEL_CONTACT1	TYPE	BUSINESS	Business
ITSM_Organization	CL_TEL_CONTACT2	TYPE	BUSINESS	Business
ITSM_Organization	CL_TEL_CONTACT3	TYPE	HOME	Home
ITSM_Organization	CL_TEL_CONTACT4	TYPE	FAX	Fax
ITSM_Organization	CL_TEL_EMP1	TYPE	BUSINESS	Business
ITSM_Organization	CL_TEL_EMP2	TYPE	BUSINESS	Business
ITSM_Organization	CL_TEL_EMP3	TYPE	HOME	Home
ITSM_Organization	CL_TEL_EMP4	TYPE	FAX	Fax
ITSM_Organization	CL_TEL_EXTORG1	TYPE	BUSINESS	Business
ITSM_Organization	CL_TEL_EXTORG2	TYPE	BUSINESS	Business
ITSM_Organization	CL_TEL_EXTORG3	TYPE	BUSINESS	Business
ITSM_Organization	CL_TEL_EXTORG4	TYPE	FAX	Fax
ITSM_Organization	CL_TEL_INTORG1	TYPE	BUSINESS	Business
ITSM_Organization	CL_TEL_INTORG2	TYPE	FAX	Fax

#### 4.8.2 Mapping Impact and Priority Code Values

The import mapping provided for this migration does not include value mapping for Impact and Priority fields associated with service calls, work orders, changes, and problems in ITSM. If you want to migrate Priority and Impact values you need to create your own import mapping for them. You will need to decide what Service Desk values the ITSM values should be mapped to.

When mapping priority and impact codes it is important to realize that Service Desk items use only one code range for Priority codes and one code range for Impact codes. In ITSM it is possible to have a code range for each module, for example a service call, problem, and change can all have a different set of priority and impact codes. Before importing it is important to check that the import and priority codes in ITSM are mapped to existing Service Desk Priority and Impact codes. If the codes are not mapped correctly you will need to map the values for those attributes. Refer to the *HP OpenView Service Desk 4.0: Data Exchange Administrator's Guide* for information on import mapping.

**NOTE:** An alternative to modifying the value mapping, is to change your values in the ITSM database.

### 4.8.3 Migrating Variable Fields

The default mapping provided for migration contains an example export configuration file *ITSM\_Varfields.ini* that can be modified for exporting and importing variable fields. If you have added or changed the variable fields in ITSM you will need to modify the example configurable extractor *ITSM\_Varfields.ini*, and create an import mapping in Service Desk. It may also be necessary to add fields in Service Desk before you import your ITSM variable fields. For example, if you migrate the field *extra information* from the service call class in ITSM to the field *extra info* in Service desk you will need to adjust the *ITSM\_Varfields.ini* file.

The following examples show the unchanged file with the areas to be changed marked in bold. The first example shows a portion of the unchanged *ITSM\_Varfields.ini* file. The second example shows the modifications made to export the variable *field extra information*.

This example is a portion of the *ITSM\_Varfields.ini* file, the file supplied with the migration includes every entity, (service call, change, problem, work order, organization, department, contact, employee, configuration item), and every type of variable field (char, code, date, lower, upper, money, number).

#### EXAMPLE 1

```
[DSN]
NAME=ODBC data source name, (default is "migration")
USR=login name of user who owns the data source tables and views
    (default is "itsm_user")
PWD=Database user's password (default is "itsm_user")

[SYSTEM]
LOG=TRUE
XML=TRUE
LOG_FILE=C:\ITSM_Varfields.log
OUTPUT_FILE=C:\ITSM_Varfields.txt
XML_OUTPUT_FILE=C:\ITSM_Varfields.xml
APPLICATION_NAME=ITSM

[CLASSES]
NAME=  SC_CHAR, ...
```

*The following example class is for variable fields of Service calls of type CHAR. During import use SOURCE\_ID to search for items by ID.*

```
[SC_CHAR]
SOURCE=  [VARIABLE_FIELD], [VARIABLE_VALUE], [APP_CODE_TEXTS]
ATT=     [SOURCE_ID], [FIELD_VALUE]
COLUMNS= [VARIABLE_VALUE].[ID1] AS [SOURCE_ID],
          [VARIABLE_VALUE].[COLUMN_VALUE] AS [FIELD_VALUE]
CONDITION= [VARIABLE_FIELD].[CATEGORY]=[VARIABLE_VALUE].[CATEGORY]
           AND [VARIABLE_FIELD].[ID]=[VARIABLE_VALUE].[COLUMN_NUMBER] AND
           [VARIABLE_FIELD].[CATEGORY]=TO_CHAR([APP_CODE_TEXTS].[COX_COD_ID]
           ) AND ([APP_CODE_TEXTS].[COX_LNG_ID]='GB' OR
           [APP_CODE_TEXTS].[COX_LNG_ID] IS NULL) AND
           [APP_CODE_TEXTS].[COX_SEARCHCODE]='SC' AND
           [VARIABLE_FIELD].[ID]=Field_ID
LOADTABLE= TRUE
```



1. Changing the class names in the ini file is optional. It may make it easier to identify the variable fields you are importing. You can give classes any name you want but the two instances must match. In the second example we have renamed the class **SC\_EXTRAINFO**. This is what the variable field will be called in the XML file after export. This is the external term you will need to map to a Service Desk field.
2. Replace **Field\_ID** in the CONDITION section with the ID of the variable field you want to import from ITSM. The field we are exporting in the second example has ID **10**. For the variable field of configuration items, you need to fill in the `Field_ID` as well as the search codes for the variable fields that belong to your three CI subcategories. If you do not want to migrate a subcategory, replace the search code by IS NULL. If you will not be using a subcategory-value, completely remove the following condition:  
`[ACT...].[COX_SEARCHCODE]='Subcategory..._searchcode'`

The following code example shows how the classes section of the ITSM\_Varfields.ini file was changed to export the variable field called extra information, the changed sections are in bold:

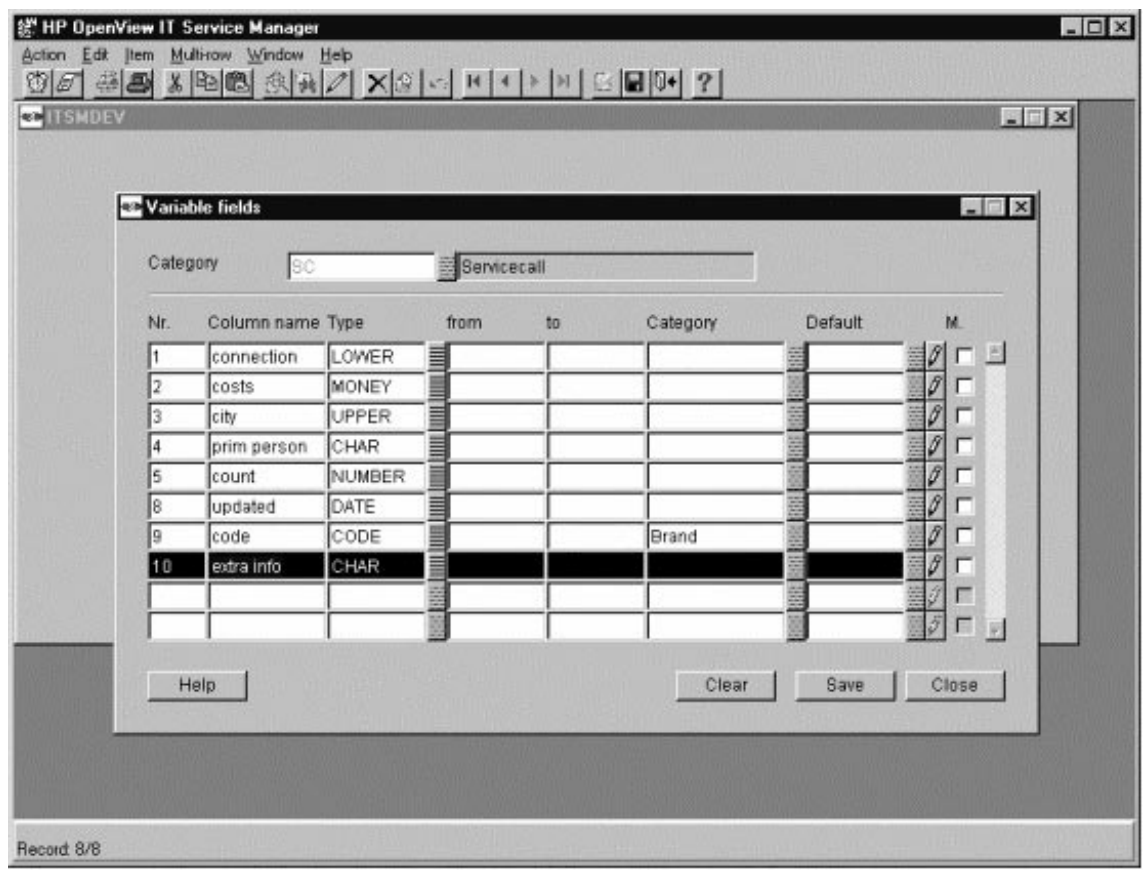
## EXAMPLE 2

```
[CLASSES]
NAME=   CL_VARFIELD_SC_EXTRAINFO

[CL_VARFIELD_SC_EXTRAINFO]
SOURCE=   [VARIABLE_FIELD], [VARIABLE_VALUE], [APP_CODE_TEXTS]
ATT=      [SOURCE_ID], [FIELD_VALUE]
COLUMNS= [VARIABLE_VALUE].[ID1] AS [SOURCE_ID],
          [VARIABLE_VALUE].[COLUMN_VALUE] AS [FIELD_VALUE]
CONDITION= [VARIABLE_FIELD].[CATEGORY]=[VARIABLE_VALUE].[CATEGORY]
           AND [VARIABLE_FIELD].[ID]=[VARIABLE_VALUE].[COLUMN_NUMBER] AND
           [VARIABLE_FIELD].[CATEGORY]=TO_CHAR([APP_CODE_TEXTS].[COX_COD_ID]
           ]) AND ([APP_CODE_TEXTS].[COX_LNG_ID]='GB' OR
           [APP_CODE_TEXTS].[COX_LNG_ID] IS NULL) AND
           [APP_CODE_TEXTS].[COX_SEARCHCODE]='SC' AND
           [VARIABLE_FIELD].[ID]=10
LOADTABLE= TRUE
```

To locate the ID number for your variable field in ITSM, from the System menu select General and then Variable fields. The ID number is visible in the left portion of the Variable fields dialog box, in the Nr: column:

Figure 4- Variable Field ID Number



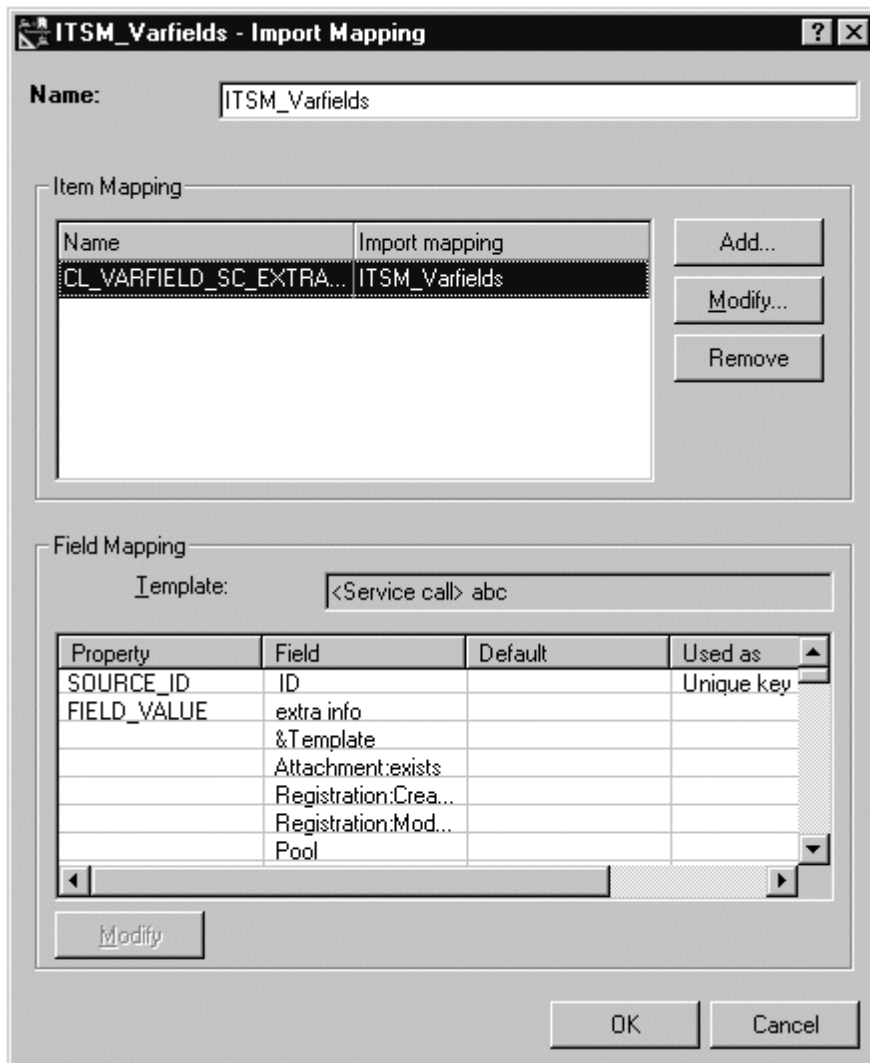
3. The variable field extra info does not exist in Service Desk. You can map the variable field to an existing field in Service Desk with a different name, or add a custom field in Service Desk. The type of field you create can be important, for example variable fields of the type Money in ITSM may contain a dot as separator, and need to be migrated to a Text type field in Service Desk. To create custom fields:
  - a. From the **Tools** menu; select **System**.
  - b. In the Administrator console navigate to the correct file by clicking **Data**, then **Custom Fields**, and then **Service Call**.
  - c. Adjust the dialog box as shown in the following example. Additional information on how to create custom fields and add fields to forms can be found in the online help:

**Figure 5 - Service Call Custom Fields**

The screenshot shows a dialog box titled "Service Call - Custom Fields". It contains the following elements:

- Field:** A dropdown menu showing "Sc. Text 1 (Text 255)".
- Name:** A text input field containing "extra info".
- Activate:** A checked checkbox.
- Min. value:** An empty text input field.
- Max. value:** An empty text input field.
- Categories:** Two radio buttons. The first, "All Categories", is selected. The second, "Enabled for the following categories", is unselected.
- Category List:** A list box containing four items, each with an unchecked checkbox:
  - RFC (Request for change)
  - RFD (Request for documentation)
  - RFI (Request for information)
  - RFT Request for training
- Buttons:** "OK" and "Cancel" buttons at the bottom.

4. Create an import mapping for `ITSM_Varfields`, for example:
  - a. Map `CL_VARFIELD_SC_EXTRAINFO` to Service Call, using template `ITSMDummy`.
  - b. Relate `SOURCE ID` to `ID` as a binding key (unique key). This will find the correct service call. For Service calls, Changes, Problems and Work orders, map `SOURCE_ID` to `ID`. For Contacts, Employees, Organizations, Departments and Configuration items map `SOURCE_ID` to `Source ID`.
  - c. Relate `extra info` to `FIELD_VALUE`, using additional value mapping if you want

**Figure 6- Custom Fields**

You can create a Data Exchange Task to export and import the variable fields. For information on creating a Data Exchange Task see “Creating Data Exchange Tasks” in the *HP OpenView Service Desk 4.0 Data Exchange Administrator’s Guide*.

## 4.9 Order of Migration

We recommend executing the tasks to import data one at a time, in the following order for optimum results. The following table contains the configuration file names, import mapping names and description of the data they are configured to migrate:

Export configuration file	Import mapping	Description
ITSM_Pools.ini	ITSM_Pools	Pools
ITSM_Accounts.ini	ITSM_Accounts	Accounts
ITSM_Organization.ini	ITSM_Organization	Contacts, External organizations (companies), Internal organizations (departments), Employees
ITSM_Cmdb.ini	ITSM_Cmdb	Configuration items
ITSM_Services.ini	ITSM_Services	Services
ITSM_ServiceLevel.ini	ITSM_ServiceLevel	Service Levels
ITSM_SLA.ini	ITSM_SLA	Service Level Agreements
ITSM_Freedays.ini	ITSM_Freedays	Freedays
ITSM_Servicecalls.ini	ITSM_Servicecalls	Service calls
ITSM_Problems.ini	ITSM_Problems	Problems
ITSM_Changes.ini	ITSM_Changes	Changes
ITSM_Workorders.ini	ITSM_Workorders	Work orders
ITSM_Checklist.ini	ITSM_Checklist	Problem solving checklist
ITSM_Varfields.ini	ITSM_Varfields	Variable fields example

**NOTE:** Due to differences between ITSM 5.6 and ITSM 5.7 two different tasks are available for the Organization and Workorder modules. One for ITSM 5.6 users and one for ITSM 5.7 users.

#### 4.10 Exporting ITSM Data

The migration tasks, can be run individually to migrate the parts of ITSM you want in Service Desk or as a group. We recommend you run each task individually, and check the log file for errors before importing running the import task.

To export data from ITSM, from the **Tools** menu select **System**, open the **Data** folder and double-click the task you want to execute, for example if you want to export service calls then double-click the task "*ITSM\_Servicecalls*":

1. In the Data Exchange dialog box, select **Export data from a storage device**.
2. Enter the ITSM\_\*.ini file configured for the data exchange task you want to execute.
3. Enter the name of the exported file. This will be the name of the XML file created by the export process. For example, ITSM\_Servicecalls.xml.
4. Clear all other fields and click **OK** at the bottom of the screen to export.

You can also export from the command line with: `sd_exchange export <config file> <log file> <xml file>`

**NOTE:** When migrating data we recommended that you execute the data exchange tasks in the sequence specified.

#### 4.10.1 Checking for Errors After Exporting

There are two ways to check for errors. One is to check the log file, this can be viewed at any time either during or after the export. Syntax errors that prevent data from being exported properly will show up in the log file. You can also use the Viewer, to view the data exported from ITSM before you import it into Service Desk. The Viewer in Data Exchange converts the XML file to HTML format and presents it in an object-tree format. To view exported data files:

1. In the Data Exchange dialog box, enter the XML file you want to view, and click **View**.
2. Verify that all items you want to import were exported correctly.
3. Compare the file with the import mapping. Only correctly mapped entities, attributes and values will be mapped.

**NOTE:** Explorer is used to view XML files. Explorer will try to load the whole file in memory, causing a “System running low on virtual memory” warning if the XML files are very big. If your XML files are large open them for viewing in a text editor instead.

#### 4.11 Importing ITSM Data

After you have performed a task to export your ITSM data into an XML file and have checked it, you are ready to import it into Service Desk. You could select to export and import the task at one time, but that will not allow you the opportunity to check that your data was exported correctly before you import it. You can run the import data task from the Data Exchange dialog box by selecting the Tools menu, then System, open the Data directory and double-click on the task you want to execute. A task is available for all major areas of the migration process. For example, tasks exist for you to export and import service calls, and another task exists for Service Level Agreements. When you double-click a task it will be opened in the Data Exchange dialog box with the default task information already filled in the fields. The following fields should be filled for importing data:

1. Name field. This is the name of the exported file. This will be the name of the XML file created by the export process.
2. Select the **Import data into the Service Desk database** check box.
3. Enter the Account established for importing data from ITSM. For importing the data, you must fill in the migration-user password “**migration**”.
4. In the Import Mapping field, enter the name of the import mapping created for this task.
5. Select the Debug check box to create a detailed log file while importing.
6. Clear all other fields and click OK at the bottom of the screen to import.
7. You can also import from the command line with: `sd_exchange import <input file> <username/password> <mapping> debug <import log> <tempdir>`

For additional information on importing data, see “*Importing Data in Batches*” in the *HP OpenView Service Desk 4.0 Data Exchange Administrator’s Guide*.

**NOTE:** Running the pre-migration.script creates a special migration account for you to use. This migration account provides additional access to the Service Desk database that is needed for migrating. Do not use this account for purposes other than migration.

#### 4.11.1 Verifying the Data Migration

1. You can evaluate the status of the migration by looking at the log file and the error log file during and after exchanging data. The log files are located in `data_exchange\log` after they are created. To create a detailed log, select the Debug mode in the Data Exchange dialog box for each task. When the export process is completed, the sentence *Extractor finished* will appear in the log file. When the import process is finished, the sentence *Finished loading relations at...* will be shown in the log file.

#### 4.12 Running the Post- Migration Action

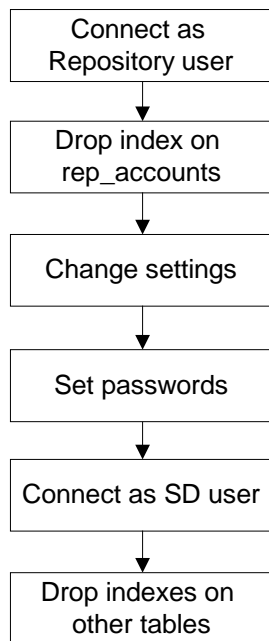
The Post\_migration action must be run on the Service Desk database after the migration has been completed. It drops the indexes that were created in the pre-migration script and resets all necessary settings:

**NOTE:** All errors will be logged in the file `dbconfconsole.log`. This file will be created at the location where you are running the script.

The action performs the following tasks:

- Drops the indexes that were created in the pre-migration action on the following tables: `rep_accounts` (on `acc_loginname`), `itsm_persons` (on `per_sourceid`), `itsm_configuration_items` (on `cit_sourceid`), `itsm_services` (on `srv_sourceid`), `itsm_organizations` (on `org_sourceid`) and `itsm_workgroups` (on `wog_sourceid`)
- Changes the following settings:  
puts the ID After Creation setting to “on”
- Gives all migrated accounts the password “itsm”
- Switches the account field of an Account to “mandatory”.

The diagram below shows the flow of the post-migration action.

**Figure 7- Post Migration action**

### 4.13 Parallel import option

Import sessions can be time consuming. An option to speed up the import process is the use of parallel import sessions. A general approach of setting up parallel import sessions can be found in the “*HP OpenView Service Desk 4.0 Data Exchange Administrator’s Guide*”. This section will explain the steps needed to use the parallel mechanism for the migration.

#### 4.13.1 About the Parallel Import Process

The parallel import process cannot be used from the user interface. Therefore command line and/or scripts are needed. This means the parallel import is more complex compared with a single import session.

The most import reason to use the parallel import option is because of time saving. One should consider the following disadvantages:

- More hardware required
- More Service Desk installations
- Higher complexity of the process
- Command line / script approach

If you decide to use the option the following import tasks will be of interest because they are often large files containing many of the same class items:

ITSM\_Organizations, ITSM\_Cmdb, ITSM\_Servicecalls.

#### 4.13.2 Using The parallel option

Parallel processing requires synchronization between dependent classes. Example: Before one can import the members of a workgroup, all persons have to be imported. Within the



context of a single import process, this dependency is defined in the sequence of extracted classes. The parallel import process is XML-file orientated, each object in the XML-file should be independent of all other objects within the context of the file. In the prescribed example, there should be two XML-files, one with the persons and one with the workgroups.

In the migration, one doesn't need to determine all these dependencies manually: the export process has an option that overrules the name of the XML file to output an XML file per class.

The option (in the [SYSTEM] section) is:

```
CLASS_TO_XML=TRUE
```

This option results in one XML file per class.

For example, the output of the extraction of: ITSM\_Servicecalls (in the directory "C:\Program Files\Hewlett-packard\OpenView\ServiceDesk\3.0\Client\data\_exchange\xml\") would be:

```
CL_CODE_SC_CAT.xml
CL_CODE_SC_CLO.xml
CL_CODE_SC_INC.xml
CL_CODE_SC_MED.xml
CL_CODE_SC_STA.xml
CL_SERVICECALL.xml
CL_SC_HISTORY.xml
CL_SC_HISTORY_INFO.xml
CL_SC_CAUSED_BY_RELTYPE.xml
CL_SC_RELATED_RELTYPE.xml
```

Each of these XML files can be processed with the parallel option.

### 4.13.3 Configuring Script Files

Two script files are needed. One that starts up each "master-session" and one to startup a "client-session".

#### The Master Sessions:

The master session script file should contain a separate entry for each class (that is not a child class) to import. The sequence is determined in the NAME tag of the section [CLASSES] of the export configuration file.

A sample script file for ITSM\_Servicecalls:

```
REM
REM @(#) $Workfile: ImportMasterSample.bat $
REM

REM Import Master Sample Script file, based on ITSM_Servicecalls.

REM Inside ITSM_Servicecalls.ini one finds the following entry:
REM
REM     [CLASSES]
REM     NAME= CL_CODE_SC_CAT, \
REM           CL_CODE_SC_CLO, \
REM           CL_CODE_SC_INC, \
REM           CL_CODE_SC_MED, \
```

```
REM          CL_CODE_SC_STA, \  
REM          CL_SERVICECALL, \  
REM          CL_SC_HISTORY, \  
REM          CL_SC_HISTORY_INFO, \  
REM          CL_SC_CAUSED_BY_RELTYPE, \  
REM          CL_SC_RELATED_RELTYPE  
REM  
REM None of these classes is a child class (ie: no parent child  
relations are  
REM defined). So each class needs a separate entry of sd_import.  
REM For an extensive description of the sd_import parameters see the  
REM Data Exchange Guide.  
  
REM Define the bin directory of SD  
SET BINPATH=Adjust to the installation\bin path of SD  
  
REM Define the application server to be used  
SET SERVER=Adjust to application server  
  
REM Define the full path to the shared directory  
SET DATA_EXCHANGE=Adjust to your shared directory  
  
REM Define the right import mapping  
SET MAPPING=ITSM_Servicecalls  
  
REM Define Login  
SET USER=migration  
SET PASSWD=migration  
  
REM Pack the command including the shared arguments  
SET COMMAND=%BINPATH%\sd_import %USER% %PASSWD% %SERVER% "-  
parallel=%DATA_EXCHANGE%" -mapping=%MAPPING%  
  
REM shorten the log dir argument  
SET LOG_DIR=-logfile=%DATA_EXCHANGE%\log  
  
%COMMAND% "-data=CL_CODE_SC_CAT.xml" "%LOG_DIR%\CL_CODE_SC_CAT.log"  
%COMMAND% "-data=CL_CODE_SC_CLO.xml" "%LOG_DIR%\CL_CODE_SC_CLO.log"  
%COMMAND% "-data=CL_CODE_SC_INC.xml" "%LOG_DIR%\CL_CODE_SC_INC.log"  
%COMMAND% "-data=CL_CODE_SC_MED.xml" "%LOG_DIR%\CL_CODE_SC_MED.log"  
%COMMAND% "-data=CL_CODE_SC_STA.xml" "%LOG_DIR%\CL_CODE_SC_STA.log"  
%COMMAND% "-data=CL_SERVICECALL.xml" "%LOG_DIR%\CL_SERVICECALL.log"  
%COMMAND% "-data=CL_SC_HISTORY.xml" "%LOG_DIR%\CL_SC_HISTORY.log"  
%COMMAND% "-data=CL_SC_HISTORY_INFO.xml "  
"%LOG_DIR%\CL_SC_HISTORY_INFO.log"  
%COMMAND% "-data=CL_SC_CAUSED_BY_RELTYPE.xml "  
"%LOG_DIR%\CL_SC_CAUSED_BY_RELTYPE.log"
```

Note: each sd\_import command consists of one line, including all of its parameters.

