HP Service Manager

Software Version: 9.41

For the supported Windows® and UNIX® operating systems

Database Management help topics for printing

Document Release Date: September 2015
Software Release Date: September 2015

Legal Notices

Warranty

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

The information contained herein is subject to change without notice.

Restricted Rights Legend

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

Copyright Notice

Trademark Notices

Adobe® is a trademark of Adobe Systems Incorporated.

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

Oracle and Java are registered trademarks of Oracle and/or its affiliates.

UNIX® is a registered trademark of The Open Group.

Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

For a complete list of open source and third party acknowledgements, visit the HP Software Support Online web site and search for the product manual called HP Service Manager Open Source and Third Party License Agreements.

Documentation Updates

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

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

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

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

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

Support

Visit the HP Software Support site at: https://softwaresupport.hp.com.

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

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

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

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

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

HPSW Solutions Catalog accesses the HPSW Integrations and Solutions Catalog portal website. This site enables you to explore HP Product Solutions to meet your business needs, includes a full list of Integrations between HP Products, as well as a listing of ITIL Processes. The URL for this website is https://softwaresupport.hp.com/group/softwaresupport/search-result/-/facetsearch/document/KM01702710.

About this PDF Version of Online Help

This document is a PDF version of the online help. This PDF file is provided so you can easily print multiple topics from the help information or read the online help in PDF format. Because this content was originally created to be viewed as online help in a web browser, some topics may not be formatted properly. Some interactive topics may not

be present in this PDF version. Those topics can be successfully printed from within the online help.

Contents

Database Administration 14

Supported RDBMS 15

Microsoft SQL Server database Unicode support 16

Case-Sensitivity 18

- Load/unload implications of case-sensitivity 18
- JavaScript/RAD implications of a case-insensitive database environment 19
- MS SQL Server collation support 20
- Oracle case-sensitivity 21
  - Requirements for using a case-insensitive Oracle database 21
  - Validate case-insensitive unique indexes 22

Backups 24

Purging out-of-box data 25

Attachments 26

- Support for blocking attachments with certain file extensions 26
  - Default forbidden list of attachment file extensions 29
  - Customize the forbidden list of attachment file extensions 33
  - Add .unsafe to the list of file extensions to skip for knowledgebase indexing 34
- View the attachment history 35

System Definition 36

- System Definition utility 37
- Access the System Definition utility 37
- Table tab 37
  - Table definition 38
  - Table management 38
- Editing System Definition nodes 39
- Access field help from the System Navigator 39
- Forms, triggers, and links 40

Update a form ... 40
Triggers ... 41
Create a Trigger record ... 42
Edit a trigger ... 43
Modify an existing link ... 44

Data persistence ... 45

Database dictionary ... 45
Database dictionary records ... 46
Making database dictionary changes ... 48
Database dictionary data types ... 50
Changing data types ... 51
SQL data type for localizable fields ... 51
Change data types: Character to number ... 52
Database dictionary data types ... 54
Arrays ... 55
Arrayed structures ... 55
Structures ... 55
Fields within structures ... 56
Organization of arrays of structures in the database dictionary ... 56
Example: Arrayed structure ... 56
Create the unique and attribute subtables ... 59
Name the unique and attribute files ... 61
View the unique and attribute subtables ... 62
Splitting SQL tables ... 62
Alias fields ... 63
Access the Database Dictionary utility ... 63
Enable DDL logging of database dictionary changes ... 64
Open a database dictionary ... 64
Copying a database dictionary ... 66
Copy a database dictionary and data records ... 66
Copy a database dictionary only ... 67
Methods for adding database dictionary records ... 69
Field name requirements ... 71
Create an active database dictionary record ... 73
Add a structure to the database dictionary ... 74
Add an alias field ... 76
Add an array ... 77
Add fields to a structure ... 79
Create a database dictionary record by importing RDBMS columns ... 80

Designate unique fields in the array of structures 81
Create an inactive database dictionary record 82
Activate a database dictionary record 83
Methods for deleting database dictionary records 83
Delete a database dictionary record 87
Remove a field from an existing database dictionary record 88
Methods for updating database dictionary records 89
Automatic update of database dictionary records 92
Automatic update of RDBMS data when database dictionary records change 93
Activate a database dictionary record 95
Add a field to an existing database dictionary record 96
Import RDBMS columns into an existing database dictionary record 97
Modify field types: Character (scalar) to array 98
Modify field types: Single to concatenated 99
Regen a database dictionary 100
Search for a field in the database dictionary 102
Update an existing database dictionary record 104
Update the data type of a database dictionary field 105
Renaming a database dictionary 106
Rename a database dictionary 106
Resetting a database table 107
Resetting the database and all records 108
Reset a database table from Database Manager 108
Schedule a reset 108
Keys and indexes in HP Service Manager 109
Key type definitions 110
Key type attributes 111
Tuning: Designing keys for queries 111
Tuning: Key selection algorithms 112
Working with primary keys 112
Usage 113
Retrieve primary key values 115
Retrieve primary key mode 116
Adding a key 117
Add a key as the first key 117
Add keys to bottom of the key list 119
Insert a key between other keys 121
Deleting a key 122
Delete a key 123

Modifying a key 124
Modify keys: Nulls and duplicates to unique 125
Modify keys: Convert the first unique key to primary key 126
Modify keys: Add Not Null constraints to the first unique key in a table 127
SQL cache 129
Regenerating database keys 129
Regenerate database keys 129
Schedule a regeneration 131
Data maps 132
Alternatives to data maps 133
Types of data maps 133
Add a data map record 135
Create data maps to related records 137
Delete a data map record 138
Update a data map record 138
View a data map record 140
Exporting records 140
Export descriptor records 141
Unload error messages 142
Unload files 143
Add an export descriptor record 144
Delete an export descriptor record 146
Export a record list to a text file 147
Export a record list to an Excel file 148
Export a record list using an export description record 149
Export an individual record using an export descriptor record 151
Export selected records from a table using an export descriptor record 153
Schedule the exporting of records using an export descriptor record 155
Schedule the unloading of records 156
Unload a database dictionary only 157
Unload a record list 159
Unload an individual record 162
Unload records from multiple files 164
Unload selected records from a file 165
Update an export descriptor record 168
Importing records 170
Import descriptor records 171
Load error messages and notifications 172
Unload files 173

Add an import descriptor record ... 174
Import a character-delimited text file ... 184
Import a fixed-width text file ... 188
Load an unload file ... 189
Schedule the importing of records using an import descriptor record ... 190
Schedule the loading of an unload file ... 191
View the contents of an unload file ... 193
Unload script utility ... 193
Add an unload script record ... 194
Delete an unload script record ... 195
Protect forms from the unload script utility ... 196
Purge records with the unload script utility ... 196
Run an unload script ... 197
Update an unload script record ... 198
Validate an unload script record ... 199
Purging and archiving records ... 200
Unload files ... 200
Using data maps to purge and archive records ... 201
Archive records ... 202
Purge and archive records ... 204
Purge records ... 207
Schedule purge and archive processes ... 208
Master data records ... 210
Add a master data record ... 212
Update a master data record ... 212
Delete a master data record ... 213
Database record auditing ... 214
Invoke auditing ... 215
Invoke auditing from Format Control ... 215
Invoke auditing from the file in Database Manager (Change Management only) ... 216
Start database record auditing ... 217
Test audit lookup functionality ... 218
Audit processing ... 220
Determine if a file has audit processing ... 220
Audit log file ... 221
Open the audit log file ... 221
Verify the field name ... 222
Verify the file name ... 222
Add an audit specifications record ... 223

Open the audit specifications table ... 223
Define an audit specifications entry ... 224
Audit specifications file ... 225
Audit specifications file description ... 226
The Subroutines form ... 227
IR Expert ... 228
What is Knowledge Engineering? ... 228
Standard record lists and IR Expert ... 229
Database dictionary and IR Expert ... 229
IR Query features ... 230
Using IR Expert to create a query ... 230
Searching the central Knowledge Base ... 231
Promoting or discarding a solution candidate ... 231
IR Expert tasks ... 231
Access IR Expert ... 231
Enable IR search for a file ... 233
Regenerate IR keys ... 234
Start IR Asynchronous mode ... 236
Load data files with IR Expert keys ... 236
Copy information from a query to a source record ... 237
Create an IR Expert query ... 237
Enable editing of Knowledge learning records ... 238
Enable learning for Knowledge records ... 238
Mark an incident record as a solution candidate ... 238
Modify an IR Expert query ... 239
Promote or delete records in the protocore ... 240
Search the central Knowledge Base ... 241
Special considerations for using IR Expert ... 242
How IR Expert evaluates documents for relevance ... 242
Operations on terms ... 242
Lexical analysis ... 243
Stemming ... 243
Stop words ... 243
Spelling correction ... 244
Using IR Expert and Adaptive Learning ... 244
IR Expert scirexpert file ... 244
Creating an IR file ... 245
Updates to IR files ... 245
IR keys and non-IR keys ... 245

Building IR keys 246
Multiple files containing IR keys 246
Find Solution 247
Edit queries for Find Solution 247
Customizing IR Expert 248
Implement IR searches 248
Chinese, Japanese, and Korean language analyzer 250
Searching for Chinese or Japanese and English words in a Chinese or Japanese environment 250
Searching for Chinese or Japanese and English words in an English environment 251
IR Expert file descriptions 251
Normals dictionary 253
Stem dictionary 253
Stemming example 253
Stop word file 254
Suffix file 254
Record retrieval 255
The Database Manager utility 255
Determining System Navigator menu options 256
Open a record using Database Manager 256
Record list 257
Refresh a record list 257
Edit a record 258
Multiple-record functions 258
Adding multiple records 259
Mass Update 259
Mass add/update function errors 260
Update multiple records with Mass Update 260
Template mass update and required fields 262
Updating multiple records 263
Add multiple records using a literal value 263
Add multiple records using a variable value 266
Avoid invalid duplicate or null key errors 268
Update multiple records using a literal value 270
Update multiple records using a variable value 271
Delete multiple records 272
View multiple records using an alternate form 274
Mass Close 275
Mass close interaction records 275
Determine the number of records in a record list 276

Print multiple records ... 277
Refresh a record list ... 277
Options for Mass Update ... 277
Single-record functions ... 278
Add a record ... 278
Clear an initial record ... 279
Delete a record ... 279
Duplicate an existing record ... 279
Print a record ... 280
Search for newly added records ... 281
Update an existing record ... 281
View a single record using an alternate form ... 282
Queries ... 282
Simple query expressions ... 282
Complex query expressions ... 283
Logical operators ... 283
Relational operators ... 284
Starts with relational operator ... 285
Equal to relational operator ... 285
Less than relational operator ... 285
Greater than relational operator ... 285
Using ranges in queries ... 286
Like relational operator ... 286
Wildcard characters ... 286
Create a query using not in statements ... 287
Create a query using an or statement or an and statement ... 287
Use like queries to search for a string of characters ... 288
Create a query using an or statement or an and statement ... 289
Create a query using array fields ... 290
Create a query using more than one field ... 290
Perform IR Expert queries ... 292
Use the equal to relational operator ... 293
Use the greater than relational operator ... 293
Use the less than relational operator ... 294
Use the like relational operator ... 295
Use the not symbol with logical or relational operators ... 295
Use the starts with relational operator ... 296
Create a query using a simple query expression ... 296
Create a query using isin in statements ... 298

Use the index function in a query 298
Use the lng function in a query 299
Use the tod function in a greater than or less than query 300
Retrieving records by using the record list method 300
Retrieve all records in a table 301
Retrieve records within a range 301
Retrieving records using the Query window method 302
When should I use the Query Window instead of the record list query method? 302
Excluding records from a query 302
Using functions in a query 302
Open the Query Window 302
Use the Query Window 303
Stored queries 303
Stored query option definitions 304
Add a stored query 304
Update a stored query 306
Append stored queries 307
Run a stored query 309
Cross-table join query improvements 310
Cross-table join functions 311
lng() 311
null() 312
nullsub() 312
operator() 313
substr() 313
tod() 314
toupper() and tolower() 314
Cross-table join query examples and use cases 314
Reserved words 315
Use case: Incident Analyst 315
Use case: Department manager 317
Use case: JavaScript Developer 319
Example: Query two tables 320
Example: Query more than two files 320
Example: Sort 321
Example: Count 321
Example: Handle two files that have the same field name 322
Example: Handle BLOB, CLOB, text and images 322
Example: Handle a one-to-many relationship 323
Limitations on cross-table join queries 323