### The Client Session

The script file for the client session is much simpler. The master supplies clients with work. A client only needs to know the shared directory.

```

REM
REM @(#) $Workfile: ImportClientSample.bat $

REM Import Client Sample Script file

REM Define the bin directory of SD
SET BINPATH=Adjust to the installation\bin path of SD

REM Define the application server to be used
SET SERVER=Adjust to application server

REM Define the full path to the shared directory
SET DATA_EXCHANGE="Adjust to your shared directory"

REM Define Login
SET USER=migration
SET PASSWD=migration

REM Pack the command including the shared arguments
SET COMMAND=%BINPATH%\sd_import %USER% %PASSWD% %SERVER% "-
parallel=%DATA_EXCHANGE%"

REM Execute the client import session
REM One import session serves multiple master sessions

%COMMAND%
```

The sample script files can be found in: <ProductPath>\Migrations\ltsm\ParallelImport

### 4.14 Tips and Troubleshooting

This section contains additional problem-solving information, along with a few tips that may or may not be useful for you during the migration process. Refer to the readme file for additional information. If you experience poor performance when exporting, it is possible that an equipment fault is to blame. To isolate the problem check the following items in this order:

1. Database
2. Network
3. Application server

If you experience performance problems when importing, check the following in this order:

1. Application server
2. Network
3. Database

#### **4.14.1 Memory Problems**

If you encounter memory problems when exporting from ITSM to the XML file, the export configuration files can be adapted as follows: Replace “LOADTABLE=TRUE” with “LOADTABLE=FALSE” for every table that causes memory problems. The export process will take longer, but it will use less memory. The attribute: LOADTABLE, is used to specify whether the records are cached in memory to process parent-child relations faster, or queries are run for each parent to find its children.

## Appendix A - Mapping Overview with Field Lengths

Here you find a complete overview of all fields migrated from ITSM to Service Desk. Field lengths are noted, to identify the cases in which truncating should be done.

ITSM Table	ITSM Field	Length	XML Class	XML Attribute	SD Entity	SD Attribute	Length
<b>Pools</b>							
DATA_SET	DESCRIPTION	50	CL_POOL	TEXT	Folder	Text	255
<b>Organization</b>							
APP_CODE_TEXTS SPECIALIST_GROUP	APP_CODE_TEXTS .COX_TEXT	30	CL_CODE_WORKGROUP	TEXT	Workgroup	Name	50
	APP_CODE_TEXTS .COX_SEARCHCODE	10		SEARCHCODE		Searchcode	50
	APP_CODE_TEXTS .COX_COD_ID	10		SOURCE_ID		Source ID	80
SPECIALIST_GROUP	'EMP' + SGR_EMPLOYEE		CL_WORKGROUP_EMP	EMP_ID	Person	use to search Person	
				Parent		use to search Members:Workgroup	
TIMEZONE	TIMEZONE.TIM_SEARC HCODE	10	CL_TIMEZONES	SEARCHCODE	Timezone	Search code	
	TIMEZONE.TIM_DESCRI PTION	30		DESCRIPTION		Name	
	TIMEZONE.TIM_OFFSET	3 + 1		OFFSET		Offset	
APP_CODE_TEXTS ADDRESS	COX_TEXT	30	CL_CODE_ORG_LOC	TEXT	Location	Searchcode	50
	VISITING_ADDRESS1	30	CL_ADDRESS_EXTORG_VIS ITING	STREET1	Address	Street1	50
	VISITING_ADDRESS2	30		STREET2		Street2	50
	VISITNG_ZIPCODE	30		ZIP		Zip/Postal code	50
	VISITING_CITY	30		CITY		City	50
	VISTING_REGION	30		STATE		State/Province	50
	COUNTRY	30		COUNTRY		Country/Region	50
	'EO' + ID	2 + 10		ORG_ID		use to search Organization	
	'BUSINESS'			TYPE		Address type	
ADDRESS	POSTAL_ADDRESS1	30	CL_ADDRESS_EXTORG_PO STAL	STREET1	Address	Street1	50
	POSTAL_ADDRESS2	30		STREET2		Street2	50
	POSTAL_ZIPCODE	30		ZIP		Zip/Postal code	50
	POSTAL_CITY	30		CITY		City	50
	POSTAL_REGION	30		STATE		State/Province	50
	COUNTRY	30		COUNTRY		Country/Region	50
	'EO' + ID	2 + 10		ORG_ID		use to search Organization	
	'MAIL'			TYPE		Address type	
ADDRESS	ADDRES.TELEPHONE1	30	CL_TEL_EXTORG1	TELNO	Telephone	Number	30
	'EO' + ADDRES.ID	2 + 10		ORG_ID		use to search Organization	
	'BUSINESS'			TYPE		Type	
	'1'			PRIMARY		Primary	
ADDRESS	ADDRES.TELEPHONE2	30	CL_TEL_EXTORG2	TELNO	Telephone	Number	30
	'EO' + ADDRES.ID	2 + 10		ORG_ID		use to search Organization	
	'BUSINESS'			TYPE		Type	
ADDRESS	ADDRES.TELEPHONE3	30	CL_TEL_EXTORG3	TELNO	Telephone	Number	30
	'EO' + ADDRES.ID	2 + 10		ORG_ID		use to search Organization	
	'BUSINESS'			TYPE		Type	
ADDRESS	ADDRES.FAX	30	CL_TEL_EXTORG4	FAXNO	Telephone	Number	30
	'EO' + ADDRES.ID	2 + 10		ORG_ID		use to search Organization	
	'FAX'			TYPE		Type	
ADDRESS	EMAIL	80	CL_EXT_ORGANIZATION	EMAIL	Organization	E-mail	80
	NAME1	30		NAME1		Name1	50
	NAME2	30		NAME2		Name2	50
	REMARK	255		REMARK		Remark	255
	SEARCHCODE	10		SEARCHCODE		Search code	50
	'EO' + ID	2 + 10		ORG_ID		Source ID	80
	'EXTERNAL'			CATEGORY		Category	
	'ACTIVE'			STATUS		Status	
	TIMEZONE.TIM_SEARC HCODE	10		TIMEZONE		Timezone	
ORGANIZATION	'EO' + ADDRESS	2 + 10	CL_EXT_ORGANIZATION_R ELATION	ORG_ID	Organization	use to search Organization	
	'EO' + PARENT	2 + 10		PARENT_ID		use to search Parent	
ORGANIZATION_UNI T	TELEPHONE	30	CL_TEL_INTORG1	TELNO	Telephone	Number	30
	'IO' + ID	2 + 10		ORG_ID		use to search Organization	
	'1'			PRIMARY		Primary	
	'BUSINESS'			TYPE		Type	

Appendix A – Mapping Overview with Field Lengths

ORGANIZATION_UNI T	FAX	30	CL_TEL_INTORG2	FAXNO	Telephone	Number	30
	'IO' + ID	2 + 10		ORG_ID		use to search Organization	
	'FAX'			TYPE		Type	
ORGANIZATION_UNI T	NAME	50	CL_INT_ORGANIZATION	NAME	Organization	Name1	50
	REMARK	255		REMARK		Remark	255
	SEARCHCODE	10		SEARCHCODE		Search code	50
	'IO' + ID	2 + 10		ORG_ID		Source ID	80
	TIMEZONE.TIM_SEAR HCODE	10		TIMEZONE		Timezone	
	'INTERNAL'			CATEGORY		Category	
	'ACTIVE'			STATUS		Status	
ORGANIZATION_UNI T	'IO' + ID	2 + 10	CL_INT_ORGANIZATION_RE LATION	ORG_ID	Organization	use to search Organization	
	'IO' + PARENT	2 + 10		PARENT_ID		use to search Parent	
CONTACT ADDRESS APP_CODE TEXTS	ADDRESS.EMAIL	80	CL_CONTACT	EMAIL	Person	E-mail	80
	CONTACT.FIRST_NAME	20		FULLNAME_FIRST NAME		Full name : First name	50
	ADDRESS.NAME1	30		FULLNAME_LASTN AME		Full name : Last name	50
	CONTACT.TITLE1	20		FULLNAME_TITLE		Full name : Title	50
	CONTACT.BIRTHDAY			BIRTHDAY		Date of Birth	
	ACT.COX_TEXT	30		GENDER		Gender	
	CONTACT.INITIALS	20		INITIALS		Initials	50
	ACT.COX_TEXT	30		JOBTITLE		Job title	50
	CONTACT.ATTN	60		NAME		Name	50
	ADDRESS.NAME1	30					
	'EO' + CONTACT.ORGANIZATI ON	2 + 10		ORG_ID		use to search Organization	
	ADDRESS.REMARK	255		REMARK		Remark	255
	ADDRESS.SEARCHCOD E	10		SEARCHCODE		Search code	50
	'CON' + CONTACT.ADDRESS	3 + 10		CONTACT_ID		Source ID	80
	'CONTACT'			CATEGORY		Category	
	'ACTIVE'			STATUS		Status	
	TIMEZONE.TIM_SEAR HCODE	10		TIMEZONE		Timezone	
ADDRESS CONTACT	ADDRESS.VISITING_AD DRESS1	30	CL_ADDRESS_CONTACT_VI SITING	STREET1	Address	Street1	50
	ADDRESS.VISITING_AD DRESS2	30		STREET2		Street2	50
	ADDRESS.VISITNG_ZIP CODE	30		ZIP		Zip/Postal code	50
	ADDRESS.VISITING_CIT Y	30		CITY		City	50
	ADDRESS.VISTING_RE GION	30		STATE		State/Province	50
	ADDRESS.COUNTRY	30		COUNTRY		Country/Region	50
	'CON' + ADDRESS.ID	3 + 10		CONTACT_ID		use to search Person	
	'BUSINESS'			TYPE		Address type	
ADDRESS CONTACT	ADDRESS.POSTAL_AD DRESS1	30	CL_ADDRESS_CONTACT_P OSTAL	STREET1		Street1	50
	ADDRESS.POSTAL_AD DRESS2	30		STREET2		Street2	50
	ADDRESS.POSTAL_ZIP CODE	30		ZIP		Zip/Postal code	50
	ADDRESS.POSTAL_CIT Y	30		CITY		City	50
	ADDRESS.POSTAL_RE GION	30		STATE		State/Province	50
	ADDRESS.COUNTRY	30		COUNTRY		Country/Region	50
	'CON' + ADDRESS.ID	3 + 10		CONTACT_ID		use to search Person	
	'MAIL'			TYPE		Address type	
ADDRESS CONTACT	ADDRES.TELEPHONE1	30	CL_TEL_CONTACT1	TELNO	Telephone	Number	30
	'CON' + ADDRESS.ID	3 + 10		CONTACT_ID		use to search Person	
	'BUSINESS'			TYPE		Type	
ADDRESS CONTACT	ADDRES.TELEPHONE2	30	CL_TEL_CONTACT2	TELNO	Telephone	Number	30
	'CON' + ADDRESS.ID	3 + 10		CONTACT_ID		use to search Person	
	'BUSINESS'			TYPE		Type	
ADDRESS CONTACT	ADDRES.TELEPHONE3	30	CL_TEL_CONTACT3	TELNO	Telephone	Number	30
	'CON' + ADDRESS.ID	3 + 10		CONTACT_ID		use to search Person	
	'HOME'			TYPE		Type	
ADDRESS CONTACT	ADDRES.FAX	30	CL_TEL_CONTACT4	FAXNO	Telephone	Number	30
	'CON' + ADDRESS.ID	3 + 10		CONTACT_ID		use to search Person	
	'FAX'			TYPE		Type	
EMPLOYEE	PRIVATE_ADDRESS	30	CL_ADDRESS_EMP	ADDRESS	Address	Street1	50
	PRIVATE_ZIPCODE	30		ZIP		Zip/Postal code	50
	PRIVATE_CITY	30		CITY		City	50
	PRIVATE_REGION	30		REGION		Country/Region	50
	'EMP' + ID	3 + 10		EMP_ID		use to search Person	
	'HOME'			TYPE		Address type	

Appendix A – Mapping Overview with Field Lengths

EMPLOYEE	TELEPHONE1	30	CL_TEL_EMP1	TELNO	Telephone	Number	30
	'EMP' + ID	3 + 10		EMP_ID		use to search Person	
	'BUSINESS'			TYPE		Type	
	'1'			PRIMARY		Primary	
EMPLOYEE	TELEPHONE2	30	CL_TEL_EMP2	TELNO	Telephone	Number	30
	'EMP' + ID	3 + 10		EMP_ID		use to search Person	
	'BUSINESS'			TYPE		Type	
EMPLOYEE	PRIVATE_TELEPHONE	30	CL_TEL_EMP3	TELNO	Telephone	Number	30
	'EMP' + ID	3 + 10		EMP_ID		use to search Person	
	'HOME'			TYPE		Type	
EMPLOYEE	FAX	30	CL_TEL_EMP4	FAXNO	Telephone	Number	30
	'EMP' + ID	3 + 10		EMP_ID		use to search Person	
	'FAX'			TYPE		Type	
EMPLOYEE APP_CODE_TEXTS	'EMP' + EMPLOYEE.ID	3 + 10	CL_EMPLOYEE	EMP_ID	Person	Source ID	80
	EMPLOYEE.SEARCHCODE	10		SEARCHCODE		Search code	50
	EMPLOYEE.FIRST_NAME	30		FULLNAME_FIRSTNAME		Full name : First name	50
	EMPLOYEE.NAME	30		FULLNAME_LASTNAME		Full name : Last name	50
	EMPLOYEE.TITLE1	30		FULLNAME_TITLE		Full name : Title	50
	'10' + EMPLOYEE.ORGANIZATION_UNIT	2 + 10		ORG_ID		use to search Organization	
	ACT.COX_TEXT	30		GENDER		Gender	
	EMPLOYEE.REMARK	70		REMARK		Remark	255
	ACT.COX_TEXT	30		LOCATION_SEARCHTEXT		use to search Location	
	EMPLOYEE.INITIALS	30		INITIALS		Initials	50
	EMPLOYEE.ATTN	60		NAME		Name	50
	EMPLOYEE.NAME	30					
	EMPLOYEE.BIRTHDAY			BIRTHDAY		Date of Birth	
	EMPLOYEE.EMAIL	80		EMAIL		E-mail	80
	'EMPLOYEE'			CATEGORY		Category	
	'ACTIVE'			STATUS		Status	
	APP_LOGIN_USER.LOGIN_NAME	30		ACCOUNT_LOGINNAME		Account	
	TIMEZONE.TIM_ID	10		TIMEZONE		Timezone	
	ACT.COX_TEXT	30		JOBTITLE		Job title	50
EMPLOYEE	'EMP' + ID	3 + 10	CL_EMPLOYEE_ACTIVE	EMP_ID	Person	use to search Person	80
	'ACTIVE'			STATUS		Status	
EMPLOYEE	'EMP' + ID	3 + 10	CL_EMPLOYEE_INACTIVE	EMP_ID	Person	use to search Person	80
	'INACTIVE'			STATUS		Status	
ORGANIZATION_UNIT	'10' + ID	2 + 10	CL_INTORG_MANAGER_RELATION	OU_ID	Organization	use to search Organization	
	'EMP' + HEAD	3 + 10		MANAGER_ID		use to search Manager	
<b>Accounts</b>							
APP_LOGIN_USER EMPLOYEE SPECIALIST	APP_LOGIN_USER.LOGIN_NAME	30	CL_ACCOUNT_LICENSED	LOGIN_NAME	Account	Login name	40
	APP_LOGIN_USER.LOGIN_NAME	30		DISPLAY_NAME		Display name	50
	'FALSE'			SSP_ACCOUNT		SSP/Integrations account	
	'FALSE'			BLOCKED		Blocked	
	'TRUE'			CONCURRENT_USER		Concurrent User	
	'01-01-2001 01:01:01'			PW_MOD_DATE		Password modification date	
APP_LOGIN_USER EMPLOYEE SPECIALIST	APP_LOGIN_USER.LOGIN_NAME	30	CL_ACCOUNT_UNLICENSED	LOGIN_NAME	Account	Login name	40
	APP_LOGIN_USER.LOGIN_NAME	30		DISPLAY_NAME		Display name	50
	'TRUE'			SSP_ACCOUNT		SSP/Integrations account	
	'01-01-2001 01:01:01'			PW_MOD_DATE		Password modification date	
	'FALSE'			BLOCKED		Blocked	
	'TRUE'						
<b>Cmdb</b>							
APP_CODE_TEXTS	COX_TEXT	30	CL_CODE_CI_MAINCAT	TEXT	CI Category	Text	255
CI_CATEGORY APP_CODE_TEXTS	CI_CATEGORY.DESCRPTION	30	CL_CODE_CI_CAT	TEXT	CI Category	Text	255
	APP_CODE_TEXTS.COX_TEXT	30		PARENT_SEARCHTEXT		use to search Parent	
CI_SUBCATEGORY CI_CATEGORY	CI_SUBCATEGORY.DESCRPTION	30	CL_CODE_CI_SUBCAT	TEXT	CI Category	Text	255
	CI_CATEGORY.DESCRPTION	30		PARENT_SEARCHTEXT		use to search Parent	
APP_CODE_TEXTS APP_CODES	APP_CODE_TEXTS.COX_TEXT	30	CL_CODE_CI_STA	TEXT	CI Status	Text	255
	APP.CODES.COD_ORDERING	10		ORDERING		Ordering	int
APP_CODE_TEXTS	COX_TEXT	30	CL_CODE_CI_BRAND	TEXT	Brand	Text	255
CONFIGURATION	LOCATION1	30	CL_CODE_CI_LOC1	TEXT	Location	Searchcode	50
CONFIGURATION	LOCATION2	30	CL_CODE_CI_LOC2	TEXT	Location	Searchcode	50
APP_CODE_TEXTS	APP_CODE_TEXTS.COX_TEXT	30	CL_CODE_CI_RELTYPE	TEXT	CI relation type	Text	255

Appendix A – Mapping Overview with Field Lengths

APP_CODES	X_TEXT						
	APP_CODES.COD_ORDERING	10		ORDERING		Ordering	int
CONFIGURATION APP_CODE_TEXTS CI_CATEGORY CI_SUBCATEGORY DATA_SET	CONFIGURATION.ID	10	CL_CI	CI_ID	Configuration Item	Source ID	80
	'EMP' + CONFIGURATION.ADMINISTRATOR 'CON' + CONFIGURATION.ADMINISTRATOR	3 + 10		ADMIN_PERSON_ID		use to search Admin. Person	
	'IO' + CONFIGURATION.ADMINISTRATOR 'EO' + CONFIGURATION.ADMINISTRATOR	2 + 10		ADMIN_ORG_ID		use to search Admin. Org.	
	ACT.COX_TEXT	30		BRAND_SEARCHTEXT		use to search Brand	
	CI_SUBCATEGORY.CODE CI_CATEGORY.CODE ACT.COX_TEXT	10 10 30		CATEGORY_SEARCHTEXT		use to search Category	
	CONFIGURATION.LOCATION1 CONFIGURATION.LOCATION2	30 30		LOCATION_SEARCHTEXT		use to search Location	
	CONFIGURATION.NAME1	50		NAME1		Name 1	255
	CONFIGURATION.NAME2	50		NAME2		Name 2	255
	CONFIGURATION.ORDER_NUMBER	10		ORDER_NO		Order number	50
	'IO' + CONFIGURATION.OWNER 'EO' + CONFIGURATION.OWNER	10		OWNER_ORG_ID		use to search Owner Org.	
	CONFIGURATION.PRICE	10		PRICE		Price	10
	CONFIGURATION.PURCHASE_DATE			PURCHASE_DATE		Purchase date	
	CONFIGURATION.REMARK	255		REMARK		Remark	255
	CONFIGURATION.CODE	10		SEARCHCODE		Search code	80
	CONFIGURATION.SERIAL_NUMBER	50		SERIAL_NO		Serial Number	50
	ACT.COX_TEXT	30		STATUS_SEARCHTEXT		use to search Status	
	'EO' + CONFIGURATION.SUPPLIER	2 + 10		SUPPLIER_ID		use to search Supplier	
	CONFIGURATION.WARRANTY_DATE			WARRANTY_DATE		Warranty date	
	DATA_SET.DESCRPTION	50		POOL_SEARCHTEXT		use to search Pool	
	CONFIGURATION.MAX_INST	10		MAX_INST		Max. Installations	int
	TRUE FALSE	10		UNIQUE		Unique	
CI_RELATION APP_CODE_TEXTS	CI_RELATION.CONFIGURATION	10	CL_CI_RELATED	CI_PARENT	CI relation	use to search CI from	
	APP_CODE_TEXTS.COX_TEXT	30		RELTYPE_SEARCHTEXT		use to search CI relation type	
	CI_RELATION.CI	10		CI_CHILD		use to search CI to	
CI_INCLUDE	CONFIGURATION	10	CL_CI_COMPONENT_PARENT	CI_ID	Configuration Item	Source ID	80
CI_INCLUDE	CI	10	CL_CI_COMPONENT_CHILD	CI_ID	Configuration Item	Source ID	80
				Parent		use to search Parent CIs:CI Parent	
CONFIGURATION	'CON' + CI_USER 'EMP' + CI_USER	3 + 10	CL_CI_USER_RELATION_PARENT	PERSON_ID	Person	Source ID	80
CONFIGURATION	ID	10	CL_CI_USER_RELATION_CHILD	CI_ID	Configuration Item	Source ID	80
				Parent		use to search Users:User	
<b>Services</b>							
APP_CODE_TEXTS APP_CODES	APP_CODE_TEXTS.COX_TEXT	30	CL_CODE_SER_STA	TEXT	Service Status	Text	255
	APP_CODES.COD_ORDERING	10		ORDERING		Ordering	int
SERVICE DATA_SET APP_CODE_TEXTS	SERVICE.SRV_NAME1 SERVICE.SRV_NAME2	50 50	CL_SERVICE	NAME	Service	Name	80
	APP_CODE_TEXT.COX_TXT	30		STATUS_SEARCHTEXT		use to search Status	
	SERVICE.SRV_ID	10		SRV_ID		Source ID	80
	DATA_SET.DESCRPTION	50		POOL_SEARCHTEXT		use to search Pool	



Appendix A – Mapping Overview with Field Lengths

	N			XT		Description	
	SERVICE.SRV_DESCRIPTION	2000		DESCRIPTION		Description	80
SUPPORTING_CONFIGURATION	SUP_CNF_ID	10	CL_SERVICE_ASS_CI	CL_ID	Configuration Item	Source ID	80
				Parent		use to search Services:Service	
SERVICE	SRV_CNF_ID	10	CL_SERVICE_CI	CL_ID	Configuration Item	Source ID	80
				Parent		use to search Services:Service	
<b>Service calls</b>							
APP_CODE_TEXTS	COX_TEXT	30	CL_CODE_SC_CAT	TEXT	Service call Category	Text	255
APP_CODE_TEXTS APP_CODES	APP_CODE_TEXTS.COX_TEXT	30	CL_CODE_SC_CLO	TEXT	Service call Closure code	Text	255
	APP_CODES.COD_ORDERING	10		ORDERING		Ordering	int
APP_CODE_TEXTS	COX_TEXT	30	CL_CODE_SC_INC	TEXT	Service call Classification	Text	255
APP_CODE_TEXTS	COX_TEXT	30	CL_CODE_SC_MED	TEXT	Medium	Text	255
APP_CODE_TEXTS APP_CODES	APP_CODE_TEXTS.COX_TEXT	30	CL_CODE_SC_STA	TEXT	Service call Status	Text	255
	APP_CODES.COD_ORDERING	10		ORDERING		Ordering	int
SERVICECALL DATA_SET APP_CODE_TEXTS APP_LOGIN_USER	DATA_SET.DESCRIPTION	50	CL_SERVICECALL	POOL_SEARCHTEXT	Service call	use to search Pool	
	SERVICECALL.ID	10		SC_ID		ID	10
	SERVICECALL.SER_EVENT_ID	50		SOURCE_ID		Source ID	80
	SERVICECALL.CLOSE_DATETIME			ACT_FINISH		Actual Finish	
	SERVICECALL.CALL_DATE			ACT_START		Actual Start	
	SERVICECALL.DESCRPTION	70		DESCRIPTION		Description	80
	APP_CODE_TEXTS.COX_TEXT	30		IMPACT_SEARCHTEXT		use to search Impact	
	SERVICECALL.INFORMATION	2000		INFORMATION		Information	4000
	APP_CODE_TEXTS.COX_TEXT	30		PRIORITY_SEARCHTEXT		use to search Priority	
	SERVICECALL.CI	10		CL_ID		use to search Configuration Item	
	'CON' + SERVICECALL.CALLER 'EMP' + SERVICECALL.CALLER	3 + 10		CALLER_ID		use to search Caller	
	APP_CODE_TEXTS.COX_TEXT	30		CATEGORY_SEARCHTEXT		use to search Category	
	APP_CODE_TEXTS.COX_TEXT	30		CLASSIFICATION_SEARCHTEXT		use to search Classification	
	APP_CODE_TEXTS.COX_TEXT	30		CLOSURE_SEARCHTEXT		use to search Closure	
	APP_CODE_TEXTS.COX_TEXT	30		MEDIUM_SEARCHTEXT		use to search Medium	
	'EO' + CONTACT.ORGANIZATION 'IO' + EMPLOYEE.ORGANIZATION_UNIT 'EO' + SERVICECALL.CALLER	2 + 10 2 + 10 2 + 10		ORG_ID		use to search Organization	
	SERVICECALL.SER_SRV_ID	10		SERVICE_ID		use to search Service	
	SERVICECALL.SOLUTION	2000		SOLUTION		Solution	4000
	APP_CODE_TEXTS.COX_TEXT	30		STATUS_SEARCHTEXT		use to search Status	
	'EMP' + SERVICECALL.SPECIALIST	3 + 10		TO_PERSON_SOURCEID		Use to search Assignment:To person	
	SERVICECALL.HD_GROUP	10		TO_GROUP_SOURCEID		Use to search Assignment:To group	
	SERVICECALL.REF_NUMBER	10		REF_NUMBER		Assignment:Reference #	50
	SERVICECALL.REMARK	250		REMARK		Assignment:Information from sender	memo
	'EO' + SERVICECALL.RETAINED	2 + 10		TO_ORG_SOURCEID		Use to search Assignment:To external Organization	
	SERVICECALL.CALL_DATE			CREATED		Registration:Created	
	APP_LOGIN_USER.LOGIN_NAME			CREATEDBY_SEARCHTEXT		Use to search Registration:Created by	
	'migration' SERVICECALL.CALLER_NAME2	30		CONTACT_ORGANIZATION		Contact Organization	40
SERVICECALL	SERVICECALL.ID	10	CL_SERVICECALL_DEADLIN	SC_ID		ID	10

Appendix A – Mapping Overview with Field Lengths

			E				
	SERVICECALL.TARGET DATE			DEADLINE		Deadline	
PROGRESS EMPLOYEE CONTACT ADDRESS APP_CODE_TEXTS	ACT.COX_SEARCHCODE + ACT.COX_TEXT + EMPLOYEE.FIRST_NAME + EMPLOYEE.NAME  CONTACT.FIRST_NAME + ADDRESS.NAME1  ADDRESS.NAME1  + ':' + ACTION		CL_SC_HISTORY	SUBJECT	History Line Servicecall	Subject	255
	PROG_DATE 'migration'			CREATED CREATEDBY_SEAR CHTEXT		Registration:Created use to search Registration:Created by	
	SERVICE	10		SC_ID		use to search Service call	
PROGRESS EMPLOYEE CONTACT ADDRESS APP_CODE_TEXTS	ACT.COX_SEARCHCODE + ACT.COX_TEXT + EMPLOYEE.FIRST_NAME + EMPLOYEE.NAME  CONTACT.FIRST_NAME + ADDRESS.NAME1  ADDRESS.NAME1  + ':' + ACTION + '>>'		CL_SC_HISTORY_INFO	SUBJECT	History Line Servicecall	Subject	255
	ACTION PROG_DATE 'migration'	2000		INFORMATION CREATED CREATEDBY_SEAR CHTEXT		Information Registration:Created use to search Registration:Created by	4000
	SERVICE	10		SC_ID		use to search Service call	
DUAL	'Caused by'		CL_SC_CAUSEDBY_RELTYPE	REL_TYPE	Service Event Relation Type	Text	255
	10			ORDERING		Ordering	int
DUAL	'Related to'		CL_SC_RELATED_RELTYPE	REL_TYPE	Service Event Relation Type	Text	255
	20			ORDERING		Ordering	int
<b>Problems</b>							
APP_CODE_TEXTS	COX_TEXT	30	CL_CODE_PR_CAT	TEXT	Problem Category	Text	255
APP_CODE_TEXTS APP_CODES	APP_CODE_TEXTS.CO X_TEXT	30	CL_CODE_PR_CLO	TEXT	Problem Closure code	Text	255
	APP_CODES.COD_ORD ERING	10		ORDERING		Ordering	int
APP_CODE_TEXTS APP_CODES	APP_CODES.TEXTS.C OX_TEXT	30	CL_CODE_PR_STA	TEXT	Problem Status	Text	255
	APP_CODES.COD_ORD ERING	10		ORDERING		Ordering	int
APP_CODE_TEXTS	COX_TEXT	30	CL_CODE_PR_COD	TEXT	Problem Classification	Text	255
PROBLEM DATA_SET APP_CODE_TEXTS APP_LOGIN_USER	DATA_SET.DESCRIP TION	50	CL_PROBLEM	POOL_SEARCHTE XT	Problem	use to search Pool	
	PROBLEM.ID	10		PR_ID		ID	10
	PROBLEM.CLOSE DAT ETIME			ACT_FINISH		Actual Finish	
	PROBLEM.CALL_DATE			ACT_START		Actual Start	
	PROBLEM.DESCRIP TION	70		DESCRIPTION		Description	80
	APP_CODE_TEXTS.CO X_TEXT	30		IMPACT_SEARCHT EXT		use to search Impact	
	PROBLEM.INFORMATIO N	2000		INFORMATION		Information	4000
	APP_CODE_TEXTS. COX_TEXT	30		PRIORITY_SEARC HTEXT		use to search Priority	
	PROBLEM.CI	10		CI_ID		use to search Configuration Item	
	APP_CODE_TEXTS. COX_TEXT	30		CATEGORY_SEAR CHTEXT		use to search Category	
	APP_CODE_TEXTS. COX_TEXT	30		CLASSIFICATION_ SEARCHTEXT		use to search Classification	
	APP_CODE_TEXTS. COX_TEXT	30		CLOSURE_SEARC HTEXT		use to search Closure code	
	APP_CODE_TEXTS. COX_TEXT	30		STATUS_SEARCHT EXT		use to search Status	
	PROBLEM.SOLUTION	2000		SOLUTION		Solution	4000
	'EMP' + PROBLEM.SPECIALIST	3 + 10		TO_PERSON_SOU RCEID		Use to search Assignment:To person	
	PROBLEM.HD_GROUP	10		TO_GROUP_SOUR CEID		Use to search Assignment:To group	
	PROBLEM.REF_NUMBE R	10		REF_NUMBER		Assignment:Reference #	50
	PROBLEM.REMARK	250		REMARK		Assignment:Infromatio memo	

Appendix A – Mapping Overview with Field Lengths

	'EO' + PROBLEM.RETAINED	2 + 10		TO_ORG_SOURCEI D		n from sender Use to search Assignment:To external Organization	
	PROBLEM.CALL_DATE			CREATED		Registration:Created	
	APP_LOGIN_USER.LUS _LOGIN_NAME			CREATEDBY_SEA RCHTEXT		Use to search Registration:Created by	
	'migration'						
PROBLEM	PROBLEM.ID	10	CL_PROBLEM_DEADLINE	PR_ID		ID	10
	PROBLEM.TARGET_DA TE			DEADLINE		Deadline	
PROBLEM_PROGRE SS EMPLOYEE CONTACT ADDRESS APP_CODE_TEXTS	ACT.COX_SEARCHCOD E + ACT.COX_TEXT + EMPLOYEE.FIRST_NAM E + EMPLOYEE.NAME  CONTACT.FIRST_NAME + ADDRESS.NAME1  ADDRESS.NAME1  + ':' + ACTION		CL_PROBLEM_HISTORY	SUBJECT	History Line Problem	Subject	255
	PROG_DATE			CREATED		Registration:Created	
	'migration'			CREATEDBY_SEA RCHTEXT		use to search Registration:Created by	
	PROBLEM	10		PR_ID		use to search Problem	
PROBLEM_PROGRE SS EMPLOYEE CONTACT ADDRESS APP_CODE_TEXTS	ACT.COX_SEARCHCOD E + ACT.COX_TEXT + EMPLOYEE.FIRST_NAM E + EMPLOYEE.NAME  CONTACT.FIRST_NAME + ADDRESS.NAME1  ADDRESS.NAME1  + ':' + ACTION + '>>'		CL_PROBLEM_HISTORY_IN FO	SUBJECT	History Line Problem	Subject	255
	ACTION	2000		INFORMATION		Information	4000
	PROG_DATE			CREATED		Registration:Created	
	'migration'			CREATEDBY_SEA RCHTEXT		use to search Registration:Created by	
	PROBLEM	10		PR_ID		use to search Problem	
SERVICECALL	COMPARE	10	CL_PR_RELATED_SC	PR_ID	Service Event Relation	use to search Problem	
	ID	10		SC_ID		use to search Service call	
	'Related to'			REL_TYPE		use to search Relation type	
<b>Work orders</b>							
APP_CODE_TEXTS APP_CODES	APP_CODE_TEXTS.CO X_TEXT	30	CL_CODE_WO_CLO	TEXT	Work order closure code	Text	255
	APP_CODES.COD_ORD ERING	10		ORDERING		Ordering	int
APP_CODE_TEXTS APP_CODES	APP_CODE_TEXTS.CO X_TEXT	30	CL_CODE_WO_STA	TEXT	Work order status	Text	255
	APP_CODES.COD_ORD ERING	10		ORDERING		Ordering	int
APP_CODE_TEXTS	COX_TEXT	30	CL_CODE_WO_CAT	TEXT	Work order category	Text	255
WORKORDERS APP_LOGIN_USER APP_CODE_TEXTS	DATA_SET.DESCRPTIO N	50	CL_WORKORDER	POOL_SEARCHTE XT	Work order	use to search Pool	
	WORKORDERS.WOR_S PENT_TIME	10		ACT_DURATION		Actual Duration	
	WORKORDERS.WOR_I D	10		WO_ID		ID	10
	WORKORDERS.WOR_C LOSED_DATE			ACT_FINISH		Actual Finish	
	WORKORDERS.WOR_S TART_DATE			ACT_START		Actual Start	
	WORKORDERS.WOR_D ESCRPTION	70		DESCRIPTION		Description	80
	APP_CODE_TEXTS.CO X_TEXT	30		IMPACT_SEARCHT EXT		use to search Impact	
	WORKORDERS.WOR_R EMARKS	2000		INFORMATION		Information	4000
	WORKORDERS.WOR_S TART_DATE			PLAN_START		Planned Start	
	APP_CODE_TEXTS.CO X_TEXT	30		PRIORITY_SEARC HTEXT		use to search Priority	
	APP_CODE_TEXTS.CO X_TEXT	30		CLOSURE_SEARC HTEXT		use to search Closure code	
	APP_CODE_TEXTS.CO X_TEXT	30		STATUS_SEARCHT EXT		use to search Status	
	APP_CODE_TEXTS.CO X_TEXT	30		CATEGORY		use to search Category	
	'EMP' + WORKORDERS.WOR_S	3 + 10		TO_PERSON_SOU RCEID		Use to search Assignment:To person	

Appendix A – Mapping Overview with Field Lengths

	PECIALIST						
	WORKORDERS.WOR_H D_GROUP	10		TO_GROUP_SOUR CEID		Use to search Assignment:To group	
	WORKORDERS.WOR_R EF_NUMBER	10		REF_NUMBER		Assignment:Reference #	50
	WORKORDERS.WOR_D ISPATCH_REMARKS	250		REMARK		Assignment:Infomatio n from sender	memo
	'EO' + WORKORDERS.WOR_C ONTRACT_OUT_ORG	2 + 10		TO_ORG_SOURCEI D		Use to search Assignment:To external Organization	
	'CON' + WORKORDERS.WOR_C ONTRACT_OUT_CON	3 + 10		TO_EXT_PERSON_ SOURCEID		Use to search Assignment:To external Person	
	WORKORDERS.WOR_S TART_DATE			CREATED		Registration:Created	
	APP_LOGIN_USER.LUS _LOGIN_NAME  'migration'			CREATEDBY_SEA RCHTEXT		Use to search Registration:Created by	
WORKORDERS	WORKORDERS.WOR_I D	10		WO_ID		ID	10
	WORKORDERS.WOR_T ARGET_DATE			DEADLINE		Deadline	
WORKORDERS	WOR_ID	10	CL_WO_SC_RELATION	WO_ID	Work order	use to search Work order	
	WOR_CONTEXT_ID	10		SC_ID		use to search Service call	
WORKORDERS	WOR_ID	10	CL_WO_PR_RELATION	WO_ID	Work order	use to search Work order	
	WOR_CONTEXT_ID	10		PR_ID		use to search Problem	
WORKORDERS	WOR_ID	10	CL_WO_CH_RELATION	WO_ID	Work order	use to search Work order	
	WOR_CONTEXT_ID	10		CH_ID		use to search Change	
WO_PROGRESS EMPLOYEE CONTACT ADDRESS APP_CODE_TEXTS	ACT.COX_SEARCHCOD E + ACT.COX_TEXT + EMPLOYEE.FIRST_NAM E + EMPLOYEE.NAME  CONTACT.FIRST_NAME + ADDRESS.NAME1  ADDRESS.NAME1  '+' + WOP_ACTION		CL_WORKORDER_HISTORY	SUBJECT	History Line Workorder	Subject	255
	WOP_PROG_DATE  'migration'			CREATED CREATEDBY_SEA RCHTEXT		Registration:Created use to search Registration:Created by	
	WOP_WO_ID	10		WO_ID		use to search Work order	
WO_PROGRESS EMPLOYEE CONTACT ADDRESS APP_CODE_TEXTS	ACT.COX_SEARCHCOD E + ACT.COX_TEXT + EMPLOYEE.FIRST_NAM E + EMPLOYEE.NAME  CONTACT.FIRST_NAME + ADDRESS.NAME1  ADDRESS.NAME1  '+' + WOP_ACTION + ' >>'		CL_WORKORDER_HISTORY _INFO	SUBJECT	History Line Workorder	Subject	255
	WOP_ACTION WOP_PROG_DATE  'migration'	2000		INFORMATION CREATED CREATEDBY_SEA RCHTEXT		Information Registration:Created use to search Registration:Created by	4000
	WOP_WO_ID	10		WO_ID		use to search Work order	
WORKORDER_CI	WCI_CI	10	CL_WO_CI_RELATION	WO_ID	Configuration Item on Work order	use to search Workorder	
	WCI_WOR_ID	10		CI_ID		use to search ConfigurationItem	
<b>Changes</b>							
APP_CODE_TEXTS	COX_TEXT	30	CL_CODE_CH_CAT	TEXT	Change Category	Text	255
APP_CODE_TEXTS APP_CODES	APP_CODE_TEXTS.CO X_TEXT	30	CL_CODE_CH_CLO	TEXT	Change Closurecode	Text	255
	APP_CODES.COD_ORD ERING	10		ORDERING		Ordering	int
APP_CODE_TEXTS APP_CODES	APP_CODE_TEXTS.CO X_TEXT	30	CL_CODE_CH_STA	TEXT	Change Status	Text	255
	APP_CODES.COD_ORD ERING	10		ORDERING		Ordering	int
APP_CODE_TEXTS	COX_TEXT	30	CL_CODE_CH_COD	TEXT	Change Classification	Text	255
CHANGE DATA_SET APP_CODE_TEXTS APP_LOGIN_USER	DATA_SET.DESCRPTIO N	50	CL_CHANGE	POOL_SEARCHTE XT	Change	use to search Pool	

Appendix A – Mapping Overview with Field Lengths

	CHANGE.ID	10		CH_ID		ID	10
	CHANGE.CLOSED_DATE			ACT_FINISH		Actual Finish	
	CHANGE.CALL_DATE			ACT_START		Actual Start	
	CHANGE.DESCRPTION	70		DESCRIPTION		Description	80
	CHANGE.INFORMATION	2000		INFORMATION		Information	4000
	APP_CODE_TEXTS.COX_TEXT	30		PRIORITY_SEARCHTEXT		use to search Priority	
	CHANGE.CI	10		CI_ID		use to search Configuration Item	
	APP_CODE_TEXTS.COX_TEXT	30		CATEGORY_SEARCHTEXT		use to search Category	
	APP_CODE_TEXTS.COX_TEXT	30		CLASSIFICATION_SEARCHTEXT		use to search Classification	
	APP_CODE_TEXTS.COX_TEXT	30		CLOSURE_SEARCHTEXT		use to search Closure code	
	APP_CODE_TEXTS.COX_TEXT	30		STATUS_SEARCHTEXT		use to search Status	
	CHANGE.DESIRED	255		DESOLUTION		Desired Solution	255
	'EMP' + CHANGE.SPECIALIST	3 + 10		TO_PERSON_SOURCEID		Use to search Assignment:To person	
	CHANGE.HD_GROUP	10		TO_GROUP_SOURCEID		Use to search Assignment:To group	
	CHANGE.REF_NUMBER	10		REF_NUMBER		Assignment:Reference #	50
	CHANGE.REMARK	255		REMARK		Assignment:Information from sender	memo
	'EO' + CHANGE.RETAINED	2 + 10		TO_ORG_SOURCEID		Use to search Assignment:To external Organization	
	CHANGE.CALL_DATE			CREATED		Registration:Created	
	APP_LOGIN_USER.LOGIN_NAME			CREATEDBY_SEARCHTEXT		Use to search Registration:Created by	
	'migration'						
CHANGE	CHANGE.ID	10	CL_CHANGE_DEADLINE	CH_ID		ID	10
	CHANGE.TARGET_DATE			DEADLINE		Deadline	
CHANGE_PROGRESS EMPLOYEE CONTACT ADDRESS APP_CODE_TEXTS	ACT.COX_SEARCHCODE + ACT.COX_TEXT + EMPLOYEE.FIRST_NAME + EMPLOYEE.NAME  CONTACT.FIRST_NAME + ADDRESS.NAME1  ADDRESS.NAME1  + '.' + ACTION		CL_CHANGE_HISTORY	SUBJECT	History Line Change	Subject	255
	PROG_DATE			CREATED		Registration:Created	
	'migration'			CREATEDBY_SEARCHTEXT		use to search Registration:Created by	
CHANGE_PROGRESS EMPLOYEE CONTACT ADDRESS APP_CODE_TEXTS	ACT.COX_SEARCHCODE + ACT.COX_TEXT + EMPLOYEE.FIRST_NAME + EMPLOYEE.NAME  CONTACT.FIRST_NAME + ADDRESS.NAME1  ADDRESS.NAME1  + '.' + ACTION  + '>>'		CL_CHANGE_HISTORY_INFO	SUBJECT	History Line Change	Subject	255
	ACTION	2000		INFORMATION		Information	4000
	PROG_DATE			CREATED		Registration:Created	
	'migration'			CREATEDBY_SEARCHTEXT		use to search Registration:Created by	
SCS_CAUSED_BY_CHANGE	CHANGE	10		CH_ID		use to search Change	
	SCS_CHA_ID	10	CL_CH_CAUSED_BY_SC	CH_ID	Service Event Relation	use to search Change	
	SCS_SER_ID	10		SC_ID		use to search Service call	
	'Caused by'			REL_TYPE		use to search Relation type	
SERVICECALL	COMPARE	10	CL_CH_RELATED_SC	CH_ID	Service Event Relation	use to search Change	
	ID	10		SC_ID		use to search Service call	
	'Related to'			REL_TYPE		use to search Relation type	
PROBLEM	COMPARE	10	CL_CH_RELATED_PR	CH_ID	Service Event Relation	use to search Change	
	ID	10		PR_ID		use to search Problem	
	'Related to'			REL_TYPE		use to search Relation type	
Variable fields							

Appendix A – Mapping Overview with Field Lengths

---

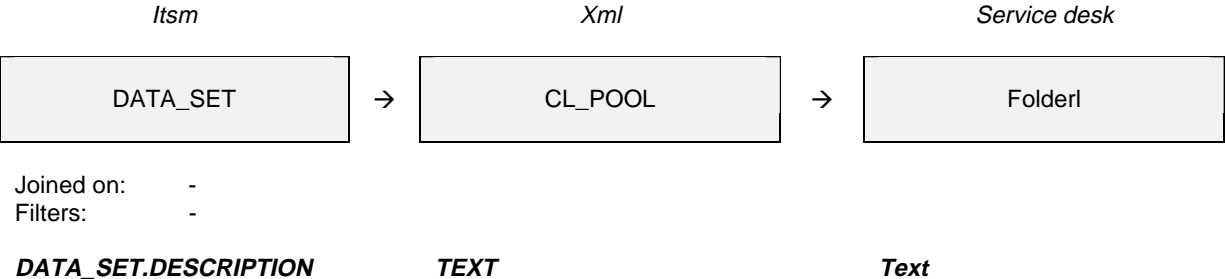
VARIABLE_FIELD VARIABLE_VALUE APP_CODE_TEXTS	VARIABLE_VALUE.ID1	10	CL_VARFIELD_EXAMPLE	SOURCE_ID	Person, Organization, Service call, Change, Problem or Work order	use to search Person, Organization, Service call, Change, Problem or Work order	80
	VARIABLE_VALUE.COL UMN_VALUE	2000		FIELD_VALUE		map to appropriate custom field	

**Appendix B - Detailed Mapping**

The following section describes how ITSM data is mapped to Service Desk data. The data exchange constructions used (like parent-child relations within the XML file), are explained when necessary.

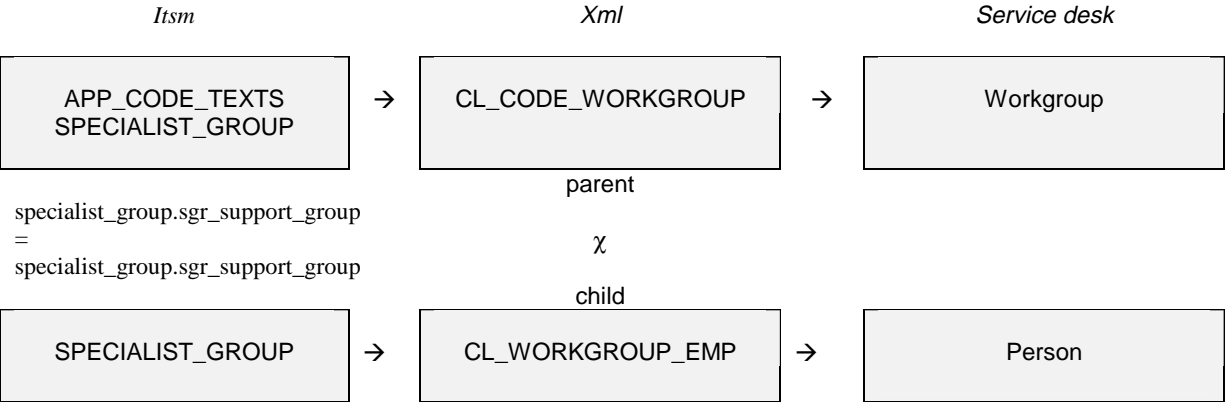
The mapping is provided in tables as follows:

**Table 11- Example of Detailed Mapping**



Parent-child relations will be presented as follows:

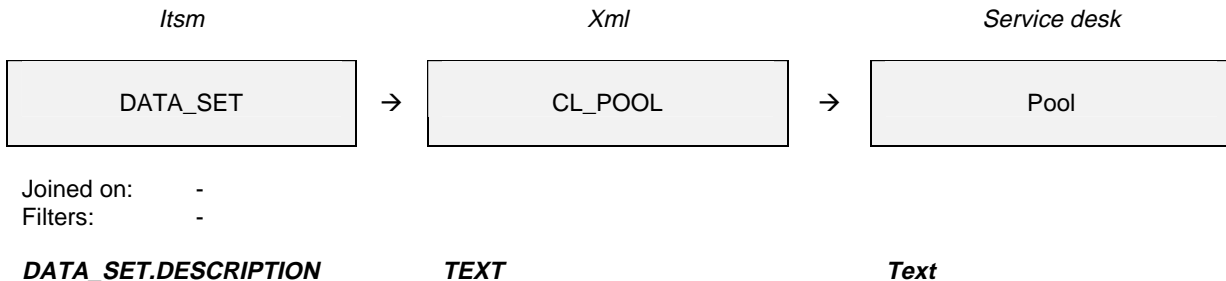
**Table 12 - Example Parent Child Relations**



## Appendix B - .1 Pools

Pools are migrated in a fairly straight-forward manner.

**Table 13 -Mapping Pools**



## Appendix B - .2 Accounts

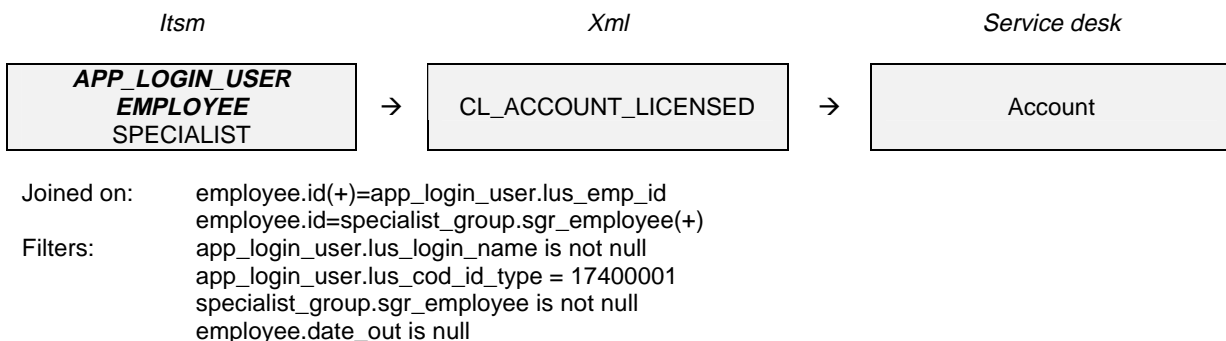
When filling the fields “SSP/Integrations account” and “Blocked”, the following rules are used: Base table is the table App\_login\_user. We then look if this is linked to records in Employee and in Specialist. We also include the field “date\_out” in our decision:

**Table 14- Decision Table for Accounts**

<b>Employee exists</b>	<b>Specialist exists</b>	<b>Date_out field is filled</b>	<b>SSP/Int. account</b>	<b>Blocked</b>
X	x	X	True	True
X	x		False	False
X		X	True	True
X			True	False
	?	?	True	False

The accounts are divided into two classes. One class with the accounts for which the user will need licenses, and one class with the accounts for which it does not. This separation is made on the field “SSP/Int. account”. Accounts with this field set on false will need a license. This way the user can simply check the number of licenses needed after exporting and before importing the accounts.

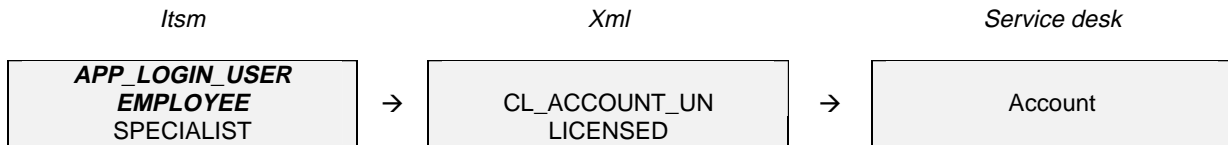
**Table 15-Mapping Licensed Accounts**





<b>APP_LOGIN_USER.LUS_LOGIN_NAME</b>	<b>LOGIN_NAME</b>	<b>Login name</b>
<b>APP_LOGIN_USER.LUS_LOGIN_NAME</b>	<b>DISPLAY_NAME</b>	<b>Display name</b>
<b>'FALSE'</b>	<b>SSP_ACCOUNT</b>	<b>SSP/Integrations account</b>
<b>'TRUE'</b>	<b>CONCURRENT_USER</b>	
<b>'01-01-2001 01:01:01'</b>	<b>PW_MOD_DATE</b>	
<b>'FALSE'<sup>1)</sup></b>	<b>BLOCKED</b>	<b>Blocked</b>
<b>'TRUE'<sup>1)</sup></b>		

**Table 16- Mapping Unlicensed Accounts**



Joined on: employee.id(+)=app\_login\_user.lus\_emp\_id  
 employee.id=specialist\_group.sgr\_employee(+)  
 Filters: app\_login\_user.lus\_login\_name is not null  
 ((app\_login\_user.lus\_cod\_id\_type <> 17400001) or (specialist\_group.sgr\_employee is null) or (employee.date\_out is not null))

<b>APP_LOGIN_USER.LUS_LOGIN_NAME</b>	<b>LOGIN_NAME</b>	<b>Login name</b>
<b>APP_LOGIN_USER.LUS_LOGIN_NAME</b>	<b>DISPLAY_NAME</b>	<b>Display name</b>
<b>'TRUE'</b>	<b>SSP_ACCOUNT</b>	<b>SSP/Integrations account</b>
<b>'01-01-2001 01:01:01'</b>	<b>PW_MOD_DATE</b>	
<b>'FALSE'<sup>1)</sup></b>	<b>BLOCKED</b>	<b>Blocked</b>
<b>'TRUE'<sup>1)</sup></b>		

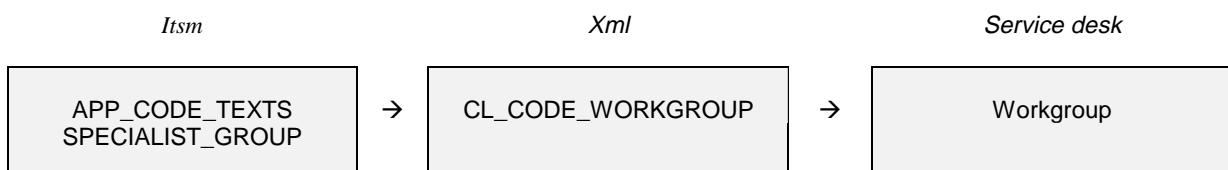
### Appendix B - .3 Organization

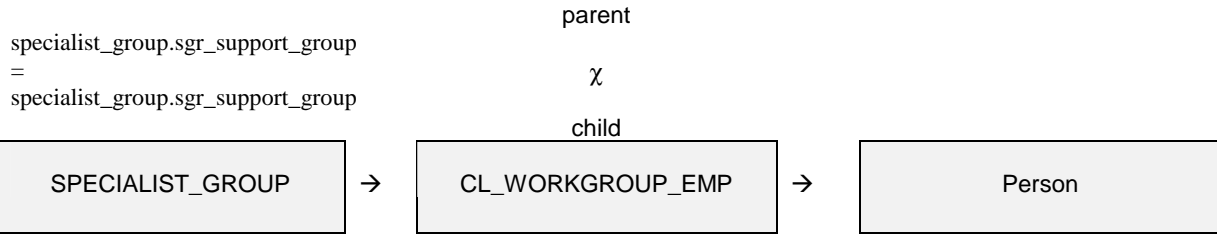
To be able to migrate workgroups and the specialists within a workgroup, a parent-child relation is necessary. This way specialists' workgroups can be set by using the parent attribute.

To migrate the relations between organizations, a special class is used for the relation. For relations between internal organizations for example, we created a class that contains the IDs of parent-organization and child-organization. By importing this class after the organizations have been imported, we can ensure that all organizations can be found when importing the relations.