Error Messages 324
Database performance tuning 325
Tuning: Improving query speed 325
Tuning: Number of fields in files 325
Troubleshooting and tuning queries 326
Tuning: Fully keyed queries 326
True queries 326
Partially keyed queries 326
Nonkeyed queries 326
Running the database maintenance utility 327
Run the database maintenance utility in foreground (interactively) on Unix 327
Run the database maintenance utility in background (batch) on Unix 328
Run the database maintenance utility in foreground (interactively) on Windows 328
Run the database maintenance utility in the background (batch) on Windows 329
Entity relationship management 330
Entity Relationship Diagram utility 330
Generate an entity relationship diagram for a file 332
Interact with an entity relationship diagram 335
View relationships of a specific field in an entity relationship diagram 336
Identify misconfigured reference fields 338
Missing Reference Report utility 340
Start the refcheck.startup scheduler 342
Enable data reference checking for a file 342
Skip data reference checking for a relationship 343
List of skipped out-of-box relationships 344
Run a data reference check against selected files 347
View the status of existing data reference checks 348
Cancel a scheduled data reference checking job 349
View missing reference report results 350
Identify the root cause of a missing reference 352
Relationship Manager 359
Build relationships 360
Search for relationships between specific tables and fields 361
Access the relationship definition of a relationship record 365
Add a relationship record 366
Update an invalid or fixed relationship 367
Send Documentation Feedback 369

Database Administration

This section contains information about converting, maintaining, tuning, and troubleshooting your Relational Database Management System (RDBMS). The topics are intended for experienced HP Service Manager system and database administrators responsible for installing and implementing databases or individuals hosting Service Manager data and assisting in database conversion.

Other information contained in this section:

- Information required to configure and use Service Manager databases
- Practical information to learn about various methods used to perform queries against information contained in the database
- Information and steps used to import, export, purge, and archive records
- Information and steps used to run the Service Manager out-of-box maintenance utilities
- Information about how to use IR Expert
- Database tuning and troubleshooting tips
- Best practices and examples using out-of-box data

Supported RDBMS

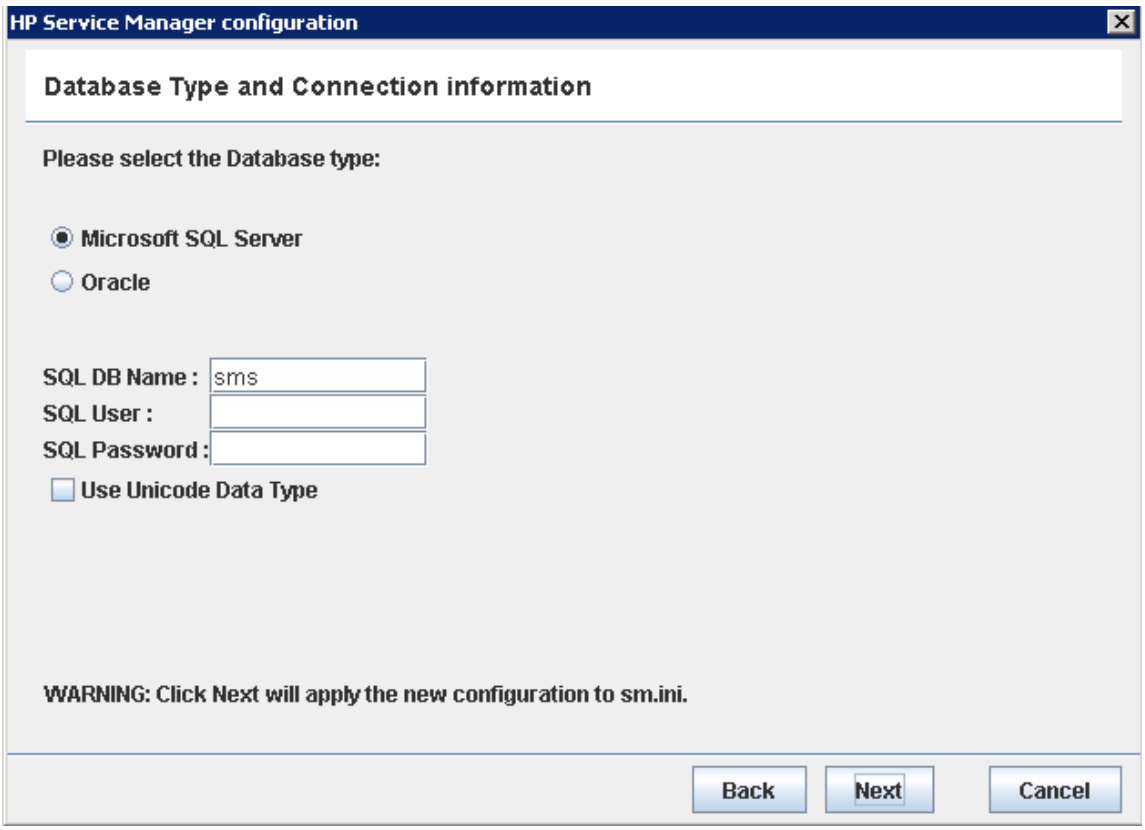
For a list of supported Relational Database Management Systems (RDBMS) for all versions of HP Service Manager, view the support matrix from the HP Software support Web site, accessible from the related topics.

HP Service Manager does not replicate vendor documentation or tasks used to maintain a Relational Database Management System (RDBMS). For the latest information, see your RDBMS Vendor documentation and the RDBMS Vendor support Web site.

Microsoft SQL Server database Unicode support

Prior to version 9.40, HP Service Manager used the VARCHAR, CHAR, and TEXT data types to store text strings in a Microsoft SQL Server database. Microsoft SQL Server relies on database collation to support a specific language. Although some languages share the same collation, most languages do not. As a result, Service Manager did not support multiple languages when running on a SQL Server database.

Service Manager 9.40 is enhanced to support the Unicode data type on SQL Server. You can store text string in multiple languages by using the Unicode data type. When you run the server configuration utility, if you select the **Use Unicode Data Type** check box for SQL Server (see the following figure), Service Manager creates data with the Unicode data type when it loads the applications and demo data.



The following table provides the mapping between the old and new data types.

Non-Unicode	Unicode
VARCHAR	NVARCHAR
CHAR	NCHAR

Non-Unicode	Unicode
	Note: This occurs only when the CHAR field length is greater than 1.
TEXT	NVARCHAR(MAX)

The operation is not reversible, which means once you have enabled this option, you must use the Unicode data type when adding new data. When your SQL Server database uses the Unicode data type, the read-only flag **Use Unicode Data Type** in the **sqlserver** record of the **sqldbinfo** table is automatically turned on.

Note: The following four system tables will not use Unicode: **dbdict**, **licenseinfo**, **lock**, and **lockshared**.

Support of the Unicode data type is available only for new installations of the Service Manager 9.40 or later applications. If you are upgrading from a previous version of the applications, this option is not available.

For information about how to enable the Unicode support, see the *Configure the Service Manager server* section in the *HP Service Manager Interactive Installation Guide*.

Case-Sensitivity

HP Service Manager works with both case-sensitive and case-insensitive databases, enabling you to set things up the way that best meets your needs. Your users will be able to search for data case-sensitively or insensitively, as you choose. To achieve this, all servers and clients must be set to the same sensitivity.

Service Manager supports both case-sensitive and case-insensitive databases on Microsoft SQL server and Oracle. If you use IBM DB2 Universal, you must set the system to be case-sensitive. You can configure the case-sensitivity of your Microsoft SQL server or Oracle database when you set it up. For more information, refer to the HP Service Manager Installation Guide.

- Prior to converting a system to be case-insensitive, you must first ensure that your RDBMS supports case-insensitivity.
- HP recommends implementing case-insensitive searching on a test system prior to setting it up on your production system.

Load/unload implications of case-sensitivity

When you load/unload data from a Service Manager system that connects to a case-sensitive database and load it into one that connects to a case-insensitive database, you can choose whether you want to Add and Update records or to Only add new records.

- If you select **Record Load Options - Add/update records**, the new records will be inserted into the table and the existing records will be updated.
  - **Example 1**: Suppose that you use a case-sensitive database to create an unload file that has two operator records - "operator01" and "operator02". You then load this unload file into a case-insensitive database which already has an operator record "OPERATOR01". In this case, "OPERATOR01" will be updated by "operator01" and "operator02" will be added.
    Note: If you try to load records with unique IDs that differ only in case from records that already exist in the case-insensitive database, the data in the unload will update the data in the current system.
  - **Example 2**: Suppose that you use a case-sensitive database to create an unload file that has two operator records - "operator01" and "OPERATOR01". You then load this unload file into a case-insensitive database which already has an operator record "OPERATOR01". In this case, "OPERATOR01" will be updated by "operator01" and subsequently by "OPERATOR01", or visa-

versa depending on the order of those records in the unload file.
Note: If you try to add/update two records with unique IDs that differ only in case to a case-insensitive database, the second record loaded will update the first record loaded.

- If you select **Record Load Options - Only add new records**, the new records will be inserted into the table.
  - **Example 1**: Suppose that you use a case-sensitive database to create an unload file that has two operator records - "operator01" and "operator02". You then load this unload file into a case-insensitive database which already has an operator record "OPERATOR01". In this case, "OPERATOR01" will remain unchanged and "operator02" will be added.
    Note: If you try to load records with unique IDs that differ only in case from records that already exist in the case-insensitive database, the data in the unload will add only the records which do not yet exist in the case-insensitive database.

  - **Example 2**: Suppose that you use a case-sensitive database to create an unload file that has two operator records - "operator01" and "OPERATOR01". You then load this unload file into a case-insensitive database which does not have either operator record "OPERATOR01" or "operator01". In this case, "operator01" will be added but "OPERATOR01" will not be.
    Note: If you try to add two records with unique IDs that differ only in case to a case-insensitive database, the second record will not be added.

JavaScript/RAD implications of a case-insensitive database environment

In a case-insensitive environment, such as when Service Manager is set up to connect to Microsoft SQL Server with a case-insensitive code page or Oracle with case-insensitive settings, you should be aware that:

- Only Service Manager queries are evaluated case-insensitively.

- All Service Manager expressions are still evaluated case-sensitively.
  This includes RAD expressions, JavaScript expressions as well as the expressions within DVD conditions in Service Manager forms.

For example, given a database that contains a contacts record where the actual last.name is stored in the database as "Smith," the following code will successfully select the record but fail the Java Script test:

```
var contactList = new SCFile( "contacts" );
var findContact = contactList.doSelect( "last.name=\"SMITH\"" );
var lastName = "SMITH";
```

```
   if ( lastName == findContact.last_name )    // This expression is evaluated case-
sensitively.
   {
      print("Hello!");                         // This code will never be executed.
   }
```

In this example:

- The query in contactList.doSelect() will be evaluated case-insensitively, therefore findContact will get the record back.
- All other JavaScript expressions will evaluate case-sensitively. Therefore, statements within the "if" block will never be executed.

MS SQL Server collation support

HP Service Manager supports two types of collations :

- Those ending in some form of _BIN (binary)
- Those ending in some form of _CI_AS (case-insensitive, accent sensitive)

_BIN is used for case-sensitive collation, meaning both, comparisons (WHERE NAME="ADMINISTRATOR") and sorting (ORDER BY NAME) are case-sensitive. Collations ending in _CS_AS claim to be case-sensitive, but that only applies to comparisons. For sorting, they use a "dictionary order," which is essentially case-insensitive.

_CI_AS is the supported case-insensitive collation. It uses case-insensitive comparisons (WHERE NAME="AdMinIstrator") and case-insensitive sorting. Do not use _CI_AI since Service Manager has no concept of accent insensitivity.

The sort order becomes important in the case of a combined SQL/IR search. In that case, IR has to match its own results with the results returned by SQL, and for that it assumes binary or case-insensitive sorted order.

Note, that such queries might take place even if you don't know about them. For example in the Service Catalog, when ordering, a user enters an IR query, but the RAD appends a long non-IR query to it. This results in a mixed query.

The other aspect of the collation is the code page. The code page defines the code page of VARCHAR columns, NVARCHAR are always in UTF-16. "General" refers to the American default Windows code page, Win 1252.

Service Manager supports the following SQL Server code pages:

Code page	Description
874	MS cp874 Thai
932	MS cp932 Japanese
936	MS cp936 Simplified Chinese
949	MS cp949 Korean
950	MS cp950 Traditional Chinese
1250	MS cp1250 Central European
1251	MS cp1251 Russian, Bulgarian, Serbian
1252	MS cp1252 Latin/Western European
1253	MS cp1253 modern Greek
1254	MS cp1254 Turkish
1255	MS cp1255 Hebrew
1256	MS cp1256 Arabic
1257	MS cp1257 Estonian, Latvian, Lithuanian
1258	MS cp12578 Vietnamese

You can generate the list of all the code pages supported by Service Manager with the command, `sm -reportlanguages`. Only those code pages that start with "mswin" are used in the context of SQL Server.

If you are using an unsupported code page, you will see this warning in the log file: `W SQL Server uses a code page <nnn> unsupported by HP Service Manager which retrieved codepage <nnnn>`.

Oracle case-sensitivity

You can set the Oracle database to the desired case-sensitivity when you create it. Refer to the HP Service Manager Installation Guide for instructions.

Refer to the Case-sensitivity section of the Service Manager Support Matrix for versions of Oracle for which Service Manager supports case-insensitive mode.

Requirements for using a case-insensitive Oracle database

If you want to use a case-insensitive Oracle database with Service Manager instead of using the sqlupper:1 solution, you must meet the following requirements:

- NLS_SORT=BINARY_CI and NLS_COMP=LINGUISTIC must be set in the Oracle instance level.
  - If these values are set as given above, Service Manager will issue an ALTER SESSION statement to match the NLS_SORT and NLS_COMP session parameter with the NLS_SORT and NLS_COMP instance parameter.
  - If these values are not set or are different from the given values, Service Manager will treat it as case-sensitive Oracle and issue an ALTER SESSION statement to modify those parameters to be NLS_SORT=BINARY and NLS_COMP=BINARY.

  This ensures that all Service Manager server processes within a horizontal or vertical scaling group have consistent Oracle case-sensitivity settings.
- All unique keys in Service Manager tables must have a matching case-insensitive unique index.

Limitations of case-insensitive Oracle

You will not be able to rename or change a column size of keyed fields from the Service Manager Dbdict Utility when using a case-insensitive Oracle instance. Instead, select generate DDL when modifying columns from Dbdict Utility. An Oracle database administrator can then drop affected function based index, apply the DDL, and recreate the function based index on the Oracle database.

Validate case-insensitive unique indexes

During the initial system load, Service Manager created a set of case-insensitive indexes for each table, based on the keys in that table. Service Manager logs these indexes first time reads a table when the `sqldebug:1` parameter is in the `sm.ini` file. You can review the settings created for each table by viewing the `sm.log file`.

The case-insensitive unique indexes should be created as Oracle function based indexes where the Column Expression is NLSSORT("<field name>",'nls_sort=''BINARY_CI''').

To verify whether the case-insensitive unique indexes are being with the correct column expression:

1. Set `sqldebug:1` in the `sm.ini`file, located in the `<SM_install_location>\Server\RUN` directory.
2. Start Service Manager.
3. Check the `sm.log` file, located in the <SM_install_location>\Server\logs directory.

The following entry in the log file indicates that the Oracle instance is set to case insensitive, and that you were able to connect to it successfully.

```
RTE I Oracle server settings for language, territory and character set: AMERICAN_
AMERICA.AL32UTF8 (AL16UTF16)
RTE I OCI Client settings for language, territory and character set: AMERICAN_
AMERICA.AL32UTF8 (UTF16)
....
RTE I Oracle instance setting for NLS_SORT is set to BINARY_CI
RTE I Oracle instance setting for NLS_COMP is set to LINGUISTIC
...
RTE I Oracle session is set up in CASE INSENSITIVE mode
```

The following information in the log file indicates that the Dbdict table has an index, DBDICTM1C989DE64, with a key called "NAME", which is case-insensitive.

```
RTE D Table Name: DBDICTM1

RTE D Schema Name                    Index Name                     Type Column
Name                    Column Expression

RTE D ------------------------------ ------------------------------ ---- ----------
------------------- ----------------------------------------

RTE D SMDB                           DBDICTM1C989DE64               U    SYS_
NC00003$                  NLSSORT("NAME",'nls_sort=''BINARY_CI''')

RTE D ------------------------------ ------------------------------ ---- ----------
------------------- ----------------------------------------
```

Backups

When deciding what files you need to back up, note that the following file types are essential to run HP Service Manager:

- sm.ini
- sm.cfg
- All RDBMS data files

Purging out-of-box data

Before your system goes live, all testing data records and out-of-box demonstration data records should be deleted. We recommend that you use the pre-defined PurgeOutofBoxData unload script record to remove this data.

To purge the data, follow these steps:

1. Go to Database Manager, and open the **unload** table.
2. Select the **PurgeOutofBoxData** record.
3. Double-check that all the listed tables should be purged.
   - If a table should not be purged, remove the line from the unload script record.
   - Add or modify a query (for example, add `name="failsave"` to the **operator** line) as needed.
   - Add more tables (such as **knownerrortasks** and **rootcausetasks**) to the list of tables to purge, if information was entered during testing.
4. Click **Proceed**.

Attachments

Service Manager stores each attachment as a set of separate records in the SYSATTACHMENTS table. The SYSATTACHMENTS table stores each attachment as a logical record consisting of:

- Unique identifier
- File name
- MIME type
- File size
- Application (the name of the table to which the attachment relates)
- Topic (the value of the primary key in the related table)

For example, Application would contain the name kmdocument for a Knowledge Management document attachment.

Each logical record consists of one or more segments where the server stores portions of the attachment. Administrators can set the attachment segment size with the `attachmentsegmentsize` system parameter.

The SYSATTACHMENTS table allows administrators to easily identify all the attachments stored in their system. In addition, it improves system performance since the server can read information about each attachment individually. The server no longer has to open all attachment records to generate a list of attachments. In addition, the server no longer has to open all other attachments for a user to make a change in any individual attachment.

The server includes the new attachment table in all new Service Manager installations. Customers upgrading from previous versions of HP ServiceCenter must convert their existing attachments to the new table. See the HP Service Manager Upgrade Guide for instructions.

Support for blocking attachments with certain file extensions

To prevent potentially dangerous attachment files from being submitted to Service Manager, Service Manager provides support for blocking certain types of attachments submitted through the Windows

and web clients, as well as through web services, based on a pre-defined list of forbidden file extensions.

At startup, the web tier and the Windows client retrieve the forbidden list, which is stored in the extensionstate table in the database; if no list is available, the clients use a default list of forbidden file types stored on the client side (see "Default forbidden list of attachment file extensions" on page 29). The clients block attachments as follows:

- When users attempt to attach a file whose extension is in the forbidden list to a record , the clients disallow attaching of the file and display an error (where <filetype> is the file extension): "File type: <filetype>, is not allowed as an attachment."
- If the clients retrieve an existing attachment whose extension is included in the forbidden list, the clients append ".UNSAFE" to the file name to prevent the file from being automatically ‘run’ on the user’s machine.

During web services processing, if a transaction submits or retrieves a file as an attachment whose extension is included in the forbidden list, the server appends “.UNSAFE” to the submitted or retrieved file name.

System Administrators can add, delete, and update records in the extensionstate table to customize the forbidden list of attachment file extensions. For details, see "Customize the forbidden list of attachment file extensions" on page 33.

If a search engine is installed and connected to the system, System Administrators also need to update the sclib knowledgebase records, to include .unsafe to the list of file extensions that should be skipped for knowledgebase indexing. See "Add .unsafe to the list of file extensions to skip for knowledgebase indexing" on page 34.

Additional protection from the web client side

The web client provides additional protection using a so-called "whitelist" to control what types of attachments are allowed to be submitted to Service Manager. From the web client, you can submit only attachments whose file types are in this whitelist. In the out-of-box system, this whitelist includes the following file types, which are considered safe:

- bmp
- jpg
- jpeg
- png

- gif
- txt
- log
- unl
- csv

This whitelist is defined in the web tier configuration file (web.xml) as shown below (see the four "allowed" parameters highlighted in bold). You can adjust the list at your own risk.

```
<servlet>
    <description>File Upload Servlet.</description>
    <display-name>File Upload Servlet</display-name>
    <servlet-name>FileUpload</servlet-name>
    <servlet-class>com.hp.ov.sm.client.webtier.FileUploadServlet</servlet-class>
    <init-param>
      <param-name>allowed</param-name>
      <param-value>bmp,jpg,jpeg,png,gif,txt,log,unl,csv</param-value>
    </init-param>
  </servlet>
  <servlet>
    <description>Image Upload Servlet.</description>
    <display-name>Image Upload Servlet</display-name>
    <servlet-name>ImageUpload</servlet-name>
    <servlet-class>com.hp.ov.sm.client.webtier.FileUploadServlet</servlet-class>
    <init-param>
      <param-name>allowed</param-name>
      <param-value>bmp,jpg,jpeg,png,gif</param-value>
    </init-param>
    <init-param>
      <param-name>success</param-name>
      <param-value>../imageUploaded.jsp</param-value>
    </init-param>
  </servlet>
  <servlet>
    <description>Attachment upload servlet</description>
    <display-name>Attachment Upload Servlet</display-name>
    <servlet-name>AttachmentUpload</servlet-name>
    <servlet-class>com.hp.ov.sm.client.webtier.FileUploadServlet</servlet-class>
    <init-param>
      <param-name>allowed</param-name>
      <param-value>bmp,jpg,jpeg,png,gif,txt,log,unl,csv</param-value>
    </init-param>
    <init-param>
      <param-name>success</param-name>
```

```
      <param-value>../attachmentUploaded.jsp</param-value>
    </init-param>
  </servlet>
  <servlet>
    <description>Attachment upload servlet for AJAX request</description>
    <display-name>Attachment Upload Servlet for AJAX request</display-name>
    <servlet-name>AttachmentUploadAjax</servlet-name>
    <servlet-class>com.hp.ov.sm.client.webtier.FileUploadAjaxServlet</servlet-
class>
    <init-param>
      <param-name>allowed</param-name>
      <param-value>bmp,jpg,jpeg,png,gif,txt,log,unl,csv</param-value>
    </init-param>
  </servlet>
  <servlet>
    <description>Unique attachment upload servlet</description>
    <display-name>Unique Upload Servlet</display-name>
    <servlet-name>UniqueUpload</servlet-name>
    <servlet-class>com.hp.ov.sm.client.webtier.FileUploadServlet</servlet-class>
    <init-param>
      <param-name>allowed</param-name>
      <param-value>bmp,jpg,jpeg,png,gif,txt,log,unl,csv</param-value>
    </init-param>
    <init-param>
      <param-name>success</param-name>
      <param-value>../uniqueUploaded.jsp</param-value>
    </init-param>
  </servlet>
```

Default forbidden list of attachment file extensions

Service Manager blocks attachments whose file name extensions are included in a forbidden list retrieved from the database. If such a forbidden list is not available from the database, Service Manager blocks attachments with a file name extension listed in the following table.