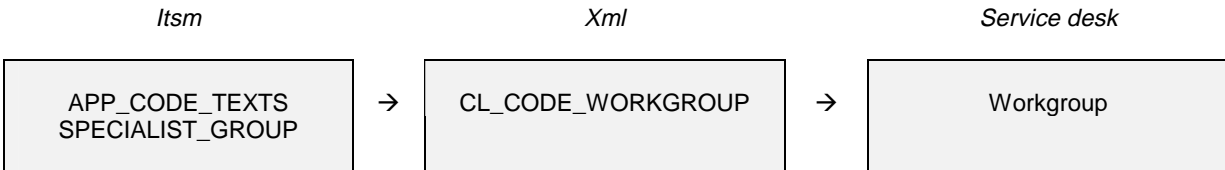
To set the active field of employees we first set it to 'active' for all of them. After that we set the field to 'active' or 'inactive' by using two special classes. This is necessary because the selection on which the 'active' or 'inactive' depends cannot be done in the SQL column definition in the employee class itself. It has to be done in the filters of the two special classes.

**Table 17- Workgroups and Specialists - Parent Child Relations**





**Table 18- Mapping Workgroups**

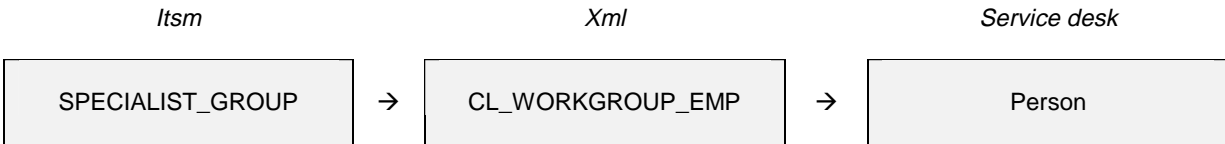


Joined on: specialist\_group.sgr\_support\_group=app\_code\_texts.cox\_cod\_id  
 Filters: cox\_cod\_id between 900000 and 999999  
 cox\_lng\_id = 'gb'

<b>APP_CODE_TEXTS .COX_TEXT</b>	<b>TEXT</b>	<b>Name</b>
<b>APP_CODE_TEXTS</b>	<b>SEARCHCODE <sup>1)</sup></b>	<b>Searchcode</b>
<b>.COX_SEARCHCODE</b>		
<b>APP_CODE_TEXTS</b>	<b>SOURCE_ID</b>	<b>Source ID</b>
<b>.COX_COD_ID</b>		

<sup>1)</sup> In export, replace '\*'','?','\_','%' and '' by '-' and add # before codes which start with 0..9.

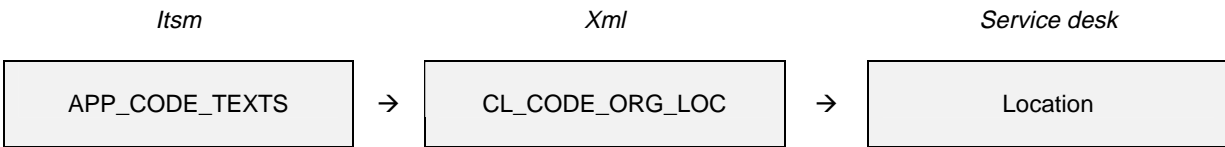
**Table 19- Mapping Specialists**



Joined on: -  
 Filters: -

<b>'EMP' + SGR_EMPLOYEE</b>	<b>EMP_ID</b>	<b>use to search Person</b>
	<b>Parent</b>	<b>use to search Members:Workgroup</b>

**Table 20- Mapping Locations**

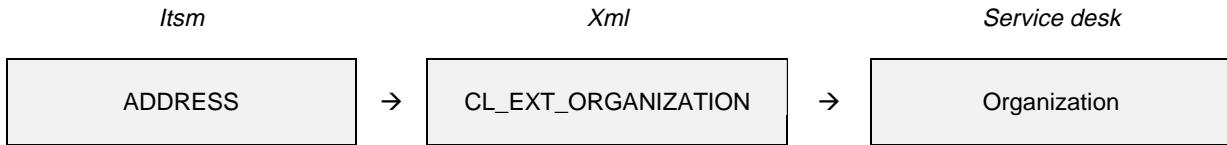


Joined on: -  
 Filters: cox\_cod\_id between 800000 and 899999  
 cox\_lng\_id = 'gb'

<b>COX_TEXT</b>	<b>TEXT <sup>1)</sup></b>	<b>Searchcode</b>
-----------------	---------------------------	-------------------

<sup>1)</sup> In export, replace '\*'','?','\_','%' and '' by '-' and add # before codes which start with 0..9.

**Table 21- Mapping External Organizations**



Joined on: timezone.tim\_id = address.adr\_tim\_id  
 Filters: address.sub\_type=1

<b>EMAIL</b>	<b>EMAIL</b>	<b>E-mail</b>
<b>NAME1</b>	<b>NAME1</b>	<b>Name1</b>
<b>NAME2</b>	<b>NAME2</b>	<b>Name2</b>
<b>REMARK</b>	<b>REMARK</b>	<b>Remark</b>
<b>SEARCHCODE</b>	<b>SEARCHCODE</b> <sup>1)</sup>	<b>Search code</b>
<b>'EO' + ID</b>	<b>ORG_ID</b>	<b>Source ID</b>
<b>'EXTERNAL'</b>	<b>CATEGORY</b>	<b>Category</b> <sup>2)</sup>
<b>TIMEZONE.TIM_SEARCHCODE</b>	<b>TIMEZONE</b>	<b>Timezone</b>
<b>'ACTIVE'</b>	<b>STATUS</b>	<b>Status</b> <sup>3)</sup>

<sup>1)</sup> In export, replace '\*', '?', '\_', '%' and ' ' by '-' and add # before codes which start with 0..9.

<sup>2)</sup> Use import mapping:  
 EXTERNAL → Company

<sup>3)</sup> Use import mapping:  
 ACTIVE → Active

**Table 22 - Mapping External Organisations Visiting Addresses**

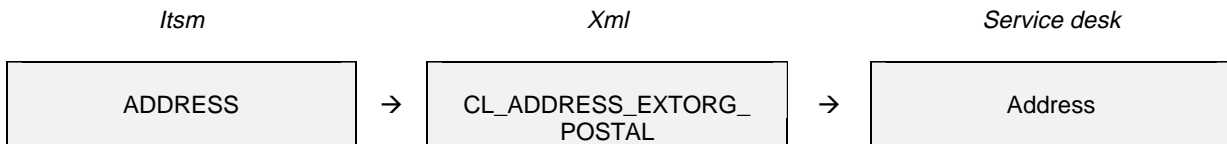


Joined on: -  
 Filters: address.visiting\_address1 is not null  
 address.sub\_type=1

<b>VISITING_ADDRESS1</b>	<b>STREET1</b>	<b>Street1</b>
<b>VISITING_ADDRESS2</b>	<b>STREET2</b>	<b>Street2</b>
<b>VISITNG_ZIPCODE</b>	<b>ZIP</b>	<b>Zip/Postal code</b>
<b>VISITING_CITY</b>	<b>CITY</b>	<b>City</b>
<b>VISTING_REGION</b>	<b>STATE</b>	<b>State/Province</b>
<b>COUNTRY</b>	<b>COUNTRY</b>	<b>Country/Region</b>
<b>'EO' + ID</b>	<b>ORG_ID</b>	<b>use to search Organization</b>
<b>'BUSINESS'</b>	<b>TYPE</b>	<b>Address type</b> <sup>1)</sup>

<sup>1)</sup> Use import mapping:  
 BUSINESS → Business

**Table 23- Mapping External Organizations Postal Addresses**



Joined on: -  
 Filters: address.postal\_address1 is not null  
 address.sub\_type=1

<b>POSTAL_ADDRESS1</b>	<b>STREET1</b>	<b>Street1</b>
<b>POSTAL_ADDRESS2</b>	<b>STREET2</b>	<b>Street2</b>

<b>POSTAL_ZIPCODE</b>	<b>ZIP</b>	<b>Zip/Postal code</b>
<b>POSTAL_CITY</b>	<b>CITY</b>	<b>City</b>
<b>POSTAL_REGION</b>	<b>STATE</b>	<b>State/Province</b>
<b>COUNTRY</b>	<b>COUNTRY</b>	<b>Country/Region</b>
<b>'EO' + ID</b>	<b>ORG_ID</b>	<b>use to search Organization</b>
<b>'MAIL'</b>	<b>TYPE</b>	<b>Address type <sup>1)</sup></b>
<sup>1)</sup> Use import mapping: MAIL → Mail		

**Table 24- Mapping External Organizations Telephone Numbers A**

<i>Itsm</i>		<i>Xml</i>		<i>Service desk</i>
ADDRESS	→	CL_TEL_EXTORG1	→	Telephone
Joined on: - Filters: address.telephone1 is not null address.sub_type=1				
<b>ADDRES.TELEPHONE1</b>		<b>TELNO</b>		<b>Number</b>
<b>'EO' + ADDRES.ID</b>		<b>ORG_ID</b>		<b>use to search Organization</b>
<b>'1'</b>		<b>PRIMARY</b>		<b>Primary</b>
<b>'BUSINESS'</b>		<b>TYPE</b>		<b>Type <sup>1)</sup></b>
<sup>1)</sup> Use import mapping: BUSINESS → Business				

**Table 25- Mapping External Organization Telephone Numbers B**

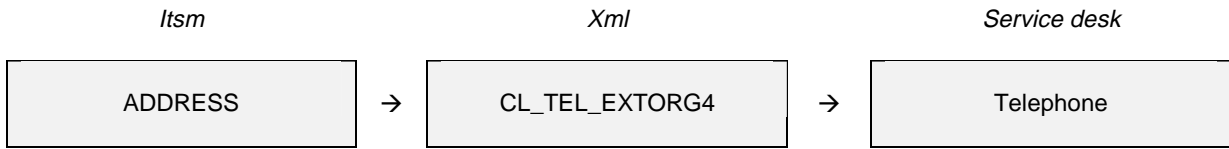
<i>Itsm</i>		<i>Xml</i>		<i>Service desk</i>
ADDRESS	→	CL_TEL_EXTORG2	→	Telephone
Joined on: - Filters: address.telephone2 is not null address.sub_type=1				
<b>ADDRES.TELEPHONE2</b>		<b>TELNO</b>		<b>Number</b>
<b>'EO' + ADDRES.ID</b>		<b>ORG_ID</b>		<b>use to search Organization</b>
<b>'BUSINESS'</b>		<b>TYPE</b>		<b>Type <sup>1)</sup></b>
<sup>1)</sup> Use import mapping: BUSINESS → Business				

**Table 26- Mapping External Organizations Telephone Numbers C**

<i>Itsm</i>		<i>Xml</i>		<i>Service desk</i>
ADDRESS	→	CL_TEL_EXTORG3	→	Telephone
Joined on: - Filters: address.telephone3 is not null address.sub_type=1				
<b>ADDRES.TELEPHONE3</b>		<b>TELNO</b>		<b>Number</b>
<b>'EO' + ADDRES.ID</b>		<b>ORG_ID</b>		<b>use to search Organization</b>
<b>'BUSINESS'</b>		<b>TYPE</b>		<b>Type <sup>1)</sup></b>

<sup>1)</sup> Use import mapping:  
BUSINESS → Business

**Table 27- Mapping External Organizations Fax Numbers**



Joined on: -  
Filters: address.fax is not null  
address.sub\_type=1

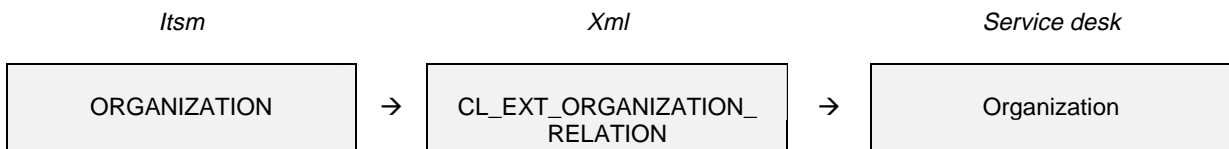
**ADDRES.FAX**  
**'EO' + ADDRES.ID**  
**'FAX'**

**FAXNO**  
**ORG\_ID**  
**TYPE**

**Number**  
**use to search Organization**  
**Type 1)**

<sup>1)</sup> Use import mapping:  
FAX → Fax

**Table 28- Mapping External Organizations Relations**



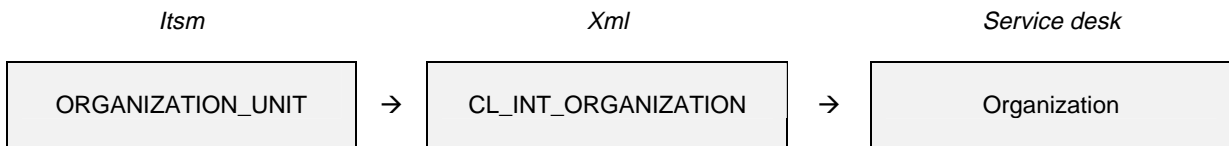
Joined on: -  
Filters: organization.parent is not null

**'EO' + ADDRESS**  
**'EO' + PARENT**

**ORG\_ID**  
**PARENT\_ID**

**use to search Organization**  
**use to search Parent**

**Table 29- Mapping Internal Organizations**



Joined on: Timezone.tim\_id = organization\_unit.oun\_tim\_id  
Filters: -

**NAME**  
**REMARK**  
**SEARCHCODE**  
**'IO' + ID**  
**'INTERNAL'**  
**TIMEZONE.TIM\_SEARCHCODE**  
**'ACTIVE'**

**NAME**  
**REMARK**  
**SEARCHCODE <sup>1)</sup>**  
**ORG\_ID**  
**CATEGORY**  
**TIMEZONE**  
**STATUS**

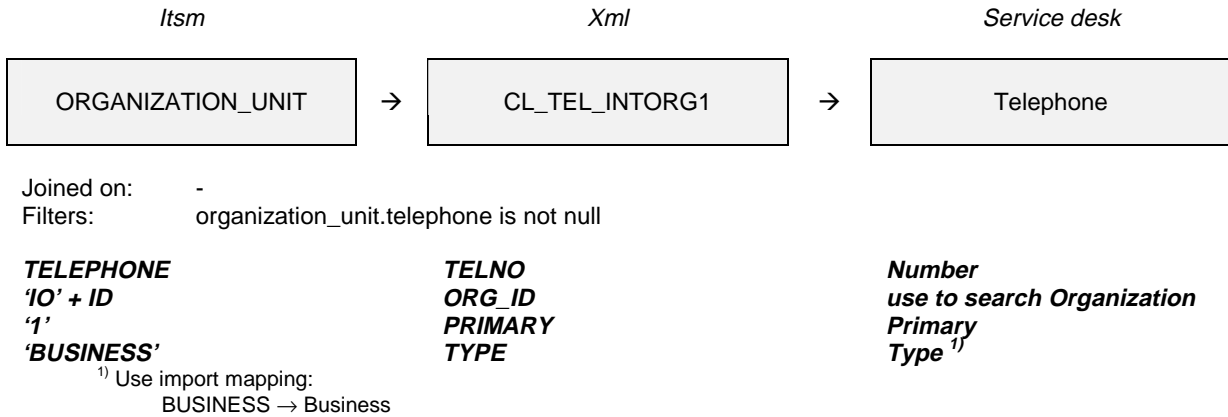
**Name1**  
**Remark**  
**Search code**  
**Source ID**  
**Category <sup>2)</sup>**  
**Timezone**  
**Status <sup>3)</sup>**

<sup>1)</sup> In export, replace '\*' , '?' , '\_' , '%' and ' ' by '-' and add # before codes which start with 0..9.

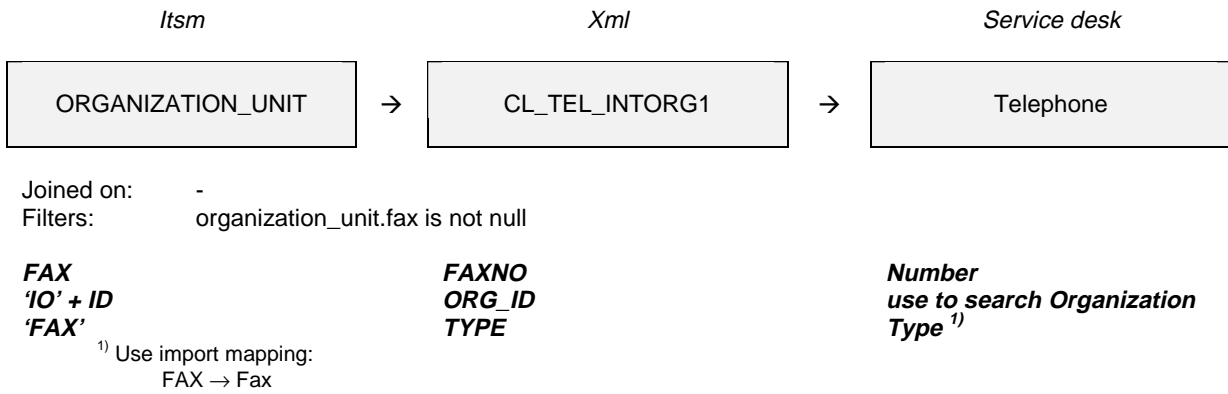
<sup>2)</sup> Use import mapping:  
INTERNAL → Organization

<sup>3)</sup> Use import mapping:  
ACTIVE → Active

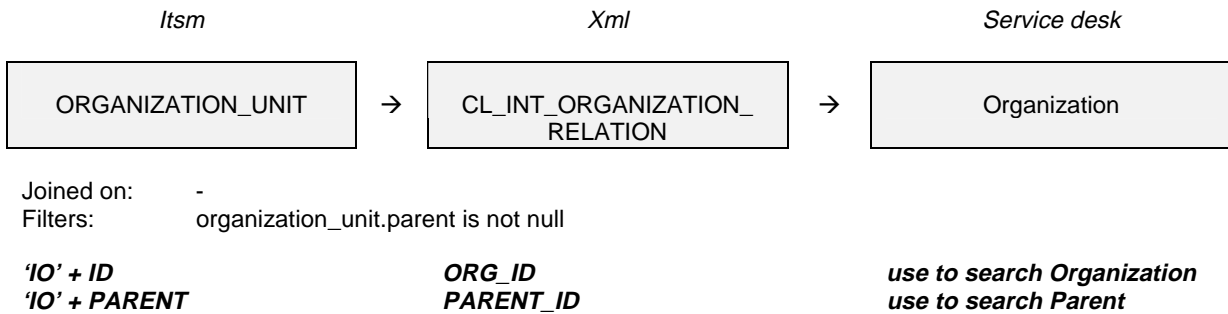
**Table 30- Mapping Internal Organization Telephone Numbers**



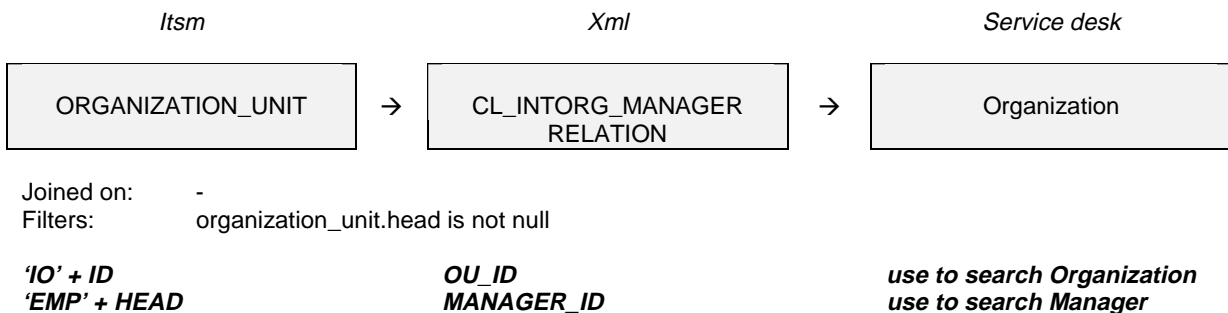
**Table 31- Mapping Internal Organizations Fax Numbers**



**Table 32 - Mapping Internal Organizations Relations**

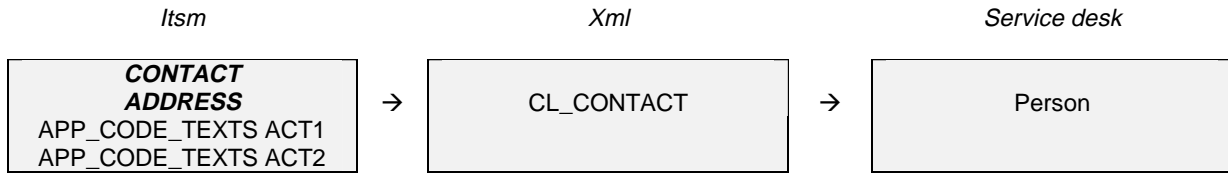


**Table 33- Mapping Internal Organizations Manager Relations**



## Appendix B - .4 Contacts

**Table 34- Mapping Contacts**



Joined on: contact.address(+)=address.id  
 contact.sex=act1.cox\_cod\_id(+)  
 contact.position=act2.cox\_cod\_id(+)  
 address.adr\_tim\_id = timezone.tim\_id

Filters: (act1.cox\_lng\_id='gb' or act1.cox\_lng\_id is null)  
 (act2.cox\_lng\_id='gb' or act2.cox\_lng\_id is null)  
 address.sub\_type=2

<b>ADDRESS.EMAIL</b>	<b>EMAIL</b>	<b>E-mail</b>
<b>CONTACT.FIRST_NAME</b>	<b>FULLNAME_FIRSTNAME</b>	<b>Full name : First name</b>
<b>ADDRESS.NAME1</b>	<b>FULLNAME_LASTNAME</b>	<b>Full name : Last name</b>
<b>CONTACT.TITLE1</b>	<b>FULLNAME_TITLE</b>	<b>Full name : Title</b>
<b>CONTACT.BIRTHDAY</b>	<b>BIRTHDAY</b>	<b>Date of Birth</b>
<b>ACT.COX_TEXT</b>	<b>GENDER</b>	<b>Gender <sup>2)</sup></b>
<b>CONTACT.INITIALS</b>	<b>INITIALS</b>	<b>Initials</b>
<b>ACT.COX_TEXT</b>	<b>JOBTITLE</b>	<b>Job title</b>
if contact.attn is not null: <b>CONTACT.ATTN</b>	<b>NAME <sup>5)</sup></b>	<b>Name</b>
else: <b>ADDRESS.NAME1</b>		
<b>'EO' + CONTACT.ORGANIZATION</b>	<b>ORG_ID</b>	<b>use to search Organization</b>
<b>ADDRESS.REMARK</b>	<b>REMARK</b>	<b>Remark</b>
<b>ADDRESS.SEARCHCODE</b>	<b>SEARCHCODE <sup>1)</sup></b>	<b>Search code</b>
<b>'CON' + CONTACT.ADDRESS</b>	<b>CONTACT_ID</b>	<b>Source ID</b>
<b>'CONTACT'</b>	<b>CATEGORY</b>	<b>Category <sup>3)</sup></b>
<b>TIMEZONE.TIM_SEARCHCODE</b>	<b>TIMEZONE</b>	<b>Timezone</b>
<b>'ACTIVE'</b>	<b>STATUS</b>	<b>Status <sup>4)</sup></b>

<sup>1)</sup> In export, replace '\*', '?', '\_', '%', and ' ' by ':' and add # before codes which start with 0..9.

<sup>2)</sup> Use import mapping:  
 MALE → Male  
 FEMALE → Female

<sup>3)</sup> Use import mapping:  
 CONTACT → Contact

<sup>4)</sup> Use import mapping:  
 ACTIVE → Active

<sup>5)</sup> Truncate to 50

**Table 35: Mapping Contacts Visiting Addresses**



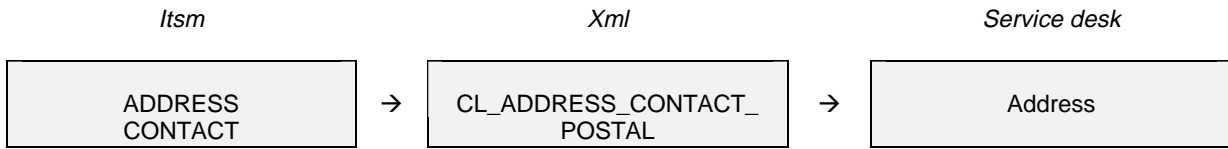
Joined on: address.id(+)=contact.address

Filters: address.visiting\_address1 is not null  
 address.sub\_type=2  
 address.visiting\_city is not null

<b>ADDRESS.VISITING_ADDRESS1</b>	<b>STREET1</b>	<b>Street1</b>
<b>ADDRESS.VISITING_ADDRESS2</b>	<b>STREET2</b>	<b>Street2</b>
<b>ADDRESS.VISITNG_ZIPCODE</b>	<b>ZIP</b>	<b>Zip/Postal code</b>
<b>ADDRESS.VISITING_CITY</b>	<b>CITY</b>	<b>City</b>
<b>ADDRESS.VISTING_REGION</b>	<b>STATE</b>	<b>State/Province</b>
<b>ADDRESS.COUNTRY</b>	<b>COUNTRY</b>	<b>Country/Region</b>
<b>'CON' + ADDRESS.ID</b>	<b>CONTACT_ID</b>	<b>Use to search Person</b>
<b>'BUSINESS'</b>	<b>TYPE</b>	<b>Address type <sup>1)</sup></b>

<sup>1)</sup> Use import mapping:  
BUSINESS → Business

**Table 36: Mapping Contacts Postal Addresses**



Joined on: address.id(+)=contact.address  
Filters: address.postal\_address1 is not null  
address.sub\_type=2  
address.postal\_city is not null

<b>ADDRESS.POSTAL_ADDRESS1</b>	<b>STREET1</b>	<b>Street1</b>
<b>ADDRESS.POSTAL_ADDRESS2</b>	<b>STREET2</b>	<b>Street2</b>
<b>ADDRESS.POSTAL_ZIPCODE</b>	<b>ZIP</b>	<b>Zip/Postal code</b>
<b>ADDRESS.POSTAL_CITY</b>	<b>CITY</b>	<b>City</b>
<b>ADDRESS.POSTAL_REGION</b>	<b>STATE</b>	<b>State/Province</b>
<b>ADDRESS.COUNTRY</b>	<b>COUNTRY</b>	<b>Country/Region</b>
<b>'CON' + ADDRESS.ID</b>	<b>CONTACT_ID</b>	<b>Use to search Person</b>
<b>'MAIL'</b>	<b>TYPE</b>	<b>Address type <sup>1)</sup></b>

<sup>1)</sup> Use import mapping:  
MAIL → Mail

**Table 37: Mapping Contacts Telephone Numbers A**

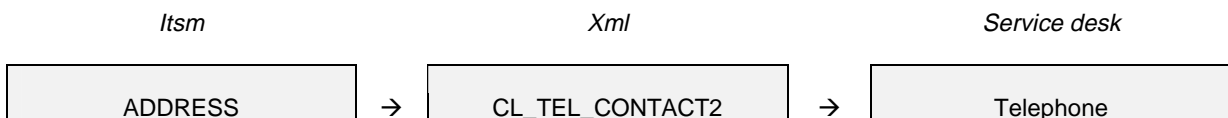


Joined on: address.id(+)=contact.address  
Filters: address.telephone1 is not null  
address.sub\_type=2

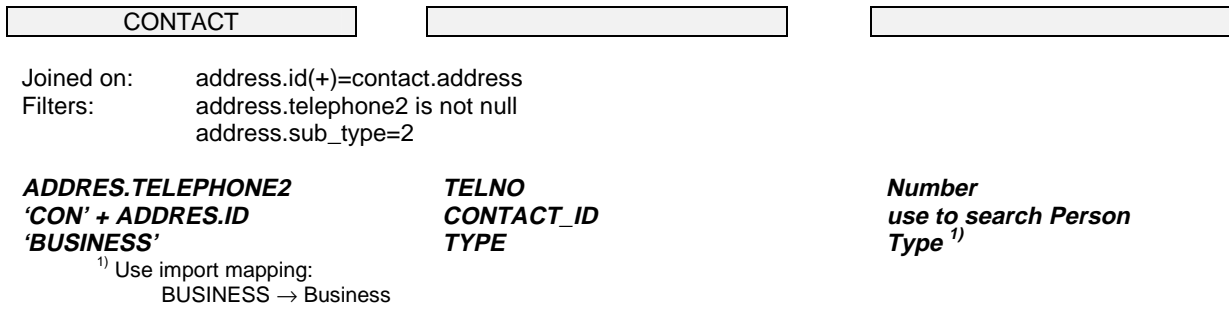
<b>ADDRES.TELEPHONE1</b>	<b>TELNO</b>	<b>Number</b>
<b>'CON' + ADDRESS.ID</b>	<b>CONTACT_ID</b>	<b>Use to search Person</b>
<b>'BUSINESS'</b>	<b>TYPE</b>	<b>Type <sup>1)</sup></b>