File name extension	File type
.ade	Access Project Extension (Microsoft)
.adp	Access Project (Microsoft)
.app	Executable Application
.asp	Active Server Page
.bas	Active Server Page

File name extension	File type
.bat	Batch Processing
.cer	Internet Security Certificate File
.chm	Compiled HTML Help
.cmd	DOS CP/M Command File, Command File for Windows NT
.com	Command
.cpl	Windows Control Panel Extension (Microsoft)
.crt	Certificate File
.csh	csh Script
.der	DER Encoded X509 Certificate File
.exe	Executable File
.fxp	FoxPro Compiled Source (Microsoft)
.gadget	Windows Vista gadget
.hlp	Windows Help File
.hta	Hypertext Application
.inf	Hypertext Application
.ins	IIS Internet Communications Settings (Microsoft)
.isp	IIS Internet Service Provider Settings (Microsoft)
.its	Internet Document Set, Internet Translation
.js	JavaScript Source Code
.jse	JScript Encoded Script File
.ksh	UNIX Shell Script
.lnk	Windows Shortcut File
.mad	Access Module Shortcut (Microsoft)
.maf	Access (Microsoft)
.mag	Access Diagram Shortcut (Microsoft)
.mam	Access Macro Shortcut (Microsoft)
.maq	Access Query Shortcut (Microsoft)

File name extension	File type
.mar	Access Report Shortcut (Microsoft)
.mas	Access Stored Procedures (Microsoft)
.mat	Access Table Shortcut (Microsoft)
.mau	Media Attachment Unit
.mav	Access View Shortcut (Microsoft)
.maw	Access Data Access Page (Microsoft)
.mda	Access Add-in (Microsoft), MDA Access 2 Workgroup (Microsoft)
.mdb	Access Application (Microsoft), MDB Access Database (Microsoft)
.mde	Access MDE Database File (Microsoft)
.mdt	Access Add-in Data (Microsoft)
.mdw	Access Workgroup Information (Microsoft)
.mdz	Access Wizard Template (Microsoft)
.msc	Microsoft Management Console Snap-in Control File (Microsoft)
.msh	Microsoft Shell
.msh1	Microsoft Shell
.msh2	Microsoft Shell
.mshxml	Microsoft Shell
.msh1xml	Microsoft Shell
.msh2xml	Microsoft Shell
.msi	Windows Installer File (Microsoft)
.msp	Windows Installer Update
.mst	Windows SDK Setup Transform Script
.ops	Office Profile Settings File
.pcd	Visual Test (Microsoft)
.pif	Windows Program Information File (Microsoft)
.plg	Developer Studio Build Log
.prf	Windows System File

File name extension	File type
.prg	Program File
.pst	MS Exchange Address Book File, Outlook Personal Folder File (Microsoft)
.reg	Registration Information/Key for W95/98, Registry Data File
.scf	Windows Explorer Command
.scr	Windows Screen Saver
.sct	Windows Script Component, Foxpro Screen (Microsoft)
.shb	Windows Shortcut into a Document
.shs	Shell Scrap Object File
.ps1	Windows PowerShell
.ps1xml	Windows PowerShell
.ps2	Windows PowerShell
.ps2xml	Windows PowerShell
.psc1	Windows PowerShell
.psc2	Windows PowerShell
.tmp	Temporary File/Folder
.url	Internet Location
.vb	VBScript File or Any VisualBasic Source
.vbe	VBScript Script File, Visual Basic for Applications Script
.vbs	VBScript Script File, Visual Basic for Applications Script
.vsmacros	Visual Studio .NET Binary-based Macro Project (Microsoft)
.vsw	Visio Workspace File (Microsoft)
.ws	Windows Script File
.wsc	Windows Script Component
.wsf	Windows Script File
.wsh	Windows Script Host Settings File
.xnk	Exchange Public Folder Shortcut

Customize the forbidden list of attachment file extensions

Applies to User Roles:

System Administrator

You can customize the forbidden list of attachment file extensions, which is stored in the database. You can add more file extensions to or remove existing ones from the list.

Note: When your customization is complete, you must restart the web tier's web application server (for example, Tomcat, WebSphere) or restart the Windows client for the changes to take effect.

To customize the forbidden list:

1. Click **System Administration** > **Base System Configuration** > **Miscellaneous** > **File Extensions**.
2. Click **Search**.

   A list of file extension records is displayed.
3. To add a new record:
   a. Select a record from the list.
   b. Update the following fields:
      - **File Extension**: Type a new extension without the dot character (.) For example, type "bat" rather than ".bat".
      - **Unsafe**: Check this box to block attachments with this file extension or uncheck this box to unblock them.
      - **File Type**: Type a description of this new file type.
   c. Click **Add**.

      The new file extension is added to the list.
4. To update an existing record:
   a. Select the record from the list.
   b. Check or uncheck the **Unsafe** box.
   c. Update the **File Type** description if necessary.

d. Click **Save**.

The record is updated.

5. To delete a record:
   a. Select the record from the list.

   b. Click **Delete**.

   c. Click **Yes** to confirm the deletion.

   The record is removed from the list.

6. Do the following for the changes to take effect:

   - If you are in the Windows client, exit and then restart the client.
   - If you are in the web client, restart the web application server.

Add .unsafe to the list of file extensions to skip for knowledgebase indexing

Applies to User Roles:

System Administrator

To prevent Service Manager from indexing .unsafe attachments for knowledgebase searches, you need to add .unsafe to the list of file extensions that should not be indexed or extracted. You do so by updating the following sclib type knowledgebases: Iincident_Library, Interaction_Library, Knowledge_Library, Knownerror_Library, and Problem_Library.

Note: This task requires that you have a search engine installed and correctly configured.

To add .unsafe to the list of file extensions to skip:

1. Click **Knowledge Management** > **Configuration** > **Knowledgebases**.

2. Click **Search**.
   A list of records displays: Iincident_Library, Interaction_Library, Knowledge_Library, Knownerror_Library, and Problem_Library.

3. Select a record in the list, and click the **Type information** tab.

4. In the **Skip these extensions** field, add **.unsafe** to the existing file extension list, using a semi-colon as the separator. For example, `jpg;bmp;gif;exe;unl;unsafe`
5. Click **Save**.
6. For the other records in the record list, repeat steps 3 through 5.

View the attachment history

Every time that a user adds or deletes an attachment in HP Service Manager, the action is recorded in the activityactions table.

Note: If a file does not have a related object file or if the activity log table is not defined, the action is not logged.

To view the attachment history of a specific record, open that record and browse to the "Activity" section. If an attachment has been added to or deleted from the record, the following information is displayed in the "Activity" section.

Field Name	Values
Number	Record ID number
Type	• Attachment added • attachment deleted
Datestamp	Current date
Operator	Current operator
Description	• <*Attached file name*> added • <*Attachment file name*> deleted
Customer visible	FALSE

System Definition

The System Definition section of the System Navigator is an administrative tool that enables you to view and modify system tables and their components:

- Fields
- Keys
- Links
- Forms
- Records

You can use the System Definition section if you have the Admin plug-in installed and the SysAdmin capability word enabled in your user profile. If you have SysAdmin rights, but the Admin plug-in is not installed, you can view the System Definition section but you cannot change it.

Administrators can do the following:

- View, add, change, or delete forms, links, records, triggers, and Data Policy records.
- Use the System Definition utility to add help or create an alias for any field.
- View, add, change, or delete database dictionary definitions.
- View, add, change, or delete SQL mappings to a back-end RDBMS.
- Open, reset, regenerate, and copy database dictionary definitions.
- Add, change, copy, and delete keys and fields.
- Count the number of records in logical tables

Each time you right-click a node in the System Definition section, the System Navigator displays a Context menu of available actions.

System Definition utility

The System Definition utility is a graphic view of existing table elements and properties and a quick way to define new tables. It enables you to accomplish the tasks traditionally handled by the database dictionary utility. The System Definition utility has two tabs: tables and fields. Service Manager displays SQL mapping information only when mapping information is available. Each time you double-click any table in the System Definition area of the System Navigator, the System Definition utility displays the Table tab first. You can access the other tabs by clicking the labeled tabs or links in the Editor.

If you change a table element, an asterisk marks the table and the element in the System Navigator until you click **Save**.

Access the System Definition utility

Applies to User Roles:

System Administrator

To access the System Definition utility:

1. From the System Navigator, click **System Definition**.
2. Expand the name of any table in the record list.

   Note: You can type the first letter of a table name to jump to that alphabetic section. For example, type o to jump to the oncall table.

Table tab

The Table tab summarizes the properties of an HP Service Manager table, provides links to other table-related objects, and allows you to map a database dictionary record to an RDBMS.

The Table overview contains these sections:

- Table definition
- Table management
- SQL Properties

- Associated objects
  - Forms
  - Links
  - Triggers

Table definition

The Table definition section contains the following information:

Field	Contents	Notes
Caption	Optional. Caption information may appear as a label in the data policy rules for the table. Service Manager uses some Caption fields to specify an index value for a table of localized text.	An administrator can customize this field. For example, you can display the name of table.
Description	Optional. The Knowledgebase uses the first line of the table description.	An administrator can customize this field. For example, you can specify information about the table.
SQL base name	SQL mapping uses a unique base name for each table. The maximum length is 13 alphanumeric characters. When you create a new table, Service Manager creates a valid SQL base name and assigns it to the new table.	To avoid future upgrade conflicts, it is helpful to specify a base name that does not overlap Service Manager out–of–box table names.
Revision number	Specifies the cumulative number of changes to the table.	Read only.
Add, delete, or edit fields and keys	Click to advance to the Fields and Keys tab.	

If you make any changes to the table definition fields or other table information, click **Save**.

Table management

Use the Table management section to complete routine maintenance activities. When you select the action, the System Definition utility prompts you for the next response required to complete the task. You can:

- Delete the table
- Rename the table
- Copy the table definition (only)
- Copy the table definition and data
- Regenerate the table
- Reset the table data

Service Manager always asks for confirmation before you complete a task that makes irreversible changes. For example, you must confirm when you try to delete, regenerate, or reset a table.

Editing System Definition nodes

When you double-click any node in System Definition:

- The System Definition utility displays fields, keys, links, and triggers.
- Forms Designer displays forms.
- Service Manager displays records collections in a record list or a single record in the associated display form.

Access field help from the System Navigator

Applies to User Roles:

System Administrator

To access field help from the System Navigator:

1. Click **System Definition > Tables >** ***tablename***.
2. Double-click a table name to view information about that table.
3. Click the **Fields and Keys** tab.
4. Select any field in the **Fields** section.

5. Select one of these links to create or edit the default help record.
   - **Create default help on this field**. This link appears when no field help exists for this field. Click this link to create field help.
   - **Edit default help on this field**. This link appears when field help already exists for this field. Click this link to edit existing field help.
6. Click **Search for specific help on this field in forms** to open and edit existing form-specific help for this field.
7. If you make changes, click **Save**.
8. Click **OK**.

Forms, triggers, and links

When you choose one of these components associated with the table, Service Manager opens the appropriate form or application to edit the component.

- If you double-click a form, Forms Designer displays the selected form.
- If you double-click a trigger, Service Manager displays the related trigger record.
- If you double-click a link, Service Manager displays the related link record.

Edit the form, trigger, or link. If you make any changes to forms, triggers, or links, click **Save**.

Update a form

Applies to User Roles:

System Administrator

To update a form:

1. Click **Tailoring** > **Forms Designer** in the System Navigator, or type `fd` on the Service Manager command line and then press **Enter**.
2. Type a form name and click **Search**.

   Service Manager opens a list of forms. Click the form you want to update to open it.

3. Click **Design** to open design mode.
4. Add controls in any of the following ways:
   - Drag and drop the control from the toolbar to the drawing canvas to create a default–sized object.
   - Select a control in the toolbar and then click anywhere on the drawing canvas to create a default–sized object.
   - Drag and drop field(s) from a table in the System Definition onto the drawing canvas.
   - Cut, copy, or paste a control within the same form, from a different form, or from different Service Manager client.

   The properties for the control you selected appear in the properties view.
5. Specify the name of the form in the Value text box of the Name property.
6. Specify the database field to access in the Value text box of the Input property.
7. Specify the height, width, and other properties as desired.
8. To delete a control from the drawing canvas, select the control and press **Delete**.
9. Click **Web Preview** to see what the form will look like in the Web client.
10. Click **OK** to see what the form will look like in the Windows client.
11. Click **OK** to save your changes.
12. Open the form using Database Manager to test your changes:
    a. Click **Tailoring** > **Database Manager**.
    b. Click **Search**.

       Service Manager opens a list of forms.
    c. Click the form you just created to open it.

Triggers

A trigger monitors a file for certain types of activity, such as Adds, Updates, or Deletes, and launches a specified JavaScript or RAD application when the appropriate conditions occur. You can supply custom

JavaScript or RAD code to launch a specific action when the trigger conditions apply.

Create a Trigger record

Applies to User Roles:

System Administrator

Triggers monitor files for certain activity types, such as Adds, Deletes, or Updates, and launches a specified RAD application when the appropriate conditions are met.

To create a trigger record:

1. From the System Navigator, click **System Definition** > **Tables** > **triggers**.
2. From the Keys sections, click **New**.
3. From the General section, select the trigger type from the More Actions menu.
4. Click **Add**.
5. From the Fields section, modify the properties of the available fields to define the trigger. You can use the table below as a general guideline for the most commonly-used parameters.

Property	Description
trigger.name	Any unique, meaningful name.
table.name	Select the file to be monitored by this trigger.
trigger.type	Select the specific activity to react to: **1 - Before Add** - When adding a new record to this file, launch the application prior to committing the addition. **2 - After Add** - When adding a new record to this file, launch the application after committing the addition. **3 - Before Update** - When modifying a record in this file, launch the application prior to committing the change. **4 - After Update** - When modifying a record in this file, launch the application after committing the change. **5 - Before Delete** - When deleting a record in this file, launch the application prior to performing the deletion.

Property	Description
	6 - After Delete - When deleting a record in this file, launch the application after performing the deletion. **Note:** Only triggers of *type 4* (After Update), and *type 6* (After Delete) are supported by Cascade Updates.
application.name	The RAD application to launch. **Note:** The RAD application may not require parameters and will only have access to $L.old, the record prior to the change, and $L.new, the post-change record. **Note:** For most Cascade Updates, set to **cascade.update.wrapper**. The only exception is if the file you are monitoring is the device file or a related device attribute file, such as joincomputer. In those cases, set to **am.cascade.update.wrapper**.

Edit a trigger

Applies to User Roles:

System Administrator

To edit a trigger:

1. Click **Tailoring** > **Database Manager**.
2. Type **triggers** in the Table field, and then click **Search**.
3. Click **Search** again.

   A list of records opens.
4. Select a record to edit.
5. Update the following information about the trigger:
   - **Trigger Name**
   - **Table Name**
   - **Trigger Type**

- **Application**
- **Script**

6. If you make changes, click **Save**.

Modify an existing link

User Role: System Administrator

To modify an existing link.

1. Click **Tailoring** > **Tailoring Tools** > **Links**.
2. Type optional search criteria, and then click **Search**.
3. Double-click a link to view or change it. If necessary, press Ctrl+H to view help for each field. When the link record opens, the following options are available from the More Actions menu:

Option	Value
Insert Line	Opens a window to prompt for the number of lines to insert, then inserts them above the cursor position.
Delete Line	Opens a window to prompt for the number of lines to delete, then deletes them beginning with the line the cursor is on.
Select Line	Allows advanced link processing.
Check Field	When the cursor is positioned on a Source Field Name or Target Field Name, prompts for a file name and then checks the database dictionary of the file to determine whether a field of that name exists. If invalid, allows selection of a valid field.

4. If you make changes, click **Save**.

Data persistence

HP Service Manager requires a relational database management system (RDBMS) for data persistence. See the support web site for a list of supported RDBMS vendors.

You can format and manage the data in your RDBMS from the database dictionary. Service Manager uses the database dictionary to keep a logical version of your RDBMS database dictionary and index.

Database dictionary

HP Service Manager maintains a logical view of your RDBMS tables and columns in the database dictionary. The database dictionary describes each table and column in your system and how they are mapped to logical entities within Service Manager. The Service Manager applications use the logical definitions in the database dictionary to query and manage the actual records in your RDBMS. You must create a database dictionary record for each table and column in your system.

Every time you add tables and columns for Service Manager to your RDBMS system, you must also create database dictionary records describing these new items before Service Manager can make use of them. The best practice HP recommends is to create the database dictionary records first, and then have Service Manager create the back-end tables and columns for you automatically. Some environments, however, do not permit Service Manager to add or update records in the back-end RDBMS.

In restricted-access RDBMS environments, Service Manager can create database definition language (DDL) describing the changes proposed by your database dictionary records. The RDBMS administrator can then create the necessary tables and columns for Service Manager . After the RDBMS has the necessary tables and columns, the Service Manager administrator can then update the database dictionary records to map to the actual RDBMS objects. To simplify the mapping process, Service Manager administrators can also import columns from an RDBMS into a database dictionary record.

You can access database dictionary records either from the System Definition utility or from the database dictionary form. Both utilities require having the `SysAdmin` capability word or its equivalent. HP recommends using the System Definition utility to create database dictionary records if Service Manager has table create-alter-drop-rights to the RDBMS. If your environment restricts access to the RDBMS, HP recommends using the database dictionary form to create the initial null mappings and then update the database dictionary by importing the RDBMS tables and columns.

To provide Service Manager with table create-alter-drop-rights to your RDBMS, you must add the RDBMS connection parameters to the server configuration file (sm.ini). The RDBMS user name you

provide Service Manager must at least have add and drop access rights. HP recommends you grant the RDBMS user owner rights over the database so that Service Manager can manage the data as needed.

To have the server create DDL for database dictionary changes, you must add the `sqlddllogging` and `sqlddldirectory` parameters to the server configuration file (sm.ini). The server creates one DDL file for each logical table in the database dictionary. For example, the file prosummary.ddl contains DDL for changes made to the probsummary database dictionary record, which in turn, maps to the physical tables such as PROSUMMARYA1, PROSUMMARYM1, and PROSUMMARYM2 in your RDBMS. You can enable exporting to DDL independently of providing Service Manager with table create-alter-drop-rights to the RDBMS. You may want to export all RDBMS changes to DDL to keep a log of changes to your RDBMS.

Database dictionary records

Database dictionary records define how RDBMS tables and columns map to logical entities in HP Service Manager. RDBMS tables map to database dictionaries. RDBMS columns map to fields. Database dictionary records also contain information about RDBMS indexes. You must create a database dictionary record for each table and column you add to Service Manager.

IMPORTANT: HP recommends that you refrain from using the following table mapping options for the disposition of arrays:
--*Field in Alias Table
--*BLOB in Alias Table
Using these mapping options can result in potential performance-related issues and unexpected product behavior.

Database dictionary records contain the following information.

Database dictionary object	Brief description	Contents of object	Example database dictionary entry	Example RDBMS equivalent or mapping
Database dictionary name	A logical name for one or more related RDBMS tables	A list of the main and alias tables in the RDBMS that make up the logical table	probsummary – logical table containing incident records	In Microsoft SQL Server, the probsummary database dictionary maps to the following RDBMS tables. • PROBSUMMARYA1 – alias table 1 • PROBSUMMARYM 1 – main table 1 • PROBSUMMARYM

Database dictionary object	Brief description	Contents of object	Example database dictionary entry	Example RDBMS equivalent or mapping
				2 – main table 2
Field name	A logical name for an RDBMS column	A value that Service Manager uses to identify a field. Field names can only contain alphanumeric characters and periods.	category – logical column containing the type of interaction	In Microsoft SQL Server, the category field maps to the SQL name CATEGORY
Field type	An abstract data type describing the information in the RDBMS column	A value that defines the Service Manager data type. Service Manager uses the abstract data type to generate queries, perform calculations, and create SQL mappings.	The field category has the Service Manager data type character	In Microsoft SQL Server, the category field maps to the VARCHAR(50) SQL data type
Field level	A logical data type describing whether a given field is part of a structure or an array of structure	A numerical value that Service Manager uses to determine what level in the logical tree a given field occupies	The field category has the field level of 1	There is no RDBMS equivalent for field level
Field index	A parameter that describes what order a field occupies in the database dictionary record	A numerical value that Service Manager uses to determine the order of a field in the database dictionary record. Service Manager lists fields in the order they are added to the database dictionary. Fields that list the same index number are aliases.	The field category has the field index of 2 and is therefore the second field in the database dictionary record	In Microsoft SQL Server, the CATEGORY column is the second column in the PROBSUMMARYM1 table
Key type	A logical key type describing an RDBMS index	A data type value that Service Manager uses to determine what SQL index type to use when mapping RDBMS index data	The probsummary database dictionary uses the unique key number	In Microsoft SQL Server, the unique key number maps to the unique index NUMBER on the RDBMS column

Database dictionary object	Brief description	Contents of object	Example database dictionary entry	Example RDBMS equivalent or mapping
Key fields	A list of fields that identify the logical key	A list of the fields that make up the key definition	The probsummary database dictionary has a unique key consisting of the number field	In Microsoft SQL Server, the unique index consists of the NUMBER column
SQL table alias	A logical name for a SQL main or alias table	An alphanumeric value that Service Manager uses to identify an RDBMS table	The probsummary database dictionary has three SQL table aliases a1, m1, and m2	In Microsoft SQL Server, the SQL table aliases map to the SQL tables PROBSUMMARYA1, PROBSUMMARYM1, and PROBSUMMARYM2
SQL table name	A listing of the actual RDBMS table names that map to the database dictionary	A RDBMS table name	When mapped to a Microsoft SQL Server, the probsummary database dictionary record lists the SQL table names PROBSUMMARYA1, PROBSUMMARYM1, and PROBSUMMARYM2	In Microsoft SQL Server, the SQL table aliases map to the SQL tables PROBSUMMARYA1, PROBSUMMARYM1, and PROBSUMMARYM2
SQL table type	A description of the RDBMS vendor	A code that identifies the RDBMS vendor	If you are mapped to a Microsoft SQL Server, the SQL type is sqlserver	There is no RDBMS equivalent for SQL type

Making database dictionary changes

Before making database dictionary (dbdict) changes, consider the following behaviors and constraints.

A database dictionary (dbdict) update will fail if:

- A SQL data type is changed and it is not a change from CHAR to VARCHAR or from VARCHAR to CHAR (VARCHAR2 on Oracle).

- A SQL data type is changed and it is not a change from NCHAR to NVARCHAR or from NVARCHAR to NCHAR on Microsoft SQL Server or Oracle (NVARCHAR2).
- A SQL data type is changed and it is not a change from GRAPHIC to VARGRAPHIC or from VARGRAPHIC to GRAPHIC on DB2.
- A SQL data type is changed and it is a change from NCHAR to CHAR or from NVARCHAR to VARCHAR on Microsoft SQL Server.
- A SQL data type CHAR or VARCHAR is shortened.
- ALTER TABLE statements were generated but failed to execute.

A full table copy is triggered for any of the following conditions:

- The RC flag changes for any field.
- The alias changes for any field (but not from or to a NULLTABLE alias).
- A new table is defined for a dbdict.
- The SQL dbtype changed significantly. (For example, either from db2universal to sqlserver, or from one sqlserver2k to sqlserver IF those two SQL dbtypes use different sqldb or sqllogin parameter values.)