<sup>1)</sup> Use import mapping:  
BUSINESS → Business

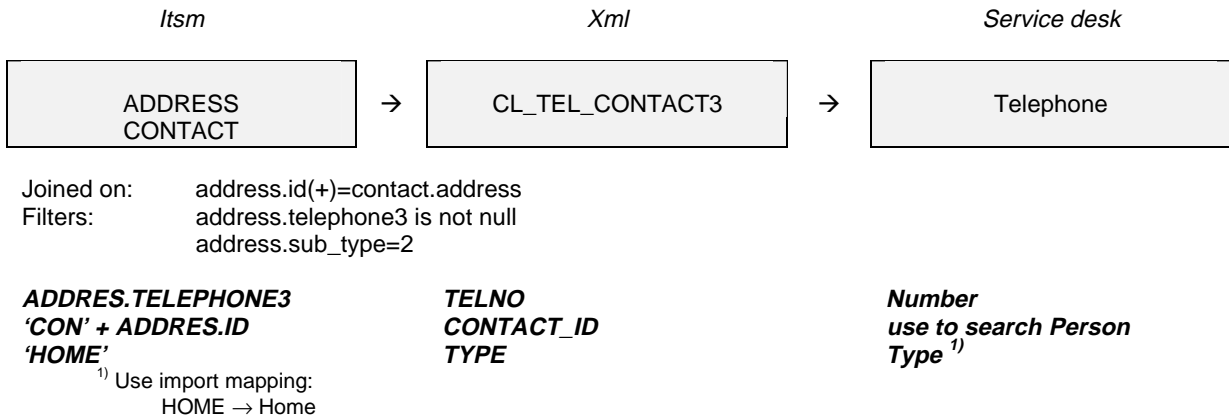
**Table 38: Mapping Contacts Telephone Numbers B**



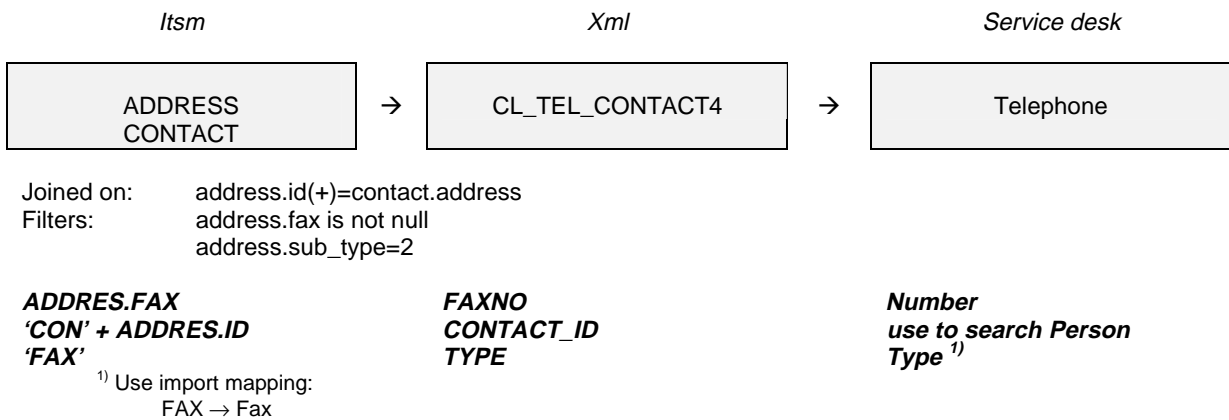




**Table 39- Mapping Contacts Telephone Numbers C**



**Table 40: Mapping Contacts Fax Numbers**



## Appendix B - .5 Employees

**Table 41: Mapping Employees**



Joined on: employee.sex=act1.cox\_cod\_id(+)  
employee.location1\_id=act2.cox\_cod\_id(+)  
employee.position=act3.cox\_cod\_id(+)  
employee.id=app\_login\_user.lus\_emp\_id(+)  
employee.emp\_tim\_id = timezone.tim\_id  
Filters: (app\_login\_user.lus\_cod\_id\_type=17400001 or app\_login\_user.lus\_cod\_id\_type is null)  
(act1.cox\_lng\_id='gb' or act1.cox\_lng\_id is null)  
(act2.cox\_lng\_id='gb' or act2.cox\_lng\_id is null)  
(act3.cox\_lng\_id='gb' or act3.cox\_lng\_id is null)

<b>'EMP' + EMPLOYEE.ID</b>	<b>EMP_ID</b>	<b>Source ID</b>
<b>EMPLOYEE.SEARCHCODE</b>	<b>SEARCHCODE<sup>1)</sup></b>	<b>Search code</b>
<b>EMPLOYEE.FIRST_NAME</b>	<b>FULLNAME_FIRSTNAME</b>	<b>Full name : First name</b>
<b>EMPLOYEE.NAME</b>	<b>FULLNAME_LASTNAME</b>	<b>Full name : Last name</b>
<b>EMPLOYEE.TITLE1</b>	<b>FULLNAME_TITLE</b>	<b>Full name : Title</b>
<b>'IO' +</b>	<b>ORG_ID</b>	<b>use to search Organization</b>
<b>EMPLOYEE.ORGANIZATION_UNI T</b>		
<b>ACT.COX_TEXT</b>	<b>GENDER</b>	<b>Gender<sup>2)</sup></b>
<b>EMPLOYEE.REMARK</b>	<b>REMARK</b>	<b>Remark</b>
<b>ACT.COX_TEXT</b>	<b>LOCATION_SEARCHTEXT<sup>1)</sup></b>	<b>use to search Location</b>
<b>EMPLOYEE.INITIALS</b>	<b>INITIALS</b>	<b>Initials</b>
<b>if employee.attn is not null:</b>	<b>NAME<sup>5)</sup></b>	<b>Name</b>
<b>EMPLOYEE.ATTN</b>		
<b>else:</b>		
<b>EMPLOYEE.NAME</b>		
<b>EMPLOYEE.BIRTHDAY</b>	<b>BIRTHDAY</b>	<b>Date of Birth</b>
<b>EMPLOYEE.EMAIL</b>	<b>EMAIL</b>	<b>E-mail</b>
<b>APP_LOGIN_USER.LUS_LOGIN_ NAME</b>	<b>ACCOUNT_LOGINNAME</b>	<b>Account</b>
<b>TIMEZONE.TIM_SEARCHCODE</b>	<b>TIMEZONE</b>	<b>Timezone</b>
<b>'EMPLOYEE'</b>	<b>CATEGORY</b>	<b>Category<sup>3)</sup></b>
<b>'ACTIVE'</b>	<b>STATUS</b>	<b>Status<sup>4)</sup></b>
<b>ACT.COX_TEXT</b>	<b>JOBTITLE</b>	<b>Job title</b>

<sup>1)</sup> In export, replace '\*' , '?' , '\_' , '%' and ' ' by '-' and add # before codes which start with 0..9.

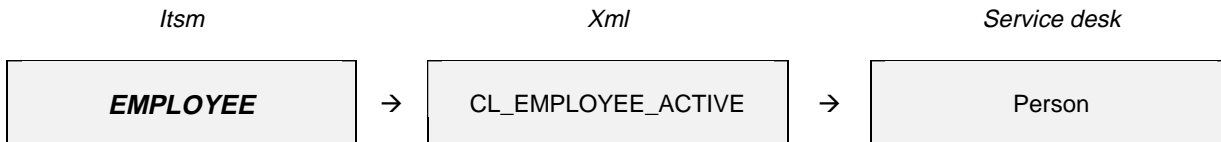
<sup>2)</sup> Use import mapping:  
MALE → Male  
FEMALE → Female

<sup>3)</sup> Use import mapping:  
EMPLOYEE → Employee

<sup>4)</sup> Use import mapping:  
ACTIVE → Active

<sup>5)</sup> Truncate to 50

**Table 42: Mapping Employees Active**

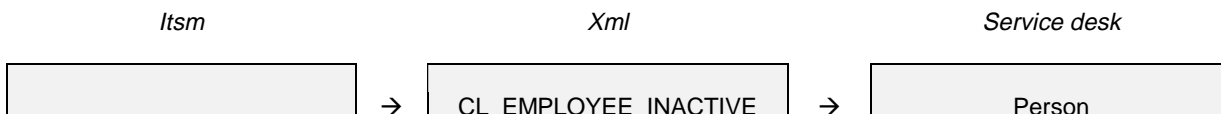


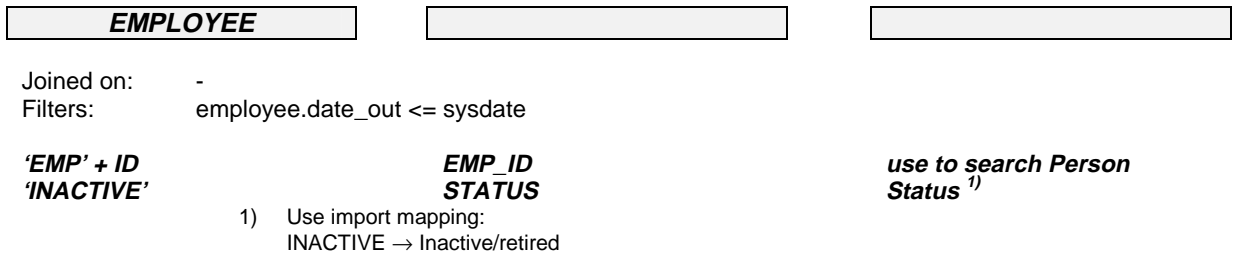
Joined on: -  
Filters: employee.date\_out > sysdate or employee.date\_out is null

<b>'EMP' + ID</b>	<b>EMP_ID</b>	<b>use to search Person</b>
<b>'ACTIVE'</b>	<b>STATUS</b>	<b>Status<sup>1)</sup></b>

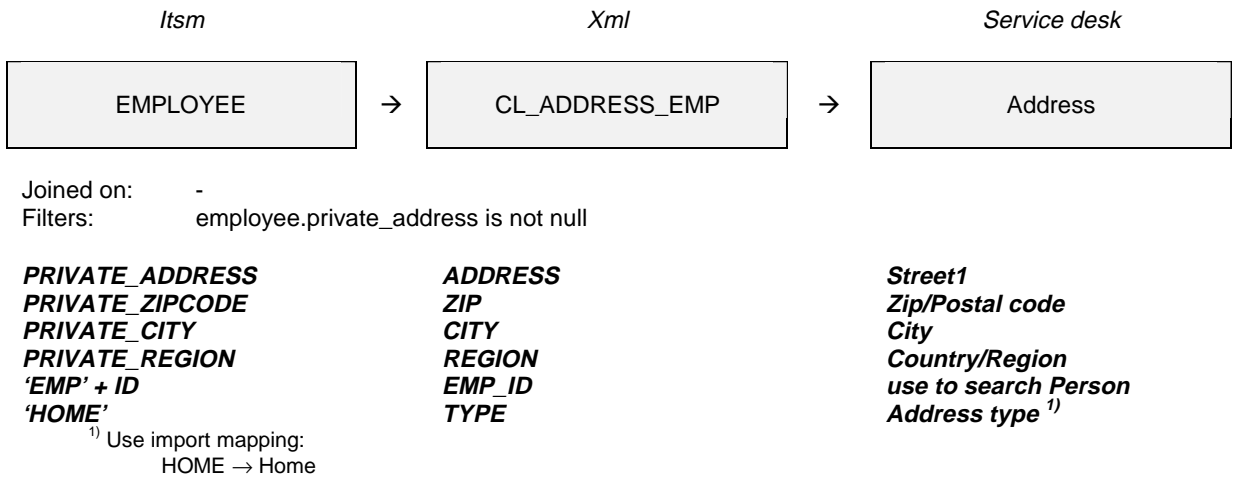
<sup>1)</sup> Use import mapping:  
ACTIVE → Active

**Table 43: Mapping Employees Inactive**

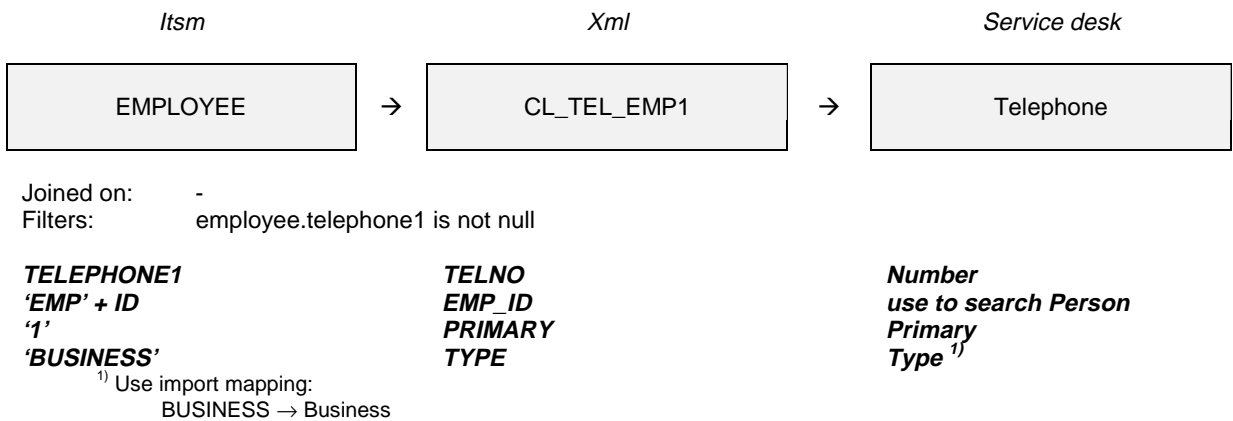




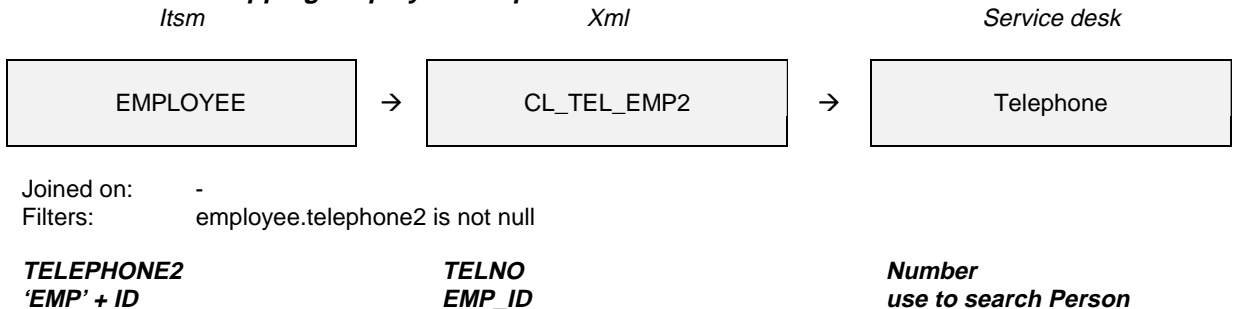
**Table 44: Mapping Employee Addresses**



**Table 45: Mapping Employee Telephone Numbers A**

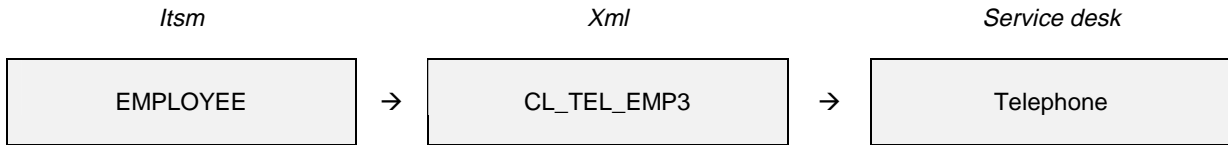


**Table 46: Mapping Employee Telephone Numbers B**



**'BUSINESS'** **TYPE** **Type**<sup>1)</sup>  
 1) Use import mapping:  
 BUSINESS → Business

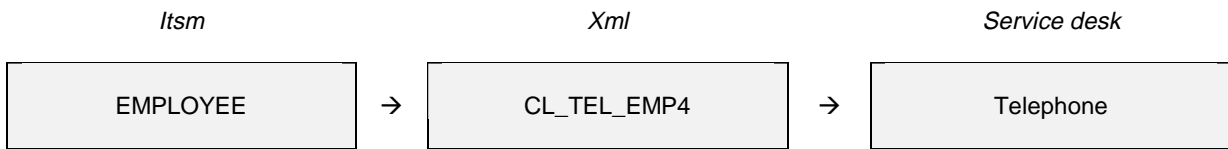
**Table 47: Mapping Employee telephone numbers C**



Joined on: -  
 Filters: employee.private\_telephone is not null

<p><b>PRIVATE TELEPHONE</b>                      'EMP' + ID                      'HOME'</p> <p>1) Use import mapping:                      HOME → Home</p>	<p><b>TELNO</b>                      EMP_ID                      TYPE</p>	<p><b>Number</b>                      use to search Person                      Type<sup>1)</sup></p>
--	---	---

**Table 48 - Mapping Employee Fax Numbers**



Joined on: -  
 Filters: employee.fax is not null

<p><b>FAX</b>                      'EMP' + ID                      'FAX'</p> <p>1) Use import mapping:                      FAX → Fax</p>	<p><b>FAXNO</b>                      EMP_ID                      TYPE</p>	<p><b>Number</b>                      use to search Person                      Type<sup>1)</sup></p>
---	---	---

When importing a table, all entities that it refers to should be already imported at that moment. Exporting and importing the tables in the following order ensures this:

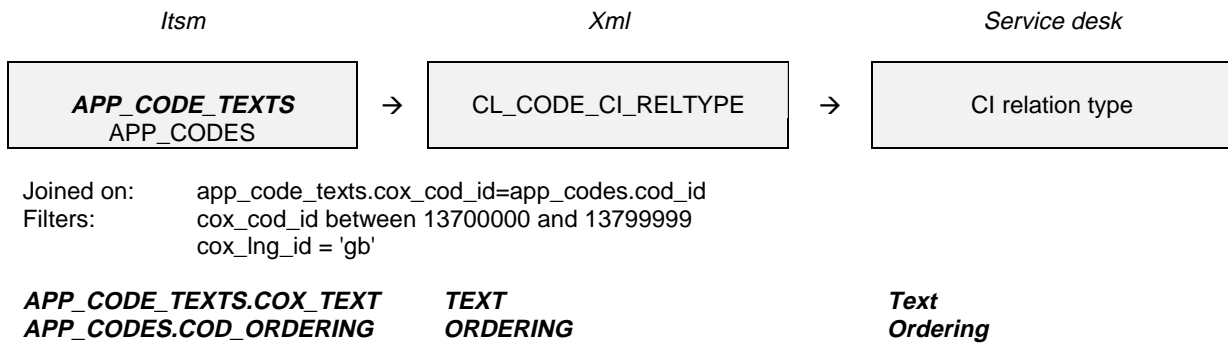
- CL\_CODE\_ORG\_LOC
- CL\_EXT\_ORGANIZATION
- CL\_EXT\_ORGANIZATION\_RELATION
- CL\_ADDRESS\_EXTORG\_VISITING
- CL\_ADDRESS\_EXTORG\_POSTAL
- CL\_TEL\_EXTORG1
- CL\_TEL\_EXTORG2
- CL\_TEL\_EXTORG3
- CL\_TEL\_EXTORG4
- CL\_INT\_ORGANIZATION
- CL\_INT\_ORGANIZATION\_RELATION
- CL\_TEL\_INTORG1
- CL\_TEL\_INTORG2
- CL\_CONTACT
- CL\_ADDRESS\_CONTACT\_VISITING
- CL\_ADDRESS\_CONTACT\_POSTAL

CL\_TEL\_CONTACT1  
 CL\_TEL\_CONTACT2  
 CL\_TEL\_CONTACT3  
 CL\_TEL\_CONTACT4  
 CL\_EMPLOYEE  
 CL\_EMPLOYEE\_ACTIVE  
 CL\_EMPLOYEE\_INACTIVE  
 CL\_ADDRESS\_EMP  
 CL\_TEL\_EMP1  
 CL\_TEL\_EMP2  
 CL\_TEL\_EMP3  
 CL\_TEL\_EMP4  
 CL\_WORKGROUP\_EMP  
 CL\_CODE\_WORKGROUP  
 CL\_INTORG\_MANAGER\_RELATION

### Appendix B - .6 CI Relation Types

The references to the CI relation types in the other CMDB classes will only succeed if the relation types have reverse relations defined.

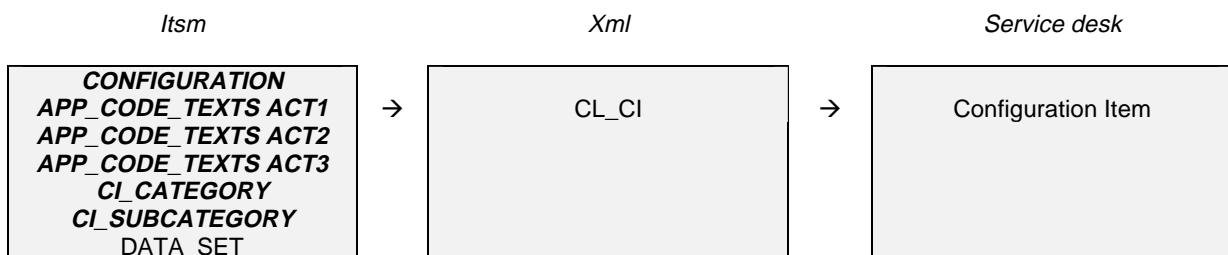
**Table 49- Example Mapping CI Relation Types**



### Appendix B - .7 Configuration Items

To migrate the component relations between CIs, two classes are introduced with a parent-child-relation. The same has been done for the user-relations between CIs and persons. For CI relations we create a class that contains the relation type: CI-from- and CI-to-IDs.

**Table 50- Mapping Configuration Items**



Joined on: configuration.brand=act1.cox\_cod\_id(+)  
 configuration.status=act2.cox\_cod\_id(+)  
 configuration.auth\_file=data\_set.id(+)  
 configuration.category=ci\_category.id(+)  
 configuration.subcategory=ci\_subcategory.id(+)  
 configuration.main\_category=act3.cox\_cod\_id(+)  
 configuration.main\_category=ci\_category.main\_category(+)  
 configuration.main\_category=ci\_subcategory.main\_category(+)  
 configuration.category=ci\_subcategory.category(+)

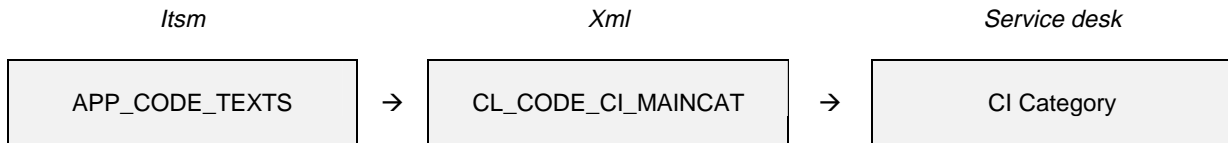
Filters: (act1.cox\_lng\_id='gb' or act1.cox\_lng\_id is null)  
 (act2.cox\_lng\_id='gb' or act2.cox\_lng\_id is null)  
 (act3.cox\_lng\_id='gb' or act3.cox\_lng\_id is null)

<b>CONFIGURATION.ID</b>	<b>CI_ID</b>	<b>Source ID</b>
if configuration.administrator_type = 11300003: 'EMP' + CONFIGURATION.ADMINISTRAT OR if configuration.administrator_type = 11300001: 'CON' + CONFIGURATION.ADMINISTRAT OR else null	<b>ADMIN_PERSON_ID</b>	<b>use to search Admin. Person</b>
if configuration.administrator_type = 11300004: 'IO' + CONFIGURATION.ADMINISTRAT OR if configuration.administrator_type = 11300002: 'EO' + CONFIGURATION.ADMINISTRAT OR else null	<b>ADMIN_ORG_ID</b>	<b>use to search Admin. Org.</b>
<b>ACT.COX_TEXT</b> if ci_subcategory.code is not null: <b>CI_SUBCATEGORY.CODE</b> else if ci_category.code is not null: <b>CI_CATEGORY.CODE</b> else:	<b>BRAND_SEARCHTEXT</b> <b>CATEGORY_SEARCHTEXT</b>	<b>use to search Brand</b> <b>use to search Category</b>
<b>ACT.COX_TEXT</b> if configuration.location1 is not null: <b>CONFIGURATION.LOCATION1</b> else: <b>CONFIGURATION.LOCATION2</b>	<b>LOCATION_SEARCHTEXT <sup>1)</sup></b>	<b>use to search Location</b>
<b>CONFIGURATION.NAME1</b> <b>CONFIGURATION.NAME2</b> <b>CONFIGURATION.ORDER_NUMB ER</b>	<b>NAME1</b> <b>NAME2</b> <b>ORDER_NO</b>	<b>Name 1</b> <b>Name 2</b> <b>Order number</b>
if configuration.owner_type = 1400002: 'IO' + <b>CONFIGURATION.OWNER</b> if configuration.owner_type = 1400001: 'EO' + <b>CONFIGURATION.OWNER</b> else null	<b>OWNER_ORG_ID</b>	<b>use to search Owner Org.</b>

<b>CONFIGURATION.PRICE</b>	<b>PRICE</b>	<b>Price</b>
<b>CONFIGURATION.PURCHASE_DATE</b>	<b>PURCHASE_DATE</b>	<b>Pruchase date</b>
<b>CONFIGURATION.REMARK</b>	<b>REMARK</b>	<b>Remark</b>
<b>CONFIGURATION.CODE</b>	<b>SEARCHCODE <sup>1)</sup></b>	<b>Search code</b>
<b>CONFIGURATION.SERIAL_NUMBER</b>	<b>SERIAL_NO</b>	<b>Serial Number</b>
<b>ACT.COX_TEXT</b> 'EO' +	<b>STATUS_SEARCHTEXT</b> <b>SUPPLIER_ID</b>	<b>use to search Status</b> <b>use to search Supplier</b>
<b>CONFIGURATION.SUPPLIER</b>		
<b>CONFIGURATION.WARRANTY_DATE</b>	<b>WARRANTY_DATE</b>	<b>Warranty date</b>
<b>DATA_SET.DESCRPTION</b>	<b>POOL_SEARCHTEXT</b>	<b>use to search Pool</b>
<b>CONFIGURATION.MAX_INST</b>	<b>MAX_INST</b>	<b>Max. Installations</b>
if configuration.kind = 11700002: <b>TRUE</b>	<b>UNIQUE</b>	<b>Unique</b>
else: <b>FALSE</b>		

<sup>1)</sup> In export, replace '\*', '?', '\_', '%', and ' ' by '-' and add # before codes which start with 0..9.