An ALTER TABLE statement will be generated to accomplish any of the following changes:

- A SQL data type changed from CHAR to VARCHAR, or vice versa (VARCHAR2 on Oracle).
- A SQL data type changed from NCHAR to NVARCHAR, or vice versa on Microsoft SQL Server.
- A SQL data type changed from VARCHAR to NVARCHAR on Microsoft SQL Server.
- A SQL data type changed from NCHAR to NVARCHAR2, or vice versa on Oracle.
- A SQL data type changed from GRAPHIC to VARGRAPHIC, or vice versa on DB2.
- A SQL data type was widened (for example, from CHAR(10) to CHAR(15)).
- A SQL column was renamed.
- A table was renamed.

On DB2, certain ALTER TABLE operations require a REORG of the affected table. HP Service Manager executes these REORGs, if:

- An ALTER TABLE ... DROP COLUMN statement was executed.
- A SQL column was renamed.
- A SQL type was changed (but not if the change was only widening VARCHAR or VARGRAPHIC columns).

A DROP TABLE statement will be generated if a table entry was completely removed from the SQL Tables tab.

If Service Manager determines it does not have rights to CREATE a TABLE, it will assume not having the rights of performing any of the above operations. In that case it will write all database definition language (DDL) into files in the <RUN directory>/ddl folder and accept changes to the dbdict without performing any of the above operations. It will, however, still verify that the SQL type changes are valid.

Database dictionary data types

Every field in a database dictionary record requires a data type value to determine the format of the data in the field. Database dictionary records use an abstract data type that the HP Service Manager server uses to determine how to map your system data to the SQL data type required by your RDBMS. For example, the database dictionary data type "number" maps to FLOAT in Microsoft SQL server and maps to NUMBER in Oracle. For a complete listing of how the database dictionary data types map to RDBMS data types, use the Database Manager utility to view the records in the `sqldbinfo` table.

Caution: Do not change the data type mappings in the **sqldbinfo** table unless directed to do so by customer support. Changing these values can result in data loss or corruption.

Note: Starting with Service Manager 9.40, the **sqlserver** record in the **sqldbinfo** table contains a read-only flag named **Use Unicode Data Type** to indicate if Microsoft SQL Server uses the Unicode data type (nvarchar).

The Service Manager server does not automatically validate field data to ensure that it matches the data type defined in the database dictionary record. If you want to validate your data prior to submitting it to the RDBMS, you have several options:

- You can customize your forms so that the input field matches the data type. By default, Forms Designer uses the data type to determine the input object to use when you drag a field on to a form.
- You can create a data policy record for the table and then create validation rules for each field in the table you want to validate.

- You can create a format control record for the table and then create validation rules for each particular record state such as add, update, delete, or initial display.

Once you save data to Service Manager, the RDBMS controls the data type and any validation. If you change the data type on the RDBMS side, Service Manager can update the database dictionary record to match the RDBMS data type with a system restart of the server. HP recommends you control data types from the database dictionary so you do not have to restart your system every time you make a change to a field.

Service Manager data types also allow you to take advantage of the following advanced features.

- You can move your system data from one RDBMS type to another
- You can map system data to multiple RDBMS types at the same time

You can use the SQL to SQL Conversion utility to enable one or both of these advanced features. The SQL to SQL Conversion utility allows you to move or map data on a table by table basis. It also allows you to customize how you want data in array fields to be mapped in the new RDBMS as well as offering several export options such as creating DDL or creating duplicate tables.

Changing data types

A System Administrator can change the type of a scalar field. For example, from a character field to a number field.

If no data exists in the file, you can edit the field and change the field type quickly without any other modification required.

If data exists in the file, you must first change the type of the field in the database dictionary record, and then convert the existing data in the modified field to be consistent with the new field type. This must be done for all records in the file that contain data in the modified field.

SQL data type for localizable fields

Consider the following if you know that a certain data field might contain localized data:

- **MS SQL Server:**
  Microsoft's SQL Server does not offer a UTF-8 codepage. Regular data types such as VARCHAR, CHAR or TEXT use single byte code pages (such as MS Windows 1252) or double-byte code pages (such as Shift-JIS), but they cannot hold characters out of multiple regions.
  For example, Shift-JIS allows you to store English and Japanese characters, but not Cyrillic or

Portuguese characters. On the other hand, MS WIN 1251 allows you to store Cyrillic characters but not Japanese.
If you have the requirement to store special characters from many different languages, you should use the SQL Server data types NVARCHAR, NCHAR, or NTEXT instead of VARCHAR, CHAR or TEXT. These data types use UTF-16 as a codepage and therefore are capable of storing special characters from many different languages. Be careful to use these data types only for fields that should be localized, and not for fields that will only use English characters, as they take up twice as much storage space. Also, MS SQL Server has a limitation of 8060 bytes per row in a single table.

- **Oracle and DB2:**
  If you want to store localized data in either DB2 or Oracle, set up your database to use UTF-8 as the codepage. HP Service Manager internally uses UTF-8 as the code page and sends data to the RDBMS clients in UTF-8. By using UTF-8 as the database server's code page, code page translations are not necessary, and VARCHAR, CHAR and CLOB data type can still hold all special characters from all languages.

Change data types: Character to number

Applies to User Roles:

System Administrator

You can change data types by using System Definition, by using Mass update, or by using the Database Dictionary utility.

Change data types by using System Definition

To change data types by using System Definition:

1. In the System Navigator, click **System Definition** > **Tables** >***table_name***> **Fields** > ***field_name***.
2. Double-click the field name.
3. In the General properties tab, select a new type from the **Data type** list.
4. Click **Save**.

Change data types by using Mass Update

In this example, records have a character in the zip field that is now defined as a number field.

If some of the records in the file contain data in the modified field that invalidates the new field type, the data is not converted and the system generates a "cannot evaluate" message for each record failing the conversion.

Service Manager updates only those records that do not fail the conversion.

To change data types by using mass update, use the syntax: `zip in $file=nullsub{val(zip in $file,1),zip in $file}`.

Change data types by using the Database Dictionary utility

This example illustrates how to change the field called **zip** in the **vendor** file from a *character* type field to a *number* type field. Assume that data exists in the file.

Note: You must use the Windows client whenever you need to add or edit a field/key in a database dictionary table.

To change data types by using the Database Dictionary utility:

1. Click **Tailoring** > **Database Dictionary**.
2. Type **vendor** in File Name field, and then click **Search**.
3. Select the **zip** field name.
4. Click **Edit Field/Key**.

   The field editing window opens.
5. In the Type field, choose **number** in the Type list selection.

   This replaces *character* with *number* in the **Type** field.
6. Click **OK** to close the window and commit the change to the file.
7. Click **OK** to update your database dictionary record.

   The message *Record updated in the dbdict file* displays to confirm the update.

   Note: This may not be the end of the process. If records exist in the file, you need to perform a Mass Update to change the field type of the zip field from character to number. (The type of field, for example, character, number, or date/time is stored in the data record with each associated value).

All records containing a value in the field zip need to be updated to reflect the new field type. For additional information on the Mass Update procedure, see the Database Manager Help.

Database dictionary data types

HP Service Manager uses the following data types to describe field data.

Data type	Purpose	Typical RDBMS mapping
Number	Assumes field data consist of numeric data.	Service Manager maps number fields to RDBMS data types reserved for numerical data, such as FLOAT or NUMBER.
Character	Assumes field data consist of alphanumeric characters (numbers, letters, special characters).	Service Manager maps character fields to RDBMS data types reserved for alphanumeric data, such as VARCHAR or VARCHAR2.
Date/time	Assumes field data contain a Service Manager formatted date or time. Service Manager date/time formats include: • MM/DD/YY HH:MM:SS • MM/DD/YYYY HH:MM:SS • DD HH:MM:SS	Service Manager maps date/time fields to RDBMS data types reserved for date/time data, such as TIMESTAMP or DATE.
Logical	Assumes the field data contain one of the following logical values: true, false, NULL or unknown. Logical fields accept yes or no values for input, but the server converts them to true or false values when stored.	Service Manager maps logical fields to RDBMS data types reserved for single character data, such as CHAR.
Array	The field data are compound data elements that contain one or more elements of a single data type or a structure field.	Service Manager maps array fields to RDBMS data types reserved for character large object (CLOB) data, such as TEXT.
Structure	The field data are compound data elements that contain several subfields, each of which may have a different data type. Structures must be child elements of an array field.	Service Manager maps structure fields that are within an array field to RDBMS data types reserved for binary large object (BLOB) data, such as IMAGE.
Expression	The field data contain parsed Rapid Application Development (RAD) language expressions.	Service Manager maps expression fields to RDBMS data types reserved for binary large object (BLOB) data, such as IMAGE.

Data type	Purpose	Typical RDBMS mapping
Label	The field data contain system data used and are not intended for customer data.	Service Manager does not map label fields to any RDBMS data type.
Record	The field data contain system data and are not intended for customer data.	Service Manager does not map record fields to any RDBMS data type.

Arrays

An array tracks multiple occurrences of a given field. The values in the field are all the same field type. For example, you can track all of the components for a printer in the computer file. To accomplish this, create an array field defined as a character type, rather than creating multiple scalar fields. A scalar field is a simple data element comprised of a single occurrence of data (as opposed to an array).

Arrayed structures

An arrayed structure is comprised of more than one field with field type definitions that may or may not be the same. You can define an arrayed structure to your database dictionary record to ensure that, on any form (format) attached to a file, you have the ability to scroll more than one field at a time. When the cursor is in a scrolling field, associated information in one or more other fields will also scroll. For example, you can track contact names and phone numbers in a Configuration Item (CI) file. When scrolling the array of contact names, you can also designate the array of associated telephone numbers to scroll.

Caution: You cannot use the SQL Query method against fields contained within an arrayed structure.

Structures

Structures are used when you have a common group of fields that you want displayed on multiple formats. For example, Incident Management has the following three structures defined in the problem database dictionary record:

- Header
- Middle
- Action

The fields defined in the **header** and **action** structures are designed for use in every problem format, regardless of the category. Instead of using the same input fields for each category format, specific formats are created and then invoked as a subformat on each category dialog. The fields defined on a given subformat may be of different **type** definitions.

Note: On the primary format, the input field defined as a structure is associated with the subformat containing some or all of the fields defined within that structure. Fields defined on the given subformat must be defined within the associated structure, but not all fields defined within the associated structure need to be contained on the given subformat.

Create a file called **employee** with many formats attached to the file (many different views depending on the user profile). A common group of these fields is displayed on all associated formats. To avoid redundant typing of the same fields on all formats, define a structure in your database dictionary record and create a separate format to contain this common group of fields.

Fields within structures

It is important to define fields within structures when these fields exist on a format(s) associated with the input field structure name. To define fields within structures, your cursor (focus) must be on the structure name in the database dictionary record. If fields do not need to be defined within a given structure in a database dictionary record, the cursor should be positioned on the descriptor structure when adding the fields.

Organization of arrays of structures in the database dictionary

In each database dictionary, an array of structures appears in a group. The group is organized hierarchically, with the field of type **array** listed first. The field of type **structure** is listed next, and is indented once from the field of type array. The remaining fields are indented and listed below the field of type structure, and can be of types Character, Number, Location, or Date/Time.

Example: Arrayed structure

An example of an arrayed structure is an unavailability graph. This structure is defined in the standard Configuration Item (CI) files distributed with the Configuration Management application.

To create your own CI files and track availability, add the following availability fields to the CI files:

Database Dictionary field name	Field type
number.of.problems	number
explicit.unavailability	date/time
implicit.unavailability	date/time

Database Dictionary field name	Field type
perceived.unavailability	date/time
start.of.operating.hours	date/time
graph.of.unavailability	array
graph.of.unavailability	structure
date	date/time
graph	character

The field names must be spelled exactly as they are shown in the above table and defined with the specified field type.

In this example, there is a devicepc file for which you want each record within this file to contain statistical information on availability.

Note: The graph fields must also be added to the CI formats.

Note: You must use the Windows client whenever you need to add or edit a field/key in a database dictionary table.

Add availability fields that pertain to the devicepc file

1. Log on to the Service Manager system with a system administrator account.
2. Click **Tailoring** > **Database Dictionary**.
3. In the File Name field, type **devicepc**, and then click **Search**.
4. Place the cursor on the **descriptor** field, and then click **New Field/Key**.