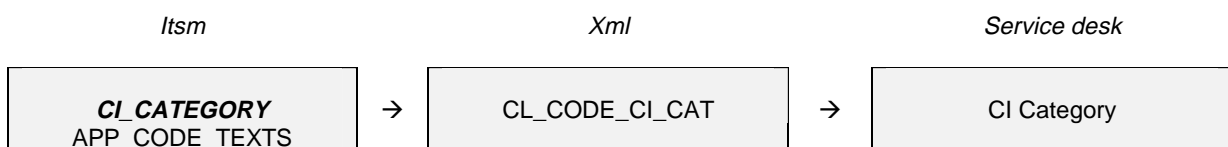
**Table 51: Mapping Configuration Items Main Categories**



Joined on: -  
 Filters: cox\_cod\_id between 12400000 and 12499999  
 cox\_lng\_id = 'gb'

<b>COX_TEXT</b>	<b>TEXT</b>	<b>Text</b>
-----------------	-------------	-------------

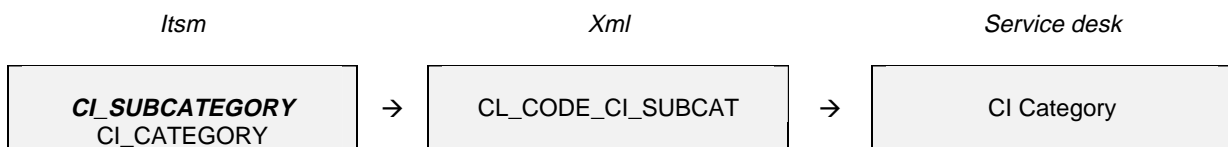
**Table 52: Mapping Configuration Items Categories**



Joined on: ci\_category.main\_category=app\_code\_texts.cox\_cod\_id(+)  
 Filters: (app\_code\_texts.cox\_lng\_id='gb' or app\_code\_texts.cox\_lng\_id is null)

<b>CI_CATEGORY.DESCRPTION</b> <b>APP_CODE_TEXTS.COX_TEXT</b>	<b>TEXT</b> <b>PARENT_SEARCHTEXT</b>	<b>Text</b> <b>use to search Parent</b>
---	---	--

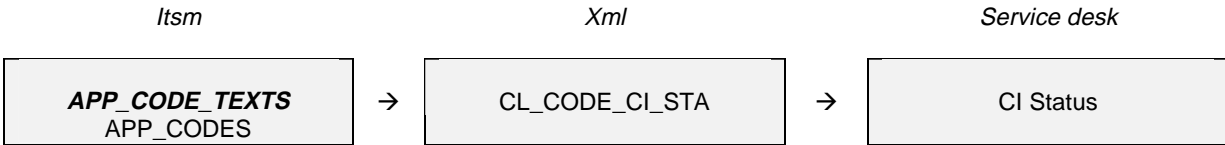
**Table 53: Mapping Configuration Items sub categories**



Joined on: ci\_subcategory.category=ci\_category.id(+)  
 ci\_subcategory.main\_category=ci\_category.main\_category(+)  
 Filters: -

<b>CI_SUBCATEGORY. DESCRIPTION</b>	<b>TEXT</b>	<b>Text</b>
<b>CI_CATEGORY. DESCRIPTION</b>	<b>PARENT_SEARCHTEXT</b>	<b>use to search Parent</b>

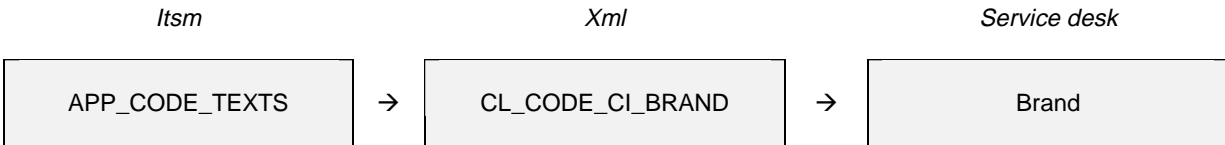
**Table 54: Mapping Configuration Items statuses**



Joined on: app\_code\_texts.cox\_cod\_id=app\_codes.cod\_id  
 Filters: cox\_cod\_id between 11900000 and 11999999  
 cox\_lng\_id = 'gb'

<b>APP_CODE_TEXTS.COX_TEXT</b>	<b>TEXT</b>	<b>Text</b>
<b>APP_CODES.COD_ORDERING</b>	<b>ORDERING</b>	<b>Ordering</b>

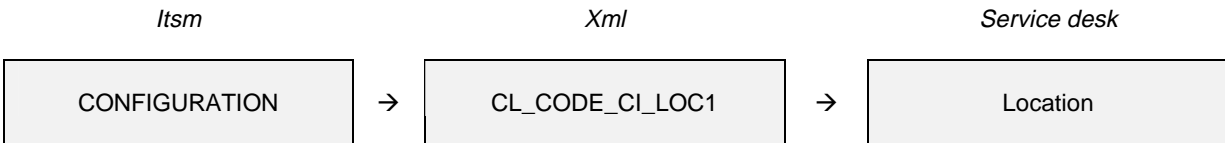
**Table 55: Mapping Configuration Items brands**



Joined on: -  
 Filters: cox\_cod\_id between 15300000 and 15399999  
 cox\_lng\_id = 'gb'

<b>COX_TEXT</b>	<b>TEXT</b>	<b>Text</b>
-----------------	-------------	-------------

**Table 56: Mapping Configuration Items locations A**

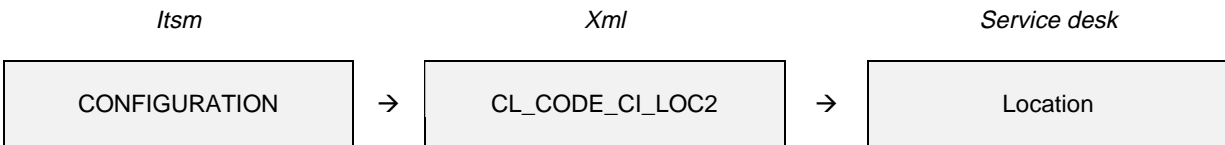


Joined on: -  
 Filters: configuration.location1 is not null

<b>LOCATION1</b>	<b>TEXT<sup>1)</sup></b>	<b>Searchcode</b>
------------------	--------------------------	-------------------

<sup>1)</sup> In export, replace '\*' , '?' , '\_' , '%' and ' ' by '-' and add # before codes which start with 0..9.

**Table 57: Mapping Configuration Items locations B**



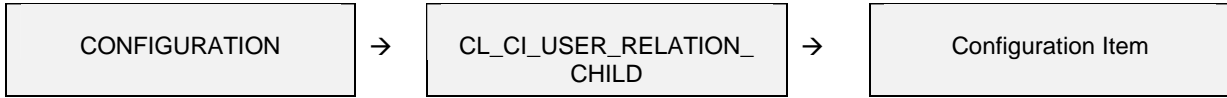
Joined on: -  
 Filters: configuration.location2 is not null



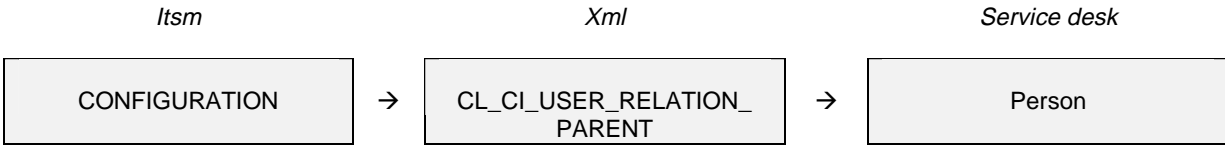


configuration.id  
=  
configuration.id

χ  
child



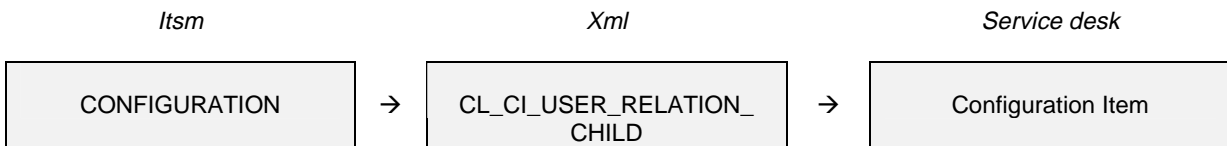
**Table 62: Mapping Configuration Items user relations parent**



Joined on: -  
Filters: configuration.ci\_user is not null  
(configuration.ci\_user\_type=11800001 or configuration.ci\_user\_type=11800003)

if ci_user_type=1180001: 'CON' + CI_USER	<b>PERSON_ID</b>	<b>Source ID</b>
if ci_user_type=1180003: 'EMP' + CI_USER		
else null		

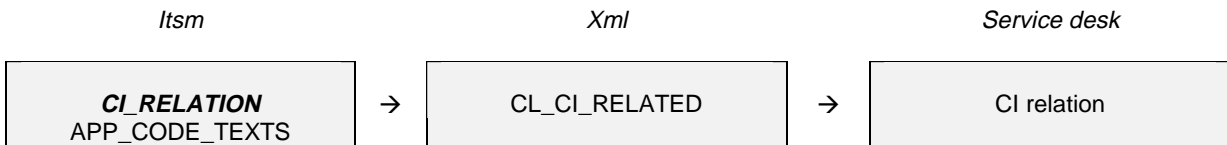
**Table 63: Mapping Configuration Items user relations child**



Joined on: -  
Filters: -

<b>ID</b>	<b>CI_ID Parent</b>	<b>Source ID use to search Users:User</b>
-----------	-------------------------	---

**Table 64: Mapping Configuration Items relations**



Joined on: ci\_relation.relation=app\_code\_texts.cox\_cod\_id(+)  
Filters: (app\_code\_texts.cox\_lng\_id='gb' or app\_code\_texts.cox\_lng\_id is null)

CI_RELATION.CONFIGURATION APP_CODE_TEXTS.COX_TEXT	<b>CI_PARENT RELTYPE_SEARCHTEXT</b>	<b>use to search CI from use to search CI relation type</b>
CI_RELATION.CI	<b>CI_CHILD</b>	<b>use to search CI to</b>

When importing a table, all entities that it refers to should be already imported at that time. Exporting and importing the tables in the following order ensures this:

CL\_CODE\_CI\_MAINCAT  
CL\_CODE\_CI\_CAT  
CL\_CODE\_CI\_SUBCAT  
CL\_CODE\_CI\_STA  
CL\_CODE\_CI\_BRAND  
CL\_CODE\_CI\_LOC1  
CL\_CODE\_CI\_LOC2  
CL\_CI  
CL\_CI\_RELATED  
CL\_CI\_COMPONENT\_CHILD  
CL\_CI\_COMPONENT\_PARENT  
CL\_CI\_USER\_RELATION\_CHILD  
CL\_CI\_USER\_RELATION\_PARENT

## Appendix B - .8 Services

Services themselves are migrated fairly straight forward. The relations with the related CI and the associated CIs are migrated using parent-child relations.

**Table 65: Mapping Services**

<i>Itsm</i>	<i>Xml</i>	<i>Service desk</i>
<b>SERVICE DATA_SET</b> APP_CODE_TEXTS	CL_SERVICE	Service
Joined on: service.srv_auth_file=data_set.id(+) service.srv_cod_id_srv_status=app_code_texts.cox_cod_id(+)		
Filters: (app_code_texts.cox_lng_id='gb' or app_code_texts.cox_lng_id is null)		
<b>if service.srv_name is not null:</b> <b>SERVICE.SRV_NAME1</b> else: <b>SERVICE.SRV_NAME2</b> <b>APP_CODE_TEXT.COX_TXT</b> <b>SERVICE.SRV_ID</b> <b>DATA_SET.DESCRPTION</b> <b>SRV_CNF_ID</b>  <b>SERVICE.SRV_DESCRIPTION</b> <sup>1)</sup> Truncate to 80	<b>NAME</b>   <b>STATUS_SEARCHTEXT</b> <b>SRV_ID</b> <b>POOL_SEARCHTEXT</b> <b>CI_ID</b>  <b>DESCRIPTION <sup>1)</sup></b>	<b>Name</b>   <b>use to search Status</b> <b>Source ID</b> <b>use to search Pool</b> <b>Configuration</b> <b>Item:Configuration Item</b> <b>Description</b>

**Table 66: Mapping Services statuses**

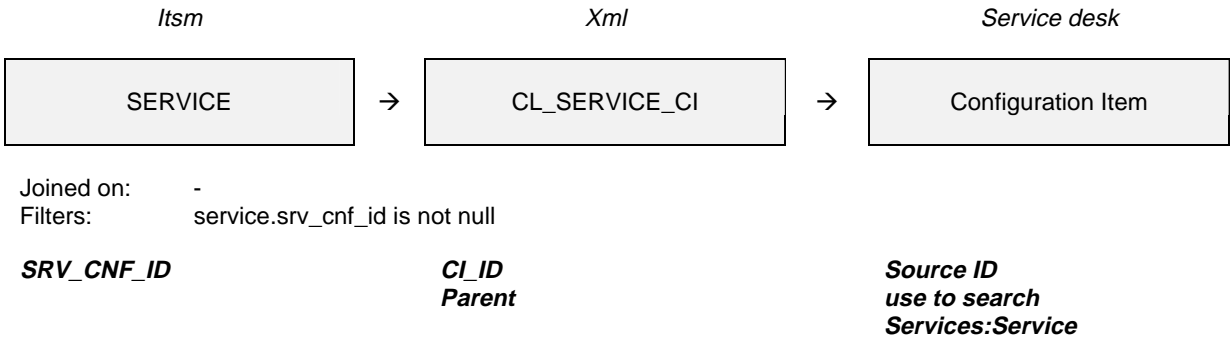
<i>Itsm</i>	<i>Xml</i>	<i>Service desk</i>
<b>APP_CODE_TEXTS</b> APP_CODES	CL_CODE_SER_STA	Service Status
Joined on: app_code_texts.cox_cod_id=app_codes.cod_id		
Filters: cox_cod_id between 15500000 and 15599999 cox_lng_id = 'gb'		
<b>APP_CODE_TEXTS.COX_TEXT</b> <b>APP_CODES.COD_ORDERING</b>	<b>TEXT</b> <b>ORDERING</b>	<b>Text</b> <b>Ordering</b>

**Table 67: Parent-child relation Services configuration item**

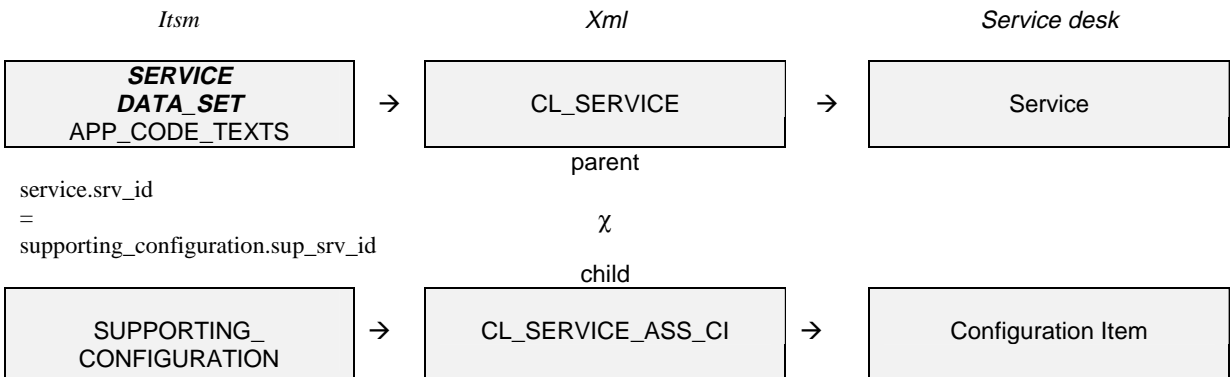
<i>Itsm</i>	<i>Xml</i>	<i>Service desk</i>
<b>SERVICE DATA_SET</b> APP_CODE_TEXTS	CL_SERVICE parent	Service
service.srv_id = service.srv_id	χ child	
SERVICE	CL_SERVICE_CI	Configuration Item



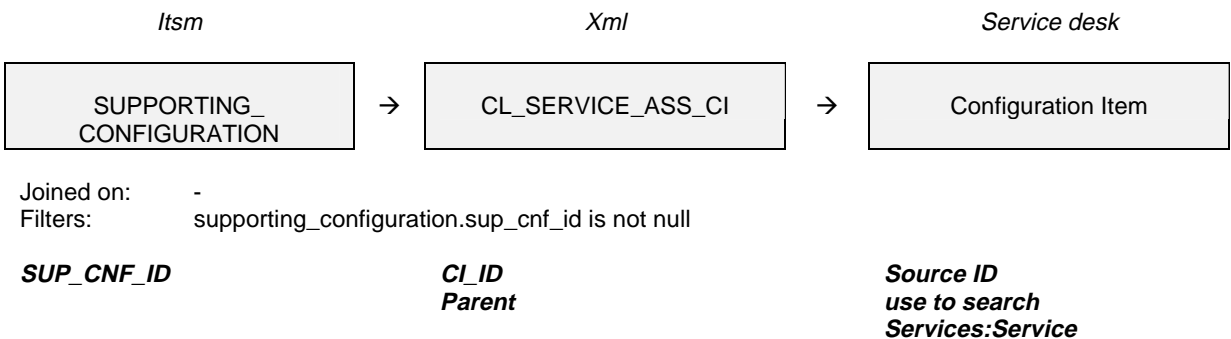
**Table 68: Mapping Services configuration item**



**Table 69: Parent-child relation Services associated CIs**



**Table 70: Mapping Services associated CIs**



When importing a table, all entities that it refers to should be already imported at that time. Exporting and importing the tables in the following order ensures this:

- CL\_CODE\_SER\_STA
- CL\_SERVICE\_CI
- CL\_SERVICE\_ASS\_CI
- CL\_SERVICE

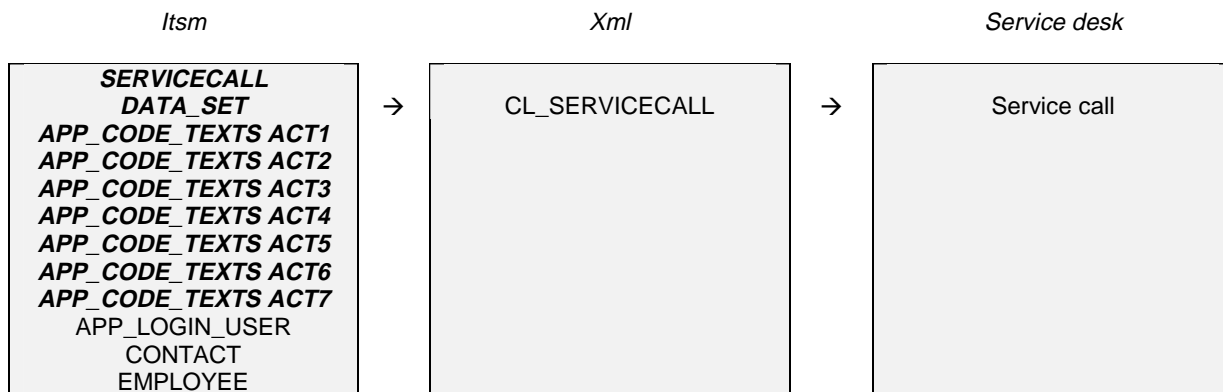
## Appendix B - .9 Service Calls

When migrating service calls attention needs to be given to how the history lines are dealt with. The history lines are split into two classes: one for the lines with action fields that fit into the subject field of the history line in Service Desk, and one for the lines with an action field that doesn't fit. In the last case the action field is truncated and put in the subject as "action >>" and the complete action is put in the information field. Because accounts cannot be used to indicate who created the history line, the name of the person or organization in the action is entered in the following way: "name : action".

Another area to be aware of when importing service calls is what happens to deadlines. These cannot be migrated in the same class as the creation-date. The deadline of a service call should be later than it's creation-date and this can only be successfully migrated using a separate class that updates the calls by giving them their deadlines.

To be able to migrate relations between service calls and other service events, we import two service event relations: "Caused by" and "Related to" so we can refer to them later.

**Table 71: Mapping Service Calls**



Joined on:

- servicecall.auth\_file=data\_set.id(+)
- servicecall.employee=app\_login\_user.lus\_emp\_id(+)
- servicecall.caller=contact.address(+)
- servicecall.caller=employee.id(+)
- servicecall.impact\_id=act1.cox\_cod\_id(+)
- servicecall.priority\_id=act2.cox\_cod\_id(+)
- servicecall.category\_id=act3.cox\_cod\_id(+)
- servicecall.incidentcode\_id=act4.cox\_cod\_id(+)
- servicecall.closurecode\_id=act5.cox\_cod\_id(+)
- servicecall.medium\_id=act6.cox\_cod\_id(+)
- servicecall.status\_id=act7.cox\_cod\_id(+)

Filters:

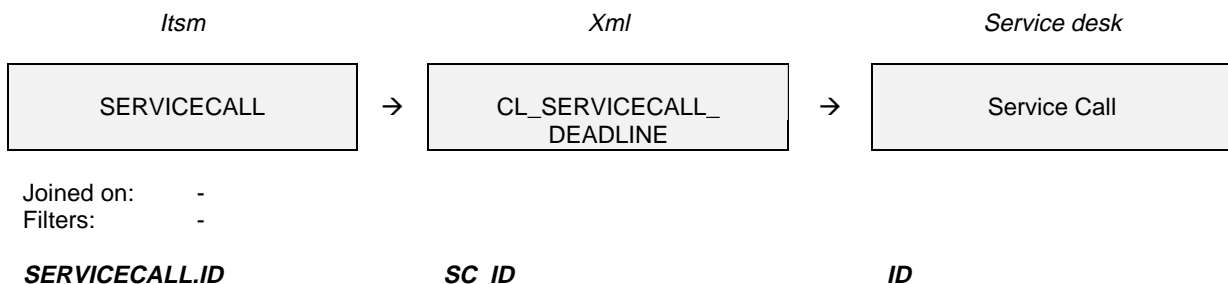
- (act1.cox\_lng\_id='gb' or act1.cox\_lng\_id is null)
- (act2.cox\_lng\_id='gb' or act2.cox\_lng\_id is null)
- (act3.cox\_lng\_id='gb' or act3.cox\_lng\_id is null)
- (act4.cox\_lng\_id='gb' or act4.cox\_lng\_id is null)
- (act5.cox\_lng\_id='gb' or act5.cox\_lng\_id is null)
- (act6.cox\_lng\_id='gb' or act6.cox\_lng\_id is null)
- (act7.cox\_lng\_id='gb' or act7.cox\_lng\_id is null)

<b>DATA_SET.DESCRPTION</b>	<b>POOL_SEARCHTEXT</b>	<b>use to search Pool</b>
<b>SERVICECALL.ID</b>	<b>SC_ID</b>	<b>ID</b>
<b>SERVICECALL.SER_EVENT_ID</b>	<b>SOURCE_ID</b>	<b>Source ID</b>

<b>SERVICECALL.CLOSE_DATETIME</b>	<b>ACT_FINISH</b>	<b>Actual Finish</b>
<b>SERVICECALL.CALL_DATE</b>	<b>ACT_START</b>	<b>Actual Start</b>
<b>SERVICECALL.DESCRPTION</b>	<b>DESCRIPTION</b>	<b>Description</b>
<b>APP_CODE_TEXTS.COX_TEXT</b>	<b>IMPACT_SEARCHTEXT</b>	<b>use to search Impact</b>
<b>SERVICECALL.INFORMATION</b>	<b>INFORMATION</b>	<b>Information</b>
<b>APP_CODE_TEXTS.COX_TEXT</b>	<b>PRIORITY_SEARCHTEXT</b>	<b>use to search Priority</b>
<b>SERVICECALL.CI</b>	<b>CI_ID</b>	<b>use to search Configuration Item</b>
<b>if caller_type = 10200001: 'CON' + SERVICECALL.CALLER</b>	<b>CALLER_ID</b>	<b>use to search Caller</b>
<b>if caller_type = 10200002: 'EMP' + SERVICECALL.CALLER</b>		
<b>else null</b>		
<b>APP_CODE_TEXTS.COX_TEXT</b>	<b>CATEGORY_SEARCHTEXT</b>	<b>use to search Category</b>
<b>APP_CODE_TEXTS.COX_TEXT</b>	<b>CLASSIFICATION_SEARCHTEXT</b>	<b>use to search Classification</b>
<b>APP_CODE_TEXTS.COX_TEXT</b>	<b>CLOSURE_SEARCHTEXT</b>	<b>use to search Closure</b>
<b>APP_CODE_TEXTS.COX_TEXT</b>	<b>MEDIUM_SEARCHTEXT</b>	<b>use to search Medium</b>
<b>if caller_type = 10200001: 'EO' + CONTACT.ORGANIZATION</b>	<b>ORG_ID</b>	<b>use to search Organization</b>
<b>if caller_type = 10200002: 'IO' + EMPLOYEE.ORGANIZATION_UNIT</b>		
<b>if caller_type = 10200003: 'EO' + SERVICECALL.CALLER</b>		
<b>else null</b>		
<b>SERVICECALL.SER_SRV_ID</b>	<b>SERVICE_ID</b>	<b>use to search Service Solution</b>
<b>SERVICECALL.SOLUTION</b>	<b>SOLUTION</b>	
<b>APP_CODE_TEXTS.COX_TEXT</b>	<b>STATUS_SEARCHTEXT</b>	<b>use to search Status</b>
<b>'EMP' + SERVICECALL.SPECIALIST</b>	<b>TO_PERSON_SOURCEID</b>	<b>Use to search Assignment:To person</b>
<b>SERVICECALL.HD_GROUP</b>	<b>TO_GROUP_SOURCEID</b>	<b>Use to search Assignment:To group</b>
<b>SERVICECALL.REF_NUMBER</b>	<b>REF_NUMBER</b>	<b>Assignment:Reference #</b>
<b>SERVICECALL.REMARK</b>	<b>REMARK</b>	<b>Assignment:Infromation from sender</b>
<b>'EO' + SERVICECALL.RETAINED</b>	<b>TO_ORG_SOURCEID</b>	<b>Use to search Assignment:To external Organization</b>
<b>SERVICECALL.CALL_DATE</b>	<b>CREATED</b>	<b>Registration:Created</b>
<b>if app_login_user.lus_login_name is not null: APP_LOGIN_USER.LUS_LOGIN_NAME</b>	<b>CREATEDBY_SEARCHTEXT <sup>1)</sup></b>	<b>Use to search Registration:Created by</b>
<b>else: 'migration'</b>		
<b>SERVICECALL.CALLERNAME2</b>	<b>CONTACT_ORGANIZATION</b>	<b>Contact Organization</b>

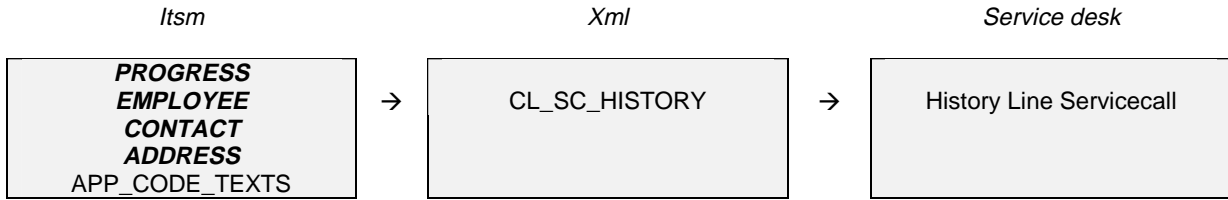
<sup>1)</sup> Truncate to 40

**Table 72: Mapping Service Call Deadlines**



**SERVICECALL.TARGET\_DATE      DEADLINE      Deadline**

**Table 73: Mapping Service Call History Lines**

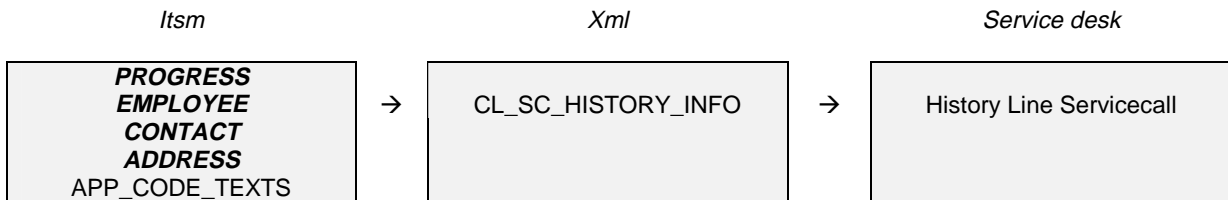


Joined on:      progress.who=employee.id(+)  
                   progress.who=contact.address(+)  
                   progress.who=address.id(+)  
                   progress.who\_type=act.cox\_cod\_id(+)

Filters:        length(concat(concat(concat(concat(concat(act.cox\_searchcode,act.cox\_text),'  
                   '),decode(progress.who\_type, 15000003, substr(concat(concat(employee.first\_name,'  
                   '),employee.name),1,40), 15000001, substr(concat(concat(contact.first\_name,'  
                   '),address.name1),1,40), 15000002, address.name1)),':'),progress.action)) <= 255  
                   (act.cox\_lng\_id='gb' or act.cox\_lng\_id is null)  
                   (progress.service in (select id from servicecall))

<b>ACT.COX_SEARCHCODE +</b>	<b>SUBJECT</b>	<b>Subject</b>
<b>ACT.COX_TEXT +</b>		
if progress.who_type = 15000003:		
<b>EMPLOYEE.FIRST_NAME +</b>		
<b>EMPLOYEE.NAME<sup>1)</sup></b>		
if progress.who_type = 15000001:		
<b>CONTACT.FIRST_NAME +</b>		
<b>ADDRESS.NAME1<sup>1)</sup></b>		
if progress.who_type = 15000002:		
<b>ADDRESS.NAME1</b>		
always:		
+ ':' + ACTION		
<b>PROG_DATE</b>	<b>CREATED</b>	<b>Registration:Created</b>
<b>'migration'</b>	<b>CREATEDBY_SEARCHTEXT</b>	<b>use to search</b>
<b>SERVICE</b>	<b>SC_ID</b>	<b>Registration:Created by</b>
<sup>1)</sup> Truncate to 40		<b>use to search Service call</b>

**Table 74- Mapping Service Call History Lines**



Joined on:      progress.who=employee.id(+)  
                   progress.who=contact.address(+)  
                   progress.who=address.id(+)  
                   progress.who\_type=act.cox\_cod\_id(+)

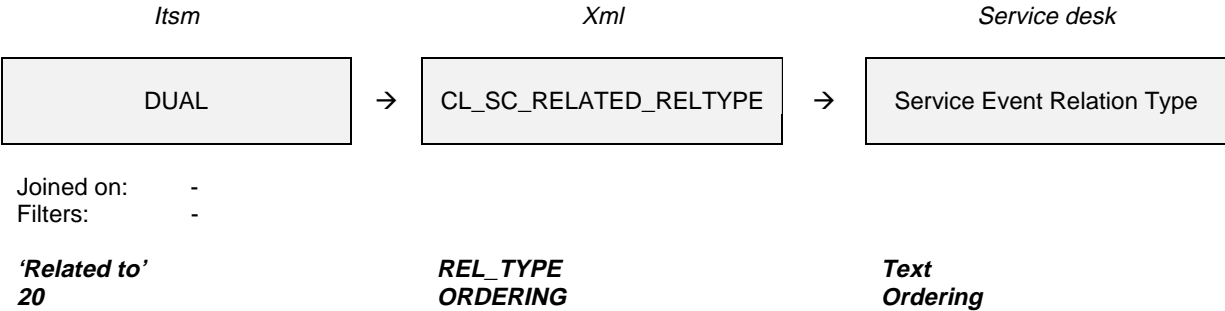
Filters:        length(concat(concat(concat(concat(concat(act.cox\_searchcode,act.cox\_text),'  
                   '),decode(progress.who\_type, 15000003, substr(concat(concat(employee.first\_name,'  
                   '),employee.name),1,40), 15000001, substr(concat(concat(contact.first\_name,'  
                   '),address.name1),1,40), 15000002, address.name1)),':'),progress.action)) > 255  
                   (act.cox\_lng\_id='gb' or act.cox\_lng\_id is null)  
                   (progress.service in (select id from servicecall))







**Table 81: Mapping Service Calls Related to Relation Type**



When importing a table, all entities that it refers to should already be imported at that time. Exporting and importing the tables in the following order ensures this:

- CL\_CODE\_SC\_CAT
- CL\_CODE\_SC\_CLO
- CL\_CODE\_SC\_INC
- CL\_CODE\_SC\_MED
- CL\_CODE\_SC\_STA
- CL\_SERVICECALL
- CL\_SC\_HISTORY
- CL\_SC\_HISTORY\_INFO
- CL\_SC\_CAUSED\_BY\_RELTYPE
- CL\_SC\_RELATED\_RELTYPE

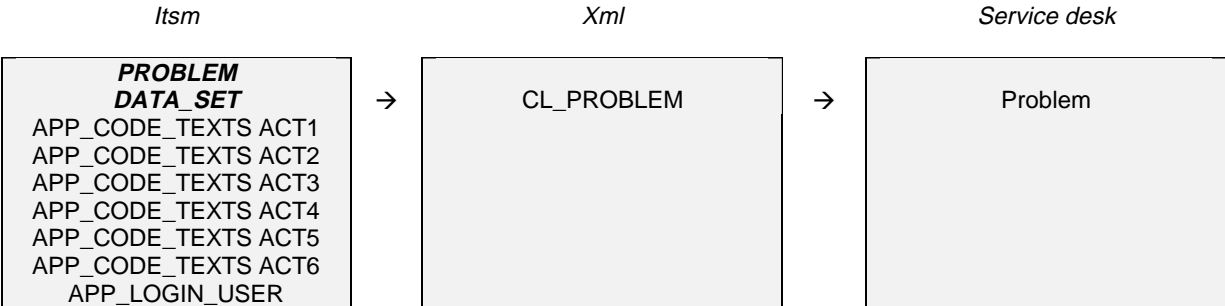
**Appendix B - .10 Problems**

One area to be aware of when importing problems is how the history lines are imported. These are migrated in the same way as described for the history lines of service calls.

Deadlines are also migrated in the same way as described for the deadlines of service calls.

Some special classes were also created to migrate the relations between different service events.

**Table 82: Mapping Problems**



Joined on:     problem.auth\_file=data\_set.id(+)  
                   problem.impact\_id=act1.cox\_cod\_id(+)  
                   problem.priority\_id=act2.cox\_cod\_id(+)  
                   problem.category\_id=act3.cox\_cod\_id(+)  
                   problem.problemcode\_id=act4.cox\_cod\_id(+)  
                   problem.closurecode\_id=act5.cox\_cod\_id(+)  
                   problem.status\_id=act6.cox\_cod\_id(+)  
                   problem.caller=app\_login\_user.lus\_emp\_id(+)

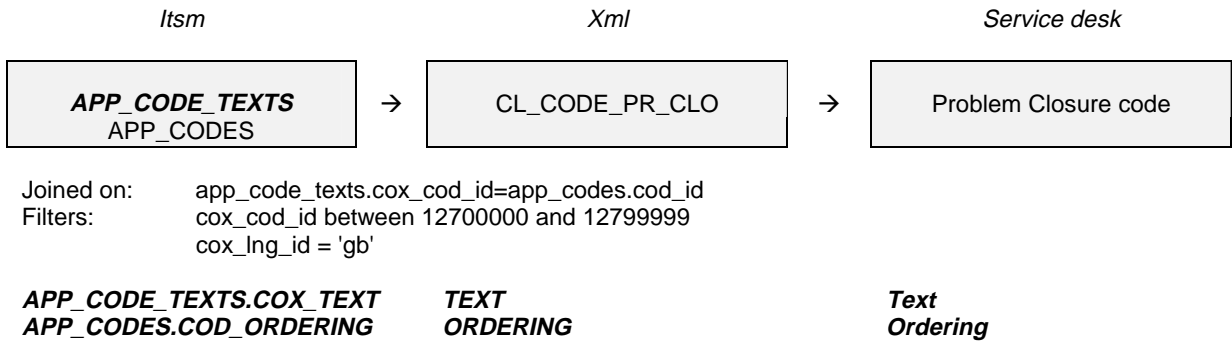
Filters:        (act1.cox\_lng\_id='gb' or act1.cox\_lng\_id is null)  
                   (act2.cox\_lng\_id='gb' or act2.cox\_lng\_id is null)  
                   (act3.cox\_lng\_id='gb' or act3.cox\_lng\_id is null)  
                   (act4.cox\_lng\_id='gb' or act4.cox\_lng\_id is null)  
                   (act5.cox\_lng\_id='gb' or act5.cox\_lng\_id is null)  
                   (act6.cox\_lng\_id='gb' or act6.cox\_lng\_id is null)

<b>DATA_SET.DESCRPTION</b>	<b>POOL_SEARCHTEXT</b>	<b>use to search Pool</b>
<b>PROBLEM.ID</b>	<b>PR_ID</b>	<b>ID</b>
<b>PROBLEM.CLOSE_DATETIME</b>	<b>ACT_FINISH</b>	<b>Actual Finish</b>
<b>PROBLEM.CALL_DATE</b>	<b>ACT_START</b>	<b>Actual Start</b>
<b>PROBLEM.DESCRPTION</b>	<b>DESCRIPTION</b>	<b>Description</b>
<b>APP_CODE_TEXTS.COX_TEXT</b>	<b>IMPACT_SEARCHTEXT</b>	<b>use to search Impact</b>
<b>PROBLEM.INFORMATION</b>	<b>INFORMATION</b>	<b>Information</b>
<b>APP_CODE_TEXTS.COX_TEXT</b>	<b>PRIORITY_SEARCHTEXT</b>	<b>use to search Priority</b>
<b>PROBLEM.CI</b>	<b>CI_ID</b>	<b>use to search Configuration</b>
		<b>Item</b>
<b>APP_CODE_TEXTS.COX_TEXT</b>	<b>CATEGORY_SEARCHTEXT</b>	<b>use to search Category</b>
<b>APP_CODE_TEXTS.COX_TEXT</b>	<b>CLASSIFICATION_SEARCHTEXT</b>	<b>use to search Classification</b>
<b>APP_CODE_TEXTS.COX_TEXT</b>	<b>CLOSURE_SEARCHTEXT</b>	<b>use to search Closure code</b>
<b>APP_CODE_TEXTS.COX_TEXT</b>	<b>STATUS_SEARCHTEXT</b>	<b>use to search Status</b>
<b>PROBLEM.SOLUTION</b>	<b>SOLUTION<sup>1)</sup></b>	<b>Solution</b>
<b>'EMP' + PROBLEM.SPECIALIST</b>	<b>TO_PERSON_SOURCEID</b>	<b>Use to search</b>
		<b>Assignment:To person</b>
<b>PROBLEM.HD_GROUP</b>	<b>TO_GROUP_SOURCEID</b>	<b>Use to search</b>
		<b>Assignment:To group</b>
<b>PROBLEM.REF_NUMBER</b>	<b>REF_NUMBER</b>	<b>Assignment:Reference #</b>
<b>PROBLEM.REMARK</b>	<b>REMARK</b>	<b>Assignment:Infromation</b>
		<b>from sender</b>
<b>'EO' + PROBLEM.RETAINED</b>	<b>TO_ORG_SOURCEID</b>	<b>Use to search</b>
		<b>Assignment:To external</b>
		<b>Organization</b>
<b>PROBLEM.CALL_DATE</b>	<b>CREATED</b>	<b>Registration:Created</b>
<b>if change.caller_type = 10200002</b>	<b>CREATEDBY_SEARCHTEXT<sup>2)</sup></b>	<b>Use to search</b>
<b>and</b>		<b>Registration:Created by</b>
<b>app_login_user.lus_login_name</b>		
<b>is not null:</b>		
<b>APP_LOGIN_USER.LUS_LOGIN_</b>		
<b>NAME</b>		
<b>else:</b>		
<b>'migration'</b>		
<sup>1)</sup> Truncate to 255		
<sup>2)</sup> Truncate to 40		

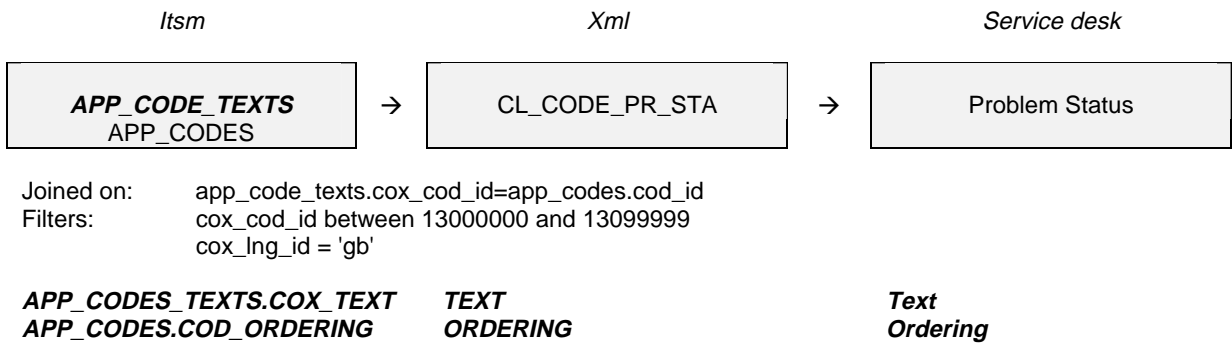




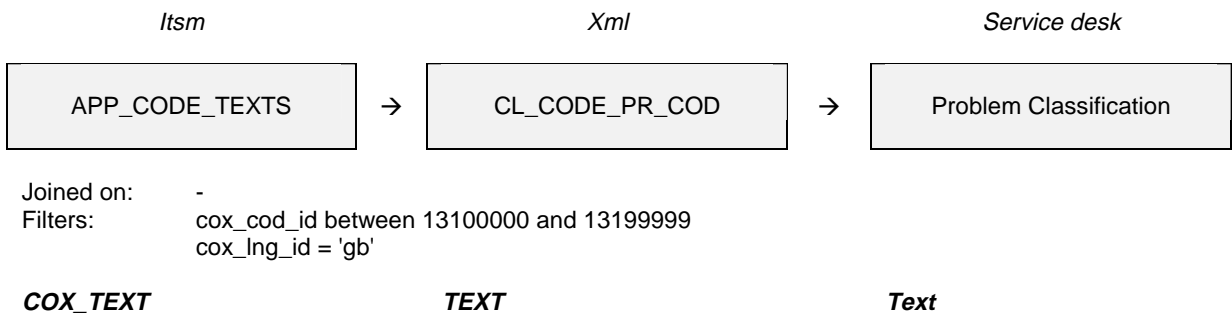
**Table 87: Mapping Problem Closures**



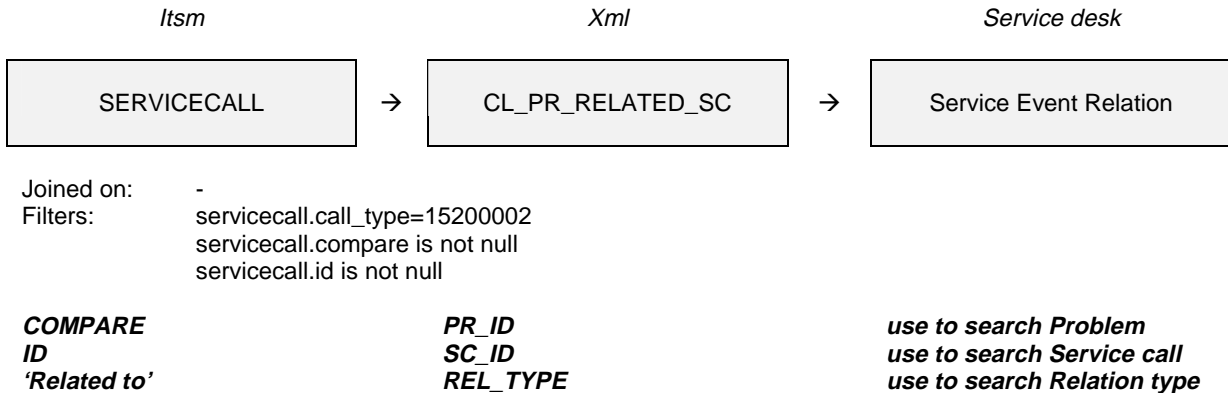
**Table 88: Mapping Problem Statuses**



**Table 89: Mapping Problem Codes**



**Table 90: Mapping Problem Related Service Calls**



When importing a table, all entities that it refers to should already be imported at that time. Exporting and importing the tables in the following order ensures this:

- CL\_CODE\_PR\_CAT
- CL\_CODE\_PR\_CLO
- CL\_CODE\_PR\_STA
- CL\_CODE\_PR\_COD
- CL\_PROBLEM
- CL\_PROBLEM\_HISTORY
- CL\_PROBLEM\_HISTORY\_INFO
- CL\_PR\_RELATED\_SC

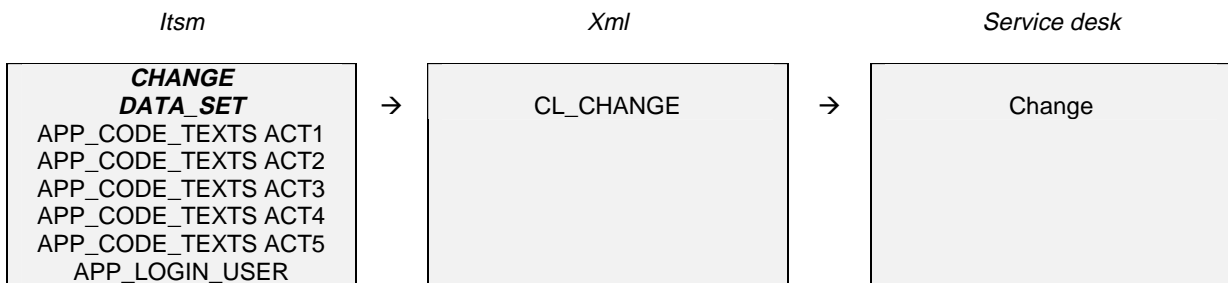
### Appendix B - .11 Changes

One area to be aware of when migrating changes is how the history lines are imported. These are migrated in the same way as we described for the history lines of service calls.

Deadlines are also migrated in the same way as described for the deadlines of service calls.

Some special classes were created to migrate the relations between different service events.

**Table 91: Mapping Changes**





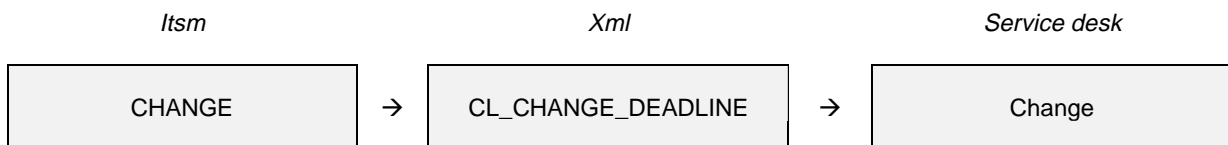
Joined on: change.auth\_file=data\_set.id(+)  
change.priority\_id=act1.cox\_cod\_id(+)  
change.category\_id=act2.cox\_cod\_id(+)  
change.changecode\_id=act3.cox\_cod\_id(+)  
change.closurecode\_id=act4.cox\_cod\_id(+)  
change.status\_id=act5.cox\_cod\_id(+)  
change.caller=app\_login\_user.lus\_emp\_id(+)

Filters: (act1.cox\_lng\_id='gb' or act1.cox\_lng\_id is null)  
(act2.cox\_lng\_id='gb' or act2.cox\_lng\_id is null)  
(act3.cox\_lng\_id='gb' or act3.cox\_lng\_id is null)  
(act4.cox\_lng\_id='gb' or act4.cox\_lng\_id is null)  
(act5.cox\_lng\_id='gb' or act5.cox\_lng\_id is null)

<b>DATA_SET.DESCRPTION</b>	<b>POOL_SEARCHTEXT</b>	<b>use to search Pool</b>
<b>CHANGE.ID</b>	<b>CH_ID</b>	<b>ID</b>
<b>CHANGE.CLOSED_DATE</b>	<b>ACT_FINISH</b>	<b>Actual Finish</b>
<b>CHANGE.CALL_DATE</b>	<b>ACT_START</b>	<b>Actual Start</b>
<b>CHANGE.DESCRPTION</b>	<b>DESCRIPTION</b>	<b>Description</b>
<b>CHANGE.INFORMATION</b>	<b>INFORMATION</b>	<b>Information</b>
<b>APP_CODE_TEXTS.COX_TEXT</b>	<b>PRIORITY_SEARCHTEXT</b>	<b>use to search Priority</b>
<b>CHANGE.CI</b>	<b>CI_ID</b>	<b>use to search Configuration Item</b>
<b>APP_CODE_TEXTS.COX_TEXT</b>	<b>CATEGORY_SEARCHTEXT</b>	<b>use to search Category</b>
<b>APP_CODE_TEXTS.COX_TEXT</b>	<b>CLASSIFICATION_SEARCHTEXT</b>	<b>use to search Classification</b>
<b>APP_CODE_TEXTS.COX_TEXT</b>	<b>CLOSURE_SEARCHTEXT</b>	<b>use to search Closure code</b>
<b>APP_CODE_TEXTS.COX_TEXT</b>	<b>STATUS_SEARCHTEXT</b>	<b>use to search Status</b>
<b>CHANGE.DESIRED</b>	<b>DES_SOLUTION</b>	<b>Desired Solution</b>
<b>'EMP' + CHANGE.SPECIALIST</b>	<b>TO_PERSON_SOURCEID</b>	<b>Use to search Assignment:To person</b>
<b>CHANGE.HD_GROUP</b>	<b>TO_GROUP_SOURCEID</b>	<b>Use to search Assignment:To group</b>
<b>CHANGE.REF_NUMBER</b>	<b>REF_NUMBER</b>	<b>Assignment:Reference #</b>
<b>CHANGE.REMARK</b>	<b>REMARK</b>	<b>Assignment:Infromation from sender</b>
<b>'EO' + CHANGE.RETAINED</b>	<b>TO_ORG_SOURCEID</b>	<b>Use to search Assignment:To external Organization</b>
<b>CHANGE.CALL_DATE</b>	<b>CREATED</b>	<b>Registration:Created</b>
if change.caller_type = 1020002 and app_login_user.lus_login_name is not null: <b>APP_LOGIN_USER.LUS_LOGIN_NAME</b>	<b>CREATEDBY_SEARCHTEXT <sup>1)</sup></b>	<b>Use to search Registration:Created by</b>
else: <b>'migration'</b>		

<sup>1)</sup> Truncate to 40

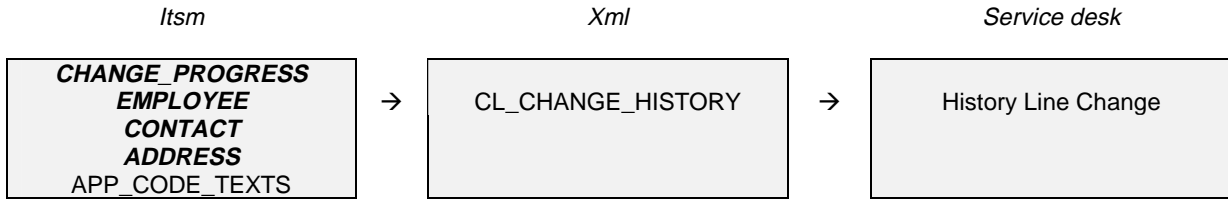
**Table 92: Mapping Change Deadlines**



Joined on: -  
Filters: -

<b>CHANGE.ID</b>	<b>CH_ID</b>	<b>ID</b>
<b>CHANGE.TARGET_DATE</b>	<b>DEADLINE</b>	<b>Deadline</b>

**Table 93: Mapping Change History Lines**



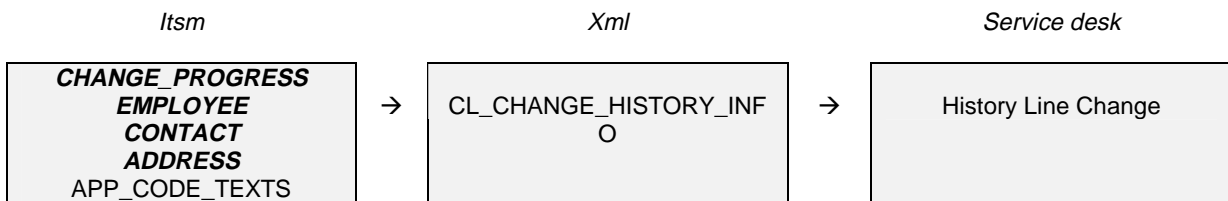
Joined on: change\_progress.who=employee.id(+)  
 change\_progress.who=contact.address(+)  
 change\_progress.who=address.id(+)  
 change\_progress.who\_type=act.cox\_cod\_id(+)

Filters: length(concat(concat(concat(concat(act.cox\_searchcode,act.cox\_text),'  
 '),decode(change\_progress.who\_type, 15000003, substr(concat(concat(employee.first\_name,'  
 '),employee.name),1,40), 15000001, substr(concat(concat(contact.first\_name,'  
 '),address.name1),1,40), 15000002, address.name1)),':'),change\_progress.action)) <= 255  
 (act.cox\_lng\_id='gb' or act.cox\_lng\_id is null)  
 (change\_progress.change in (select id from change))

<b>ACT.COX_SEARCHCODE +</b>	<b>SUBJECT</b>	<b>Subject</b>
<b>ACT.COX_TEXT +</b>		
<b>if change_progress.who_type =</b>		
<b>15000003:</b>		
<b>EMPLOYEE.FIRST_NAME +</b>		
<b>EMPLOYEE.NAME<sup>1)</sup></b>		
<b>if change_progress.who_type =</b>		
<b>15000001:</b>		
<b>CONTACT.FIRST_NAME +</b>		
<b>ADDRESS.NAME1<sup>1)</sup></b>		
<b>if change_progress.who_type =</b>		
<b>15000002:</b>		
<b>ADDRESS.NAME1</b>		
<b>always:</b>		
<b>+ ':' + ACTION</b>		
<b>PROG_DATE</b>	<b>CREATED</b>	<b>Registration:Created</b>
<b>'migration'</b>	<b>CREATEDBY_SEARCHTEXT</b>	<b>use to search</b>
		<b>Registration:Created by</b>
<b>CHANGE</b>	<b>CH_ID</b>	<b>use to search Change</b>

<sup>1)</sup> Truncate to 40

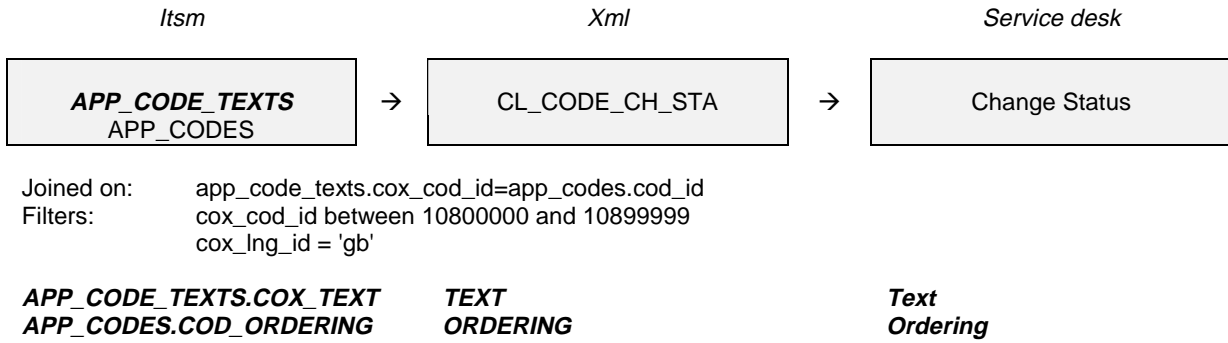
**Table 94: Mapping Change Long History Lines**