   The field window opens.
5. In the Name field, type **graph.of.unavailability**.
6. Click the list in the Type field and choose **array**.
7. Click **Add**.

   The message *Enter data type of array's element* displays at the top of the screen and the window refreshes.

The **Name** field still displays graph.of.unavailability, but the **Type** field is blank.

To create an arrayed structure, define the type of array as a structure, such as a character, number, or date and time structure.

Click the list in the Type field and choose **structure**.

8. Click **Add**.

   There are two entries in the database dictionary record for the field **graph.of.unavailability**. The first entry defines the field as an array. The second entry defines the data type for the array. Both entries are listed as **graph.of.unavailability**. The system default, which can be overridden, is to use the same field name for both entries.

   The two entries are added to the bottom of the list of fields in the devicepc database dictionary record.

   Note: The second entry is indented in the above example. This occurs only if the number of fields specified in the record is less than 50.

Associate the new field to the structure

1. Tab to the *second* occurrence of the **graph.of.unavailability** field, the structure type.

   For any field to be part of a structure, the cursor must be positioned on that structure field name before clicking New for the new field to be associated with it.

2. Click **New**.

   The Structure field contains the value **graph.of.unavailability**, which is the current structure to which you are adding fields.

3. In the Name field, type **date**.

4. Click the list in the Type field and choose **date/time**.

5. Click **Add**.

   The date field is displayed under the structure field **graph.of.unavailability**.

6. Select the *second* occurrence of the **graph.of.unavailability** field, the structure type.

Note: For any fields to be part of a structure, the cursor must be positioned on that structure field name before clicking New.

7. Click **New**.

   Notice the Structure field contains the value **graph.of.unavailability**, which is the current structure to which you are adding fields.

8. In the Name field of the window, type **graph**.

9. Click the list in the Type field and choose **character**.

10. Click **Add**.

    The graph field is added as the last field under the structure field **graph.of.unavailability**, and the message *graph field added to the pcdevice file* is displayed.

    In this example, the graph field is behind the window at the bottom of the screen.

When you have completed adding fields to the database dictionary record

1. Click **OK**.

   This closes and automatically updates the file. The message *Record updated in the dbdict file* displays when complete.

2. Press **Back** to end the database dictionary session and return to the main menu.

You have now added all of the required fields comprising the **graph.of.unavailability** structure. You can now edit your database dictionary record again, update it, or cancel without updating.

Note: You must update your database dictionary record when adding a new field or the database dictionary record will remain the same. You must also add new fields to the appropriate formats in Forms Designer.

Create the unique and attribute subtables

Applies to User Roles:

System Administrator

You must use the Windows client whenever you need to add or edit a new field/key in a database dictionary table.

To create the unique and attribute subtables:

1. Name the unique and attribute files.
   a. Search for the dbdict from which you want to create subtables. In this example, the **device** dbdict is used.

   b. Scroll down the **Type** column to locate an array of structures.
   Note: An array of structures is identified by an **Array** field type with a field of the same name.

   c. Type `Structure` directly below the Array field. The **Structure** field name is indented. In this example, the array of structures is **features**.

   d. Locate the **features** field of Type `Structure`.

   e. Insert your cursor in the **Type** column for the features structure field.

   f. Click **Edit Field/Key**.

   g. In the **field.window** dialog box, type a name for the **attribute** file. The standard naming convention for the attribute file is: *dbdict + array + attr*. In this example, the attribute file is named **devicefeaturesattr**.
   Note: No spaces or special characters are allowed in the subtable names.

   h. In the **field.window** dialog box, type a name for the **unique** file. The standard naming convention for the unique file is: *dbdict + array + uniq*. In this example, the unique file is named **devicefeaturesuniq**.
   Note: If the attribute file field is left blank, only the unique subtable will be created.

   i. Click **OK**.

2. Designate unique fields in the array of structure.
   a. Locate the field you want to designate as unique. In this example, the unique field is **feature.id**.

   b. Insert your cursor in the **Type** column for the field you want to designate as unique.

   c. Click **Edit Field/Key**.

   d. Click the **Unique in Structured Array** check box.

e. Click **OK**.

f. Repeat steps a – e for each field you want to designate as unique.

3. From the database dictionary, click **OK**.

HP Service Manager updates the dbdict and creates the unique and attribute subtables from the array of structures.

Name the unique and attribute files

Applies to User Roles:

System Administrator

To name the unique and attribute files:

1. Search for the dbdict from where you want to create subtables.
   In this example, the **device** dbdict is used.
2. Scroll down the **Type** column to locate an array of structures.

   Note: An array of structures is identified by an **Array** field type with a field of the same name.
3. Type `Structure` directly below the Array field. The **Structure** field name is indented.
   In this example, the array of structures is **features**.
4. Locate the **features** field of Type `Structure`.
5. Insert your cursor in the **Type** column for the features structure field.
6. Click **Edit**.
7. In the **field.window** dialog box, type a name for the **attribute** file.

   The standard naming convention for the attribute file is: *dbdict + array + attr*.

   In this example, the attribute file is named **devicefeaturesattr**.

   Note: No spaces or special characters are allowed in the subtable names.
8. In the **field.window** dialog box, type a name for the **unique** file.

   The standard naming convention for the unique file is: *dbdict + array + uniq*.

In this example, the unique file is named **devicefeaturesuniq**.

Note: If the attribute file field is left blank, only the unique subtable will be created.

9. Click **OK**.

View the unique and attribute subtables

Applies to User Roles:

System Administrator

To view the unique and attribute subtables:

1. Click **Tailoring** > **Database Manager**.
2. Search for the dbdict that contains the array of structures from which the subtables were created. The unique and attribute files you named should appear in the record list.
3. Double-click on the file you want to view.

Splitting SQL tables

The HP Service Manager server reads the row size limitation and column number limitation from file "sqldbinfo" and uses the information when making a decision about whether to split SQL tables or not. This applies to all supported database servers.

Note: By default, for SQL Server, all fields are put into M1 and will never be put into M2 or other “M” tables.

Caution: Modifications to the "sqldbinfo" table require a restart of the Service Manager server.

Constraints when disabling the row size limitation on SQL Server

If you want to disable the row size limitation on SQL Server, you need to consider the following constraints:

- The length of individual columns must still fall within the limit of 8,060 bytes for varchar, nvarchar, varbinary, sql_variant, and CLR user-defined type columns. Only their combined lengths can exceed the 8,060-byte row limit of a table.

- The sum of other data type columns, including char and nchar data, must fall within the 8,060-byte row limit. Large object data is also exempt from the 8,060-byte row limit.
- After removing the row size limitation, if a file does have an M2 table, before you manually move all of the fields in M2 into M1, newly added fields will still be put into M2.
- If you want to disable table splitting, you need to manually set the row size limitation in the “sqldbinfo" table to a very large number, which is supported by SQLSERVER 2008 and later with ROW_OVERFLOW_DATA. The Row-Overflow feature is great for allowing occasionally rows longer than 8,060 bytes, but it is not well suited for the majority of rows being over-sized, which can lead to a drop in query performance.

Alias fields

An alias field is a field that has a different name but the same index, level, and type as an existing field. Alias fields are used to streamline file maintenance; force fully keyed queries, such as those in the Incident Management and Service Desk applications; and facilitate virtual joining. Alias fields are only supported for scalar fields of the following data types: number, character, date/time, logical. The functionality is not designed to work for other data types like arrays and structures.

If regular purge and archive tasks are performed on files with different field names for the search criteria, such as `close.time` in the `problem` and `probsummary` files, `msg.time` in the msglog file, and `evtime` in the `eventin` file, you can build an alias in each file called **archive.time** and then perform all archive activity based on archive.time without regard to the file name.

In Service Desk and Incident Management, alias keys are used to ensure that a fully keyed query is performed each time an argument is entered from a record list. For more information see the Incident Management Help.

Source fields are defined as an alias when they must be defined as both a normal link *and* a virtual join; *and* the target file or field is different. For example, in the **devicepc** database dictionary record and the **device.pc** format and link records, the **vj.logical.name** field is the alias of the **logical.name** field.

Access the Database Dictionary utility

Applies to User Roles:

System Administrator

Use one of the following methods to access the Database Dictionary utility:

- To access the Database Dictionary utility from the System Navigator, click **Tailoring** > **Database Dictionary**.
- To access the Database Dictionary utility from the HP Service Manager command line, type `dbdict` on the command line, and then press **Enter**.

Enable DDL logging of database dictionary changes

Applies to User Roles:

System Administrator

You must have administrative access to the server operating system to use this procedure.

To enable DDL logging of database dictionary changes:

1. Log on to the server running HP Service Manager.
2. Stop the Service Manager server.
3. Open the Service Manager initialization file (sm.ini) with a text editor.
4. Add the following parameters to enable DDL logging of database dictionary changes:

   ```
   sqlddllogging:1
   sqlddldirectory:<path>
   ```

   For <path>, specify the relative path to the directory where you want Service Manager to save DDL files. This path is relative to the server's RUN directory.
5. Save the Service Manager initialization file.
6. Restart the Service Manager server.

Open a database dictionary

Applies to User Roles:

System Administrator

You can open a database dictionary (dbdict) from Forms Designer, by using System Definition, or by using the Database Dictionary utility. Choose one of the following methods to open a dbdict.

Open a dbdict by using System Definition

To open a dbdict by using System Definition:

1. From the System Navigator, click **System Definition** > **Tables** > ***<dbdict_name>***.

   Note: The files are listed alphabetically. You can type the first letter of a table name to jump to that alphabetic section. For example, type o to jump to the oncall table.

2. Click the plus (+) next to the file name to open the folder.

3. Click the plus (+) next to the *Fields* or *Keys* folder to open it.

4. Double-click a field or key to edit it.

Open a dbdict when using Forms Designer

To open a database dictionary when using Forms Designer:

1. Open an associated form using Forms Designer. For this example, access the *contacts* form.

2. Select **System Definition** from the More Actions menu. The dbdict for that form opens.

Open a dbdict by using the Database Dictionary utility

This example opens the contacts dbdict.

To open a database dictionary by using the Database Dictionary utility:

1. Click **Tailoring** > **Database Dictionary**.

2. Type the name of the dbdict file you want to edit into the **File Name** field.

   For this example, type: **contacts**.

3. Click **Search**.

   The dbdict definition record(s) with that name opens. If more than one file is displayed, select the correct file to view it.

To see more fields, use the scroll bar on the right of the field names list. To see more keys, use the scroll bar on the right of the keys list.

Copying a database dictionary

You can make copies of your database dictionary for backup purposes or to create another file with similar dictionary definitions.

When you copy a database dictionary, you have the option to copy the record only or the record and all individual data records contained in the file. If you copy the database dictionary and all associated records, data in the original file is inaccessible during the copy process. The amount of time it takes to copy the file and records will vary depending on how many records are in the original file.

Copy a database dictionary and data records

Applies to User Roles:

System Administrator

You can copy a database dictionary by using System Definition or by using the Database Dictionary utility. Choose one of the following methods to copy a database dictionary.

Copy a database dictionary and data records, by using System Definition

To copy a database dictionary and data records, by using System Definition:

1. From the System Navigator, click **System Definition**.
2. Double-click a table.

   An overview of the table opens.
3. In the **Table management** section of the overview, click **Copy the definition and data**.

   A confirmation box opens.
4. In the **New table name** field, type the name of the table, and then click **OK**.

   A message displays, stating that the table definition and data were copied to *new name*.
5. Click **OK**.

Copy a database dictionary and data records, by using the Database Dictionary utility

Example: Make a copy of all the records in the *assignment* dbdict by creating a new file with the same database dictionary structure and containing all records contained in the original file. To copy records from a file by copying the entire file, copy the associated data records as well.