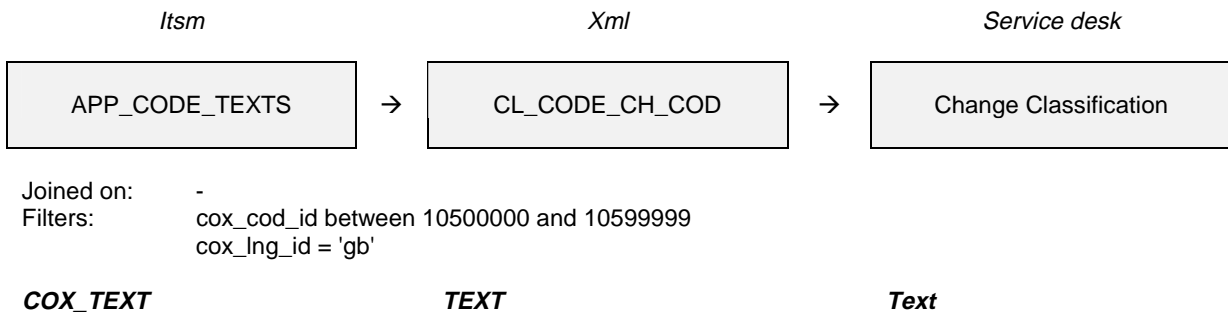
Joined on: change\_progress.who=employee.id(+)  
 change\_progress.who=contact.address(+)  
 change\_progress.who=address.id(+)  
 change\_progress.who\_type=act.cox\_cod\_id(+)



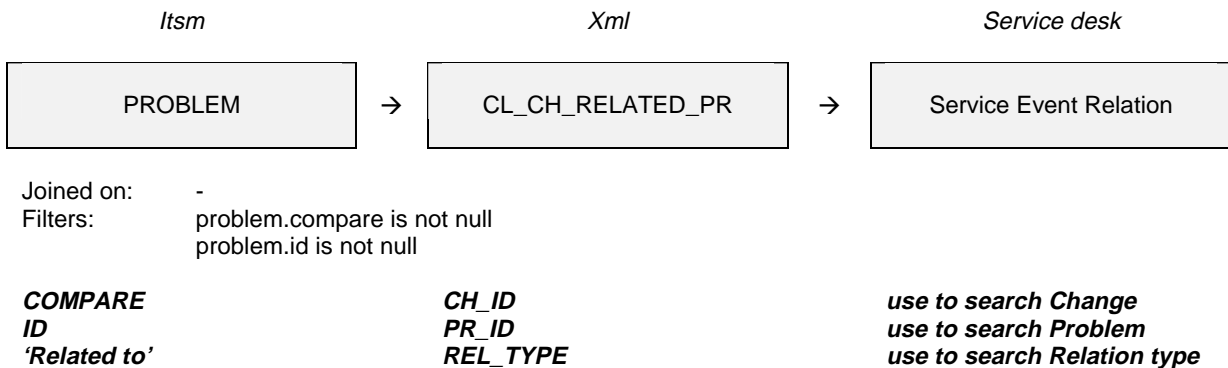
**Table 97: Mapping Change statuses**



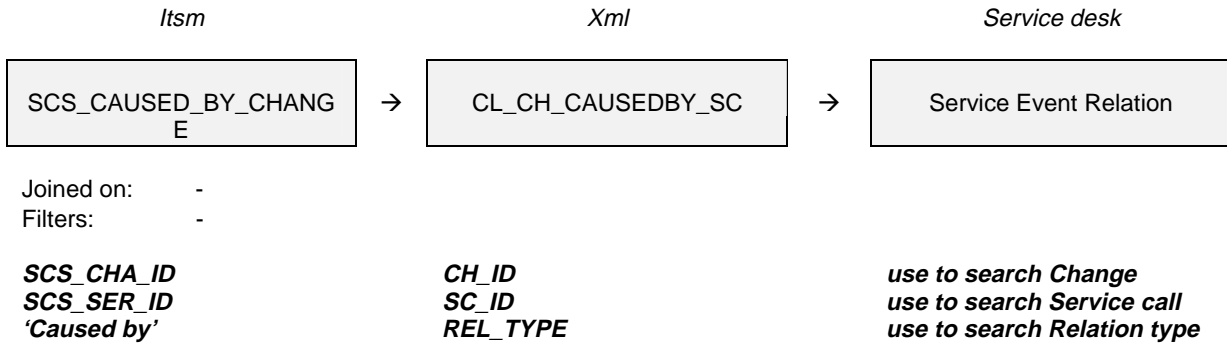
**Table 98: Mapping Change Codes**



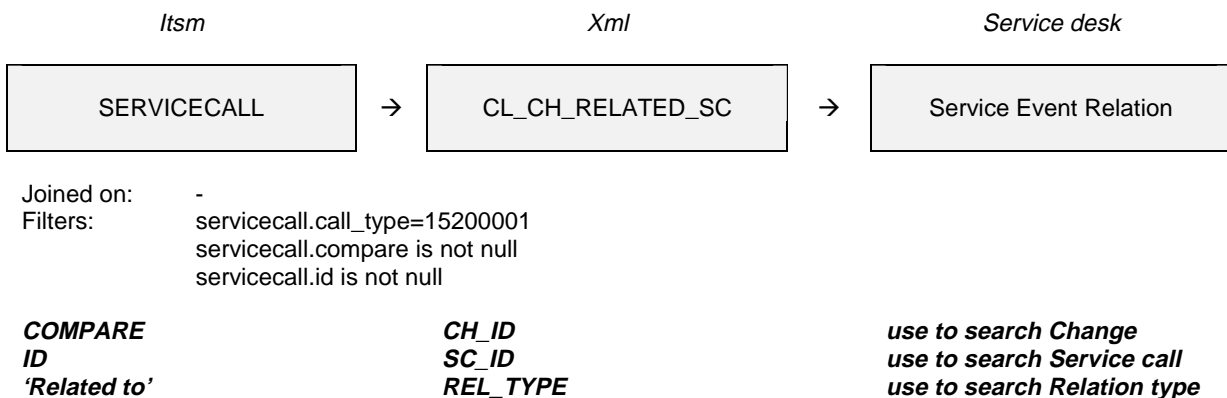
**Table 99: Mapping Change Related Problems**



**Table 100: Mapping Change Caused by Service Call**



**Table 101: Mapping Change Related Service Calls**



When importing a table, all entities that it refers to should already be imported at that time. Exporting and importing the tables in the following order ensures this:

- CL\_CODE\_CH\_CAT
- CL\_CODE\_CH\_CLO
- CL\_CODE\_CH\_STA
- CL\_CODE\_CH\_COD
- CL\_CHANGE
- CL\_CHANGE\_HISTORY
- CL\_CHANGE\_HISTORY\_INFO
- CL\_CH\_RELATED\_PR
- CL\_CH\_CAUSEDBY\_SC
- CL\_CH\_RELATED\_SC

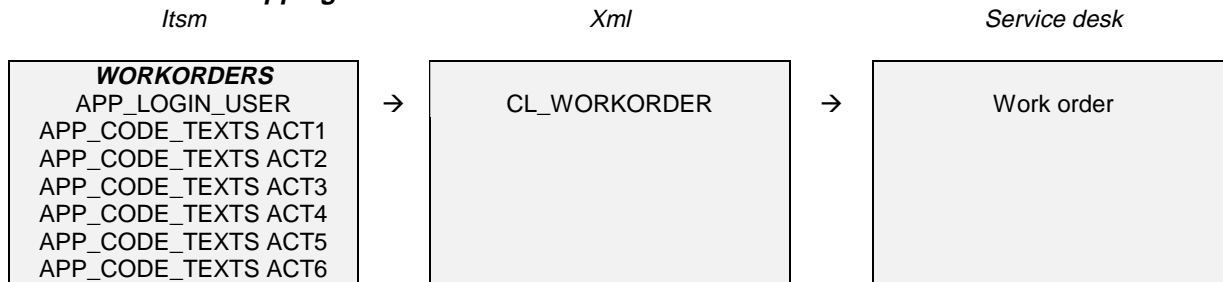
## Appendix B - .12 Work Orders

History lines are migrated in the same way as described for the history lines of service calls.

Deadlines are also migrated in the same way as described for the deadlines of service calls.

Some special classes were created to migrate the relations between different service events.

**Table 102: Mapping Work Orders**



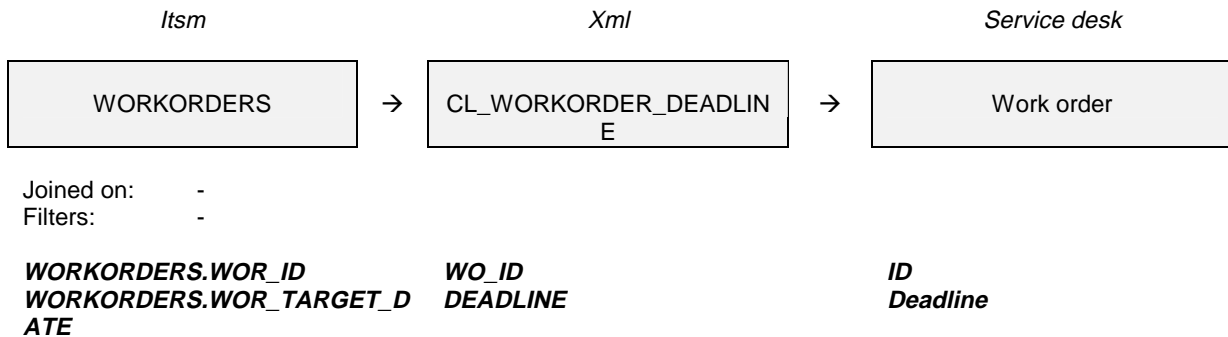
Joined on: workorders.wor\_auth\_file=data\_set.id(+)  
 workorders.wor\_employee=app\_login\_user.lus\_emp\_id(+)  
 workorders.wor\_status\_id=act1.cox\_cod\_id(+)  
 workorders.wor\_priority\_id=act2.cox\_cod\_id(+)  
 workorders.wor\_impact\_id=act3.cox\_cod\_id(+)  
 workorders.wor\_closurecode\_id=act4.cox\_cod\_id(+)  
 workorders.wor\_context\_type=act5.cox\_cod\_id(+)  
 workorders.wor\_classification\_id=act6.cox\_cod\_id(+)

Filters:  
 (act1.cox\_lng\_id='gb' or act1.cox\_lng\_id is null)  
 (act2.cox\_lng\_id='gb' or act2.cox\_lng\_id is null)  
 (act3.cox\_lng\_id='gb' or act3.cox\_lng\_id is null)  
 (act4.cox\_lng\_id='gb' or act4.cox\_lng\_id is null)  
 (act5.cox\_lng\_id='gb' or act5.cox\_lng\_id is null)  
 (act6.cox\_lng\_id='gb' or act6.cox\_lng\_id is null)

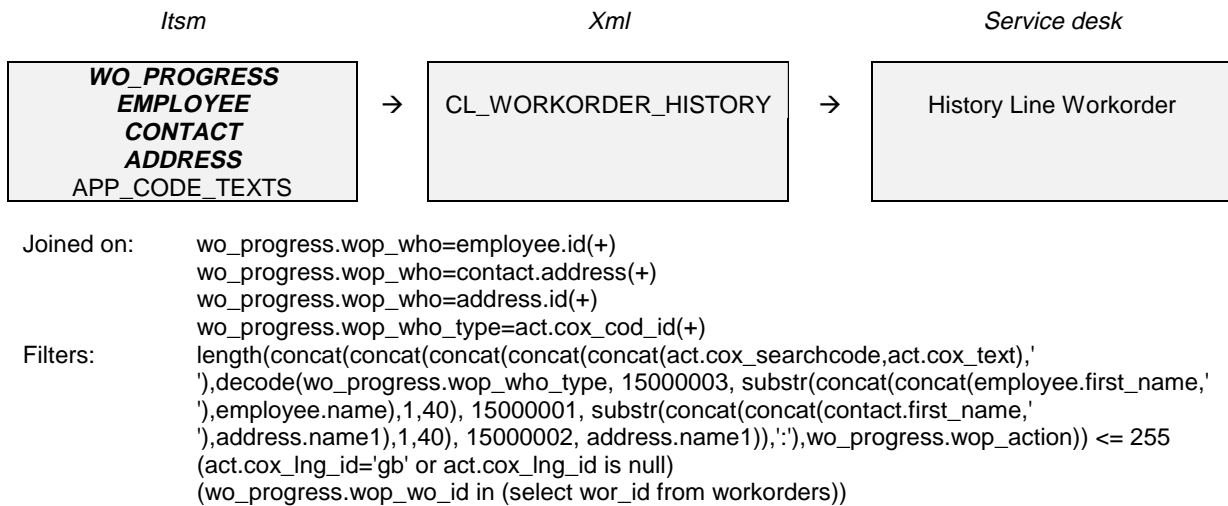
<b>WORKORDERS.WOR_SPENT_TIME</b>	<b>ACT_DURATION</b>	<b>Actual Duration</b>
<b>APP_CODE_TEXTS.COX_TEXT</b>	<b>TEXT</b>	<b>Text</b>
<b>APP_CODES.COD_ORDERING</b>	<b>ORDERING</b>	<b>Ordering</b>
<b>APP_CODE_TEXTS.COX_TEXT</b>	<b>TEXT</b>	<b>Text</b>
<b>APP_CODES.COD_ORDERING</b>	<b>ORDERING</b>	<b>Ordering</b>
<b>COX_TEXT</b>	<b>TEXT</b>	<b>Text</b>
<b>DATA_SET.DESCRPTION</b>	<b>POOL_SEARCHTEXT</b>	<b>use to search Pool</b>
<b>WORKORDERS.WOR_ID</b>	<b>WO_ID</b>	<b>ID</b>
<b>WORKORDERS.WOR_CLOSED_DATE</b>	<b>ACT_FINISH</b>	<b>Actual Finish</b>
<b>WORKORDERS.WOR_START_DATE</b>	<b>ACT_START</b>	<b>Actual Start</b>
<b>WORKORDERS.WOR_DESCRIPTION</b>	<b>DESCRIPTION</b>	<b>Description</b>
<b>APP_CODE_TEXTS.COX_TEXT</b>	<b>IMPACT_SEARCHTEXT</b>	<b>use to search Impact</b>
<b>WORKORDERS.WOR_REMARKS</b>	<b>INFORMATION</b>	<b>Information</b>
<b>WORKORDERS.WOR_START_DATE</b>	<b>PLAN_START</b>	<b>Planned Start</b>
<b>APP_CODE_TEXTS.COX_TEXT</b>	<b>PRIORITY_SEARCHTEXT</b>	<b>use to search Priority</b>
<b>APP_CODE_TEXTS.COX_TEXT</b>	<b>CLOSURE_SEARCHTEXT</b>	<b>use to search Closure code</b>
<b>APP_CODE_TEXTS.COX_TEXT</b>	<b>STATUS_SEARCHTEXT</b>	<b>use to search Status</b>
<b>APP_CODE_TEXTS.COX_TEXT</b>	<b>CATEGORY</b>	<b>use to search Category</b>
<b>'EMP' + WORKORDERS.WOR_SPECIALIST</b>	<b>TO_PERSON_SOURCEID</b>	<b>Use to search Assignment:To person</b>
<b>WORKORDERS.WOR_HD_GROUP</b>	<b>TO_GROUP_SOURCEID</b>	<b>Use to search Assignment:To group</b>
<b>WORKORDERS.WOR_REF_NUMBER</b>	<b>REF_NUMBER</b>	<b>Assignment:Reference #</b>
<b>WORKORDERS.WOR_DISPATCH_REMARKS</b>	<b>REMARK</b>	<b>Assignment:Information from sender</b>
<b>'EO' + WORKORDERS.WOR_CONTRACT_OUT_ORG</b>	<b>TO_ORG_SOURCEID</b>	<b>Use to search Assignment:To external Organization</b>

<b>' CON' + WORKORDERS.WOR_CONTRAC T_OUT_CON WORKORDERS.WOR_START_DA TE</b>	<b>TO_EXT_PERSON_SOURCEID  CREATED</b>	<b>Use to search Assignment:To external Person Registration:Created</b>
<b>if app_login_user.lus_login_name is not null: APP_LOGIN_USER.LUS_LOGIN_ NAME else: 'migration'</b>	<b>CREATEDBY_SEARCHTEXT <sup>1)</sup></b>	<b>Use to search Registration:Created by</b>
<sup>1)</sup> Truncate to 40		

**Table 103: Mapping Work Orders Deadlines**



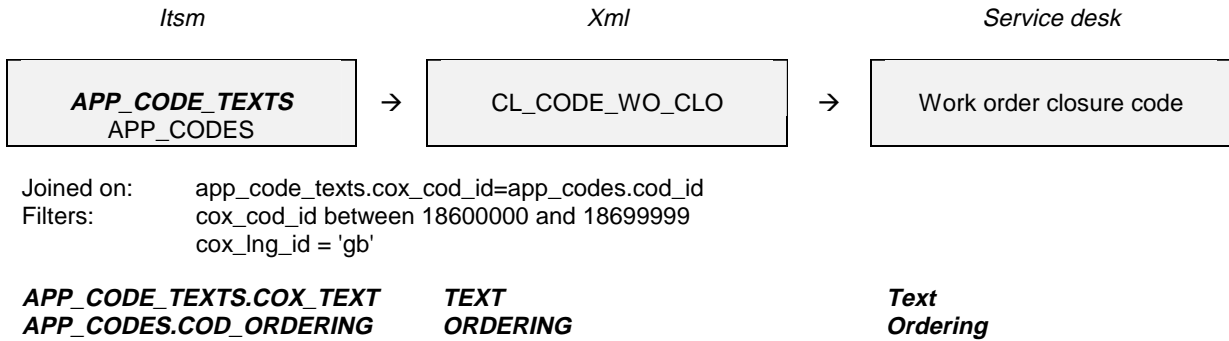
**Table 104: Mapping Work Order History Lines**



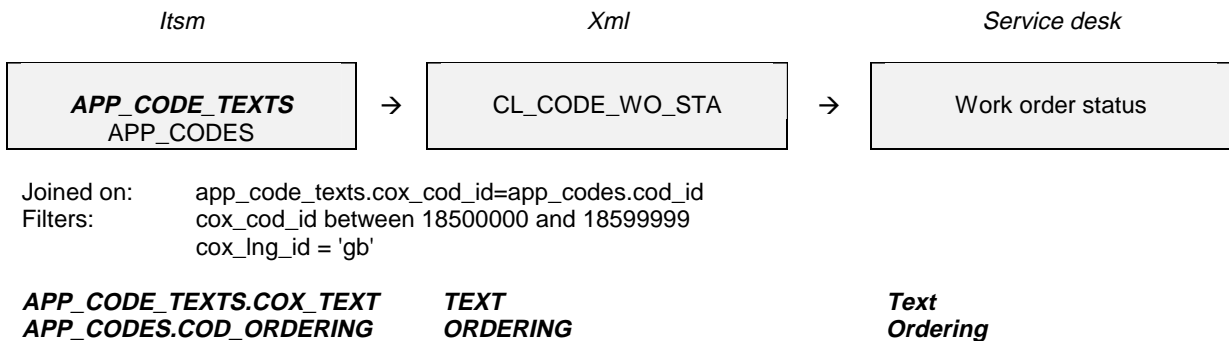




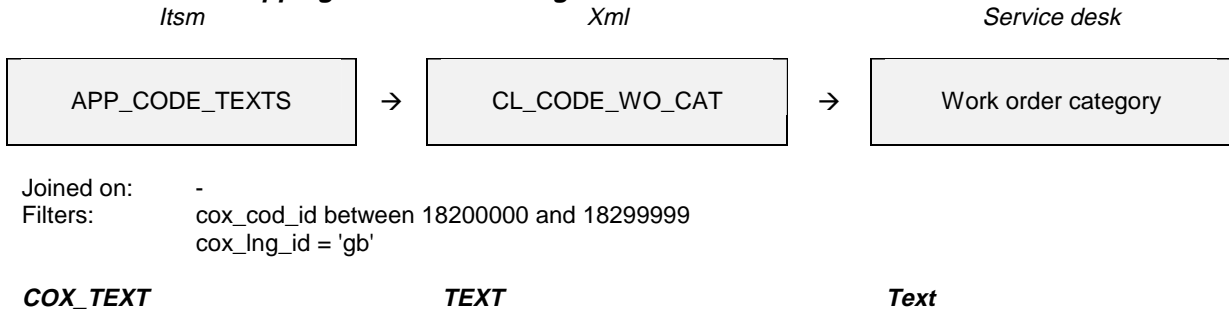
**Table 106: Mapping Work Order Closures**



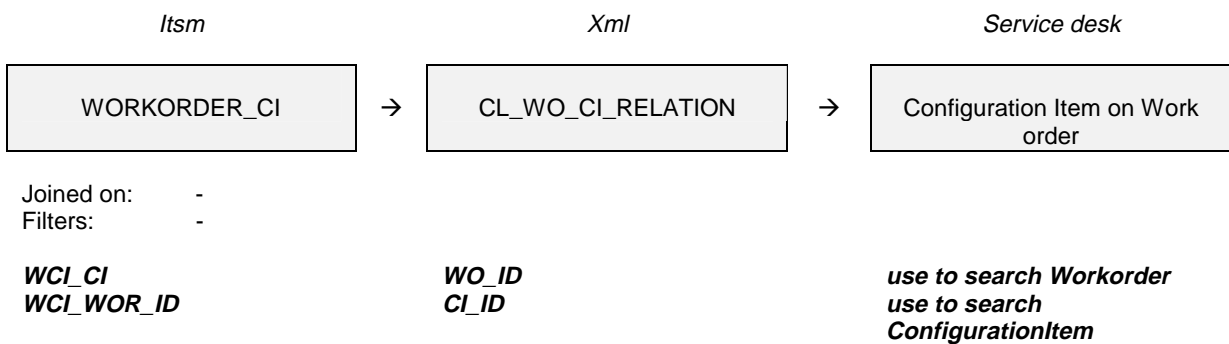
**Table 107: Mapping Work Order Statuses**



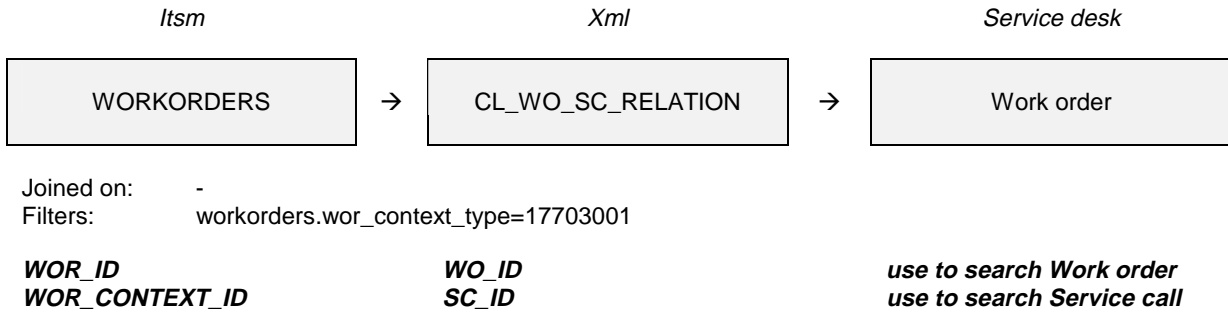
**Table 108: Mapping Work Order Categories**



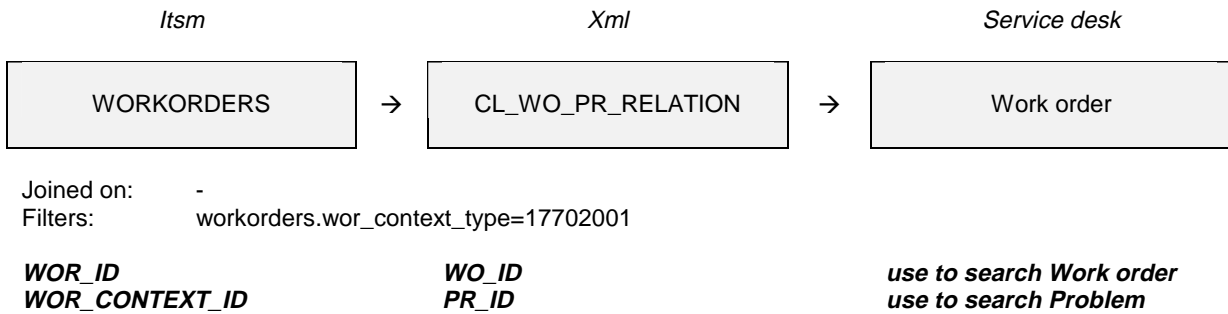
**Table 109: Mapping Work Order Related CIs**



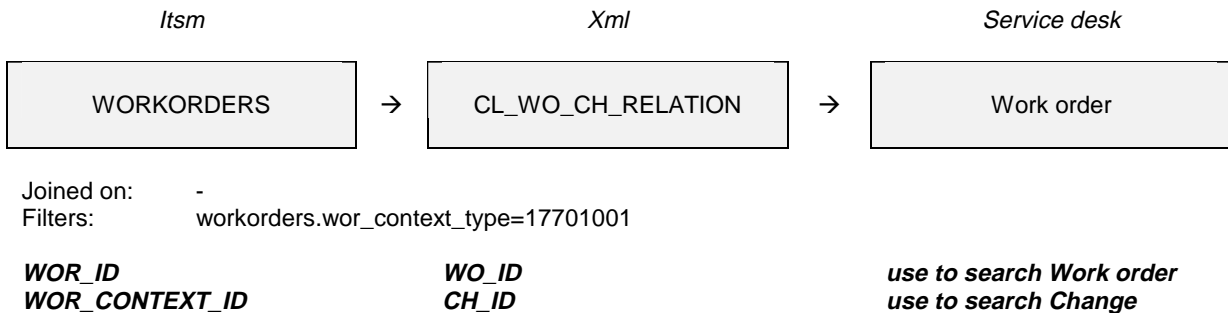
**Table 110: Mapping Work Order Related Service Calls**



**Table 111: Mapping Work Order Related Problems**



**Table 112: Mapping Work Order Related Changes**



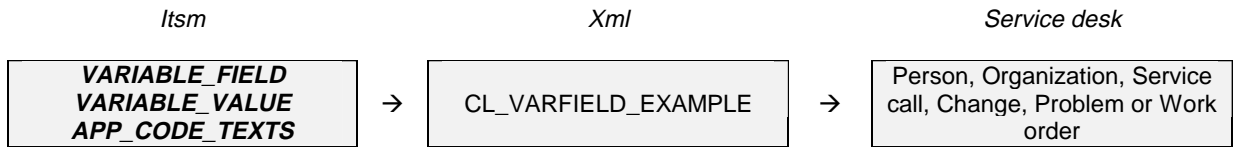
When importing a table, all entities that it refers to should already be imported at that time. Exporting and importing the tables in the following order ensures this:

- CL\_CODE\_WO\_CLO
- CL\_CODE\_WO\_STA
- CL\_CODE\_WO\_CAT
- CL\_WORKORDER
- CL\_WORKORDER\_HISTORY
- CL\_WORKORDER\_HISTORY\_INFO
- CL\_WO\_CI\_RELATION
- CL\_WO\_SC\_RELATION
- CL\_WO\_PR\_RELATION
- CL\_WO\_CH\_RELATION

## Appendix B - .13 Variable Fields Example

It is not possible to migrate variable fields automatically. An example of how it can be done is provided. The explanation can be found earlier in this Migration Guide.

**Table 113: Mapping Variable Fields**



Joined on:     variable\_field.category=variable\_value.category  
                   variable\_field.id=variable\_value.column\_number  
                   variable\_field.category=app\_code\_texts.cox\_cod\_id

Filters:        (app\_code\_texts.cox\_lng\_id='gb' or app\_code\_texts.cox\_lng\_id is null)  
                   app\_code\_texts.cox\_searchcode='category'  
                   variable\_field.column\_name='field name'

**VARIABLE\_VALUE.ID1**

**SOURCE\_ID**

**use to search Person, Organization, Service call, Change, Problem or Work order**

VARIABLE\_VALUE.COLUMN\_VALUE     FIELD\_VALUE

map to appropriate custom field