To copy a database dictionary and data records, by using the Database Dictionary utility:

1. Open the assignment dbdict.
2. Open the More Actions menu and click **Copy/Rename**.
3. In the **New Name** field, enter **assignment1**.
   Note: File names must be one word. No blanks and no special delimiters are allowed.
4. Select the **Copy dbdict and data** option to copy the dbdict and all records from the **assignment** file to the **assignment1** file.
5. Click OK to copy the dbdict and data stored in the file.
   Note: The length of time it takes to copy a file with all records will vary depending on how many records exist in the file. Copy a file with all associated records during non-peak operation times.

   The database dictionary for the assignment1 file opens and HP Service Manager displays this message: **Dbdict successfully copied*

   Note: When an asterisk displays in front of a message, it means there are additional messages to display which may have relevant information regarding this operation.

   You can now make changes to the **enduserbackup** database dictionary.
6. Click OK to exit when you have completed modifications to this file.
   Note: You will need to update your database dictionary only if you have added, deleted, or modified fields and/or keys.

Copy a database dictionary only

Applies to User Roles:

System Administrator

You can copy a database dictionary by using System Definition or by using the Database Dictionary utility. Choose one of the following methods to copy a database dictionary.

Copy a database dictionary only, by using System Definition

To copy a database dictionary only, by using System Definition:

1. From the System Navigator, click **System Definition**.
2. Double-click a table.

   An overview of the table opens.
3. In the **Table management** section of the overview, click **Copy the definition**.

   A confirmation box opens.
4. In the **New table name** field, type the name of the table, and then click **OK**.

   A message displays, stating that the table definition was copied to *new name*.
5. Click **OK**.

Copy a database dictionary only, by using the Database Dictionary utility

Example: Make a copy of the database dictionary in the assignment file. In this example, we will copy only the database dictionary, not the data records.

To copy a database dictionary only, by using the Database Dictionary utility:

1. Open the **assignment** file using database dictionary.
2. Open the More Actions menu and click **Copy/Rename**.
3. In the New Name field of the rename screen, type `assignment2`.
   Note: File names must be one word; no blanks and no special delimiters are allowed.
4. Select the **Copy dbdict only** radio button.
5. Click **OK** to begin file copy. The copied dbdict opens. HP Service Manager displays this message: *Dbdict successfully copied*.

Methods for adding database dictionary records

Your database dictionary records should match the tables and columns HP Service Manager manages in the RDBMS. HP recommends that you use database dictionary records as your primary data definition source and allow Service Manager to make changes to RDBMS tables, columns, and indexes as needed. If you provide Service Manager with an RDBMS account with table create-alter-drop-rights, any time you make a change in a database dictionary record, the server automatically makes the necessary changes in the RDBMS.

If you do not provide Service Manager with table create-alter-drop-rights to the RDBMS, an RDBMS administrator must make all the necessary changes each time you add, remove, or update a database dictionary record. In such cases, HP recommends you enable DDL logging to capture database dictionary changes. The Service Manager administrator can then provide the DDL to the RDBMS administrator who can approve or modify it as needed. After the RDBMS administrator has created the necessary tables, columns, and indexes, the Service Manager administrator can import the existing RDBMS columns.

The following table summarizes the methods you can use to add database dictionary records and the best practices HP recommends.

Action	From the System Definition utility	From the Database Dictionary form	From RDBMS utilities
Add a database dictionary record	**Best practice**: Provide Service Manager table create-alter-drop-rights to the RDBMS, and create a new database dictionary record. Service Manager creates the required tables, columns, and indexes in the RDBMS.	• **Best practice**: Provide Service Manager table create-alter-drop-rights to the RDBMS, and create a new database dictionary record. Service Manager creates the required tables, columns, and indexes on the RDBMS. • Enable DDL logging and use the Database Dictionary form to create a new database dictionary record. Have the RDBMS administrator add tables, columns, and indexes based on the DDL, and then use the Database Dictionary form to import the existing RDBMS columns.	You cannot create a Service Manager database dictionary record from RDBMS utilities.
Add a table	**Best practice**: Provide Service Manager table	• **Best practice**: Provide Service Manager table create-alter-	Have the RDBMS administrator add

Action	From the System Definition utility	From the Database Dictionary form	From RDBMS utilities
	create-alter-drop-rights to the RDBMS, and then create a new database dictionary record. Service Manager creates the required table in the RDBMS.	drop-rights to the RDBMS, and create a new database dictionary record. Service Manager creates the required tables in the RDBMS. • Enable DDL logging and use the Database Dictionary form to create a new database dictionary record. Have the RDBMS administrator add a table based on the DDL, and then use the Database Dictionary form to import the existing RDBMS columns.	a table directly to the RDBMS, and then use the Database Dictionary form to import the existing RDBMS columns.
Add a field	**Best practice**: Provide Service Manager table create-alter-drop-rights to the RDBMS, and add a field to a database dictionary record. Service Manager creates the required column in the RDBMS.	• **Best practice**: Provide Service Manager table create-alter-drop-rights to the RDBMS, and use the Database Dictionary form to add a field to a database dictionary record. Service Manager creates the required column in the RDBMS. • Enable DDL logging and use the Database Dictionary form to add a field to a database dictionary record. Have the RDBMS administrator add a column based on the DDL, and then use the Database Dictionary form to import the existing RDBMS columns.	You cannot create a Service Manager field from RDBMS utilities.
Add a column	**Best practice**: Provide Service Manager table create-alter-drop-rights to the RDBMS, and add a field to a database dictionary record. Service Manager creates the required column in the RDBMS.	• **Best practice**: Provide Service Manager table create-alter-drop-rights to the RDBMS, and use the Database Dictionary form to add a field to a database dictionary record. Service Manager creates the required column in the RDBMS. • Enable DDL logging and use the Database Dictionary form to add	Have the RDBMS administrator add a column directly to the RDBMS, and then use the Database Dictionary form to import the existing RDBMS columns.

Action	From the System Definition utility	From the Database Dictionary form	From RDBMS utilities
		a field to a database dictionary record. Have the RDBMS administrator add a column based on the DDL, and then use the Database Dictionary form to import the existing RDBMS columns.	
Add a key	**Best practice**: Provide Service Manager table create-alter-drop-rights to the RDBMS, and add a key to a database dictionary record. Service Manager creates the required index in the RDBMS.	• **Best practice**: Provide Service Manager table create-alter-drop-rights to the RDBMS, and use the Database Dictionary form to add a key to a database dictionary record. Service Manager creates the required index in the RDBMS. • Enable DDL logging and use the Database Dictionary form to add a key to a database dictionary record. Have the RDBMS administrator add an index based on the DDL.	You cannot create a Service Manager key from RDBMS utilities.
Add an index	**Best practice**: Provide Service Manager table create-alter-drop-rights to the RDBMS, and add a key to a database dictionary record. Service Manager creates the required index in the RDBMS.	• **Best practice**: Provide Service Manager table create-alter-drop-rights to the RDBMS, and use the Database Dictionary form to add a key to a database dictionary record. Service Manager creates the required index in the RDBMS. • Enable DDL logging and use the Database Dictionary form to add a key to a database dictionary record. Have the RDBMS administrator add an index based on the DDL.	Have the RDBMS administrator add an index directly to the RDBMS, and then manage it from RDBMS utilities.

Field name requirements

You must follow these rules when creating field names:

- Valid characters include uppercase and lowercase letters of the alphabet, numeric characters 0 through 9, and the period character.
- Field names must begin with a letter of the alphabet.
- Spaces and all other special characters are not permitted.
- Do not use the following HP Service Manager reserved words (uppercase or lowercase):
  - AND
  - BEGIN
  - DO
  - ELSE
  - END
  - F
  - FALSE
  - FOR
  - IF
  - IN
  - ISIN
  - LIKE
  - MOD
  - N
  - NO
  - NOT
  - OR
  - STEP
  - T

- THEN
- TO
- TRUE
- U
- UNKNOWN
- WHILE
- Y
- YES

Create an active database dictionary record

Applies to User Roles:

System Administrator

This procedure assumes the HP Service Manager system has table create-alter-drop-rights to the RDBMS system. If Service Manager does not have table create-alter-drop-rights, your database dictionary record remains inactive and mapped to null tables until you activate it. If your system does not have table create-alter-drop-rights to the RDBMS, HP recommends you enable the option to create DDL to provide to your RDBMS administrator. You can then activate the database dictionary record after your RDBMS administrator has created the tables and columns specified in the DDL.

For Windows clients:

To create an active database dictionary record:

1. Log on to the Service Manager system with a system administrator account.
2. In the System Navigator, click **System Definition** > **Tables**.
3. Right-click **Tables**, and then click **New Table**.
4. Type the **Table name**.
5. Add one field for each RDBMS column you want to add.

6. Add a unique key to the database dictionary record.
7. Click **OK**.

For Web clients:

To create an active database dictionary record:

1. Log on to the Service Manager system with a system administrator account.
2. In the System Navigator, click **Tailoring** > **Database Dictionary**.
3. Type the name of the new table in the **File Name** field, and then click the New icon.
4. Add one field for each RDBMS column you want to add.
5. Add a unique key to the database dictionary record.
6. Click **OK**.

Add a structure to the database dictionary

Applies to User Roles:

System Administrator

You can add a structure to the database dictionary by using System Definition or by using the Database Dictionary utility. Choose one of the following methods to add a structure.

Note: You must use the Windows client whenever you need to add a new field/key to a database dictionary table.

Add a structure by using System Definition

To add a structure by using System Definition:

1. In the System Navigator, click **System Definition** > **Tables** > ***filename*** > **Fields**.
2. Click **New Field**.
3. Type a name in the **Field Name** field.
4. Click **OK**. The new field name is now listed in its alphabetical place in the fields list.

5. Select the new field from the list, and in the **General properties** section select **Structure** in the **Data type** field, and then specify other general properties and SQL mappings properties for the field.
6. Select the structure field from the Fields list, and click **New Field**.
7. Type a name in the **Field name** field, and click **OK**. The field is added to the structure.
8. Specify general properties, aliases, and SQL mapping properties for the field.
9. Repeat steps 6 through 8 to add more fields to the structure.
10. Click **Save** to save the file.

Add a structure by using the Database Dictionary utility

This example illustrates how to add a structure that contains fields for basic employee information. The fields are name, telephone number, and address. These fields will appear on all formats or views of the employee file. Salary, hire.date, review.date, rehire, termination.date, position, and department will appear only on certain forms.

To add a structure by using the Database Dictionary utility:

1. From the database dictionary, open the **employee** file.
2. On the Fields tab, place the cursor on the **descriptor** structure.
3. Click **New Field/Key**.
4. In the Name field, type **basic.employee.info**.
5. In the Type field, choose **structure** from the Type list.
6. Click **Add**.

   You can now add fields within the structure you just created.
7. In the Name and Type fields, type a name and select a data type for the first field you want to add to the structure. For example, type **person.name** and select **character**.
8. Click **Add**.

   The structure with its first field is added to the Fields tab.

9. On the Fields tab, select the structure field you added, and click **New Field/Key** to add more fields to the structure field. For example, add two more fields named **telephone** (data type: **character**) and **address** (data type: **character**) .
10. Click **OK** to save your edits.
11. Click **SM Alters** when you are asked how you want to update the database dictionary.

    This returns the message: Record updated in the dbdict file.

Add an alias field

Applies to User Roles:

System Administrator

You can add an alias field by using System Definition or by using the Database Dictionary utility. Choose one of the following methods to add an alias field.

Note: You must use the Windows client whenever you need to add or edit a field/key in a database dictionary table.

Add an alias field by using System Definition

1. In the System Navigator, click **System Definition** > **Tables**.

   Note: To expand any table in the record list, type the first letter of a table name to jump to that alphabetic section. For example, type o to jump to the oncall table.

2. Expand **Fields**, and then select a field.

   The Fields and keys definitions for the activity table opens.

3. In the Aliases section, click **New**.
4. In the **Alias name** field, type an alias name.
5. Click **OK**.
6. Click **Save** to save your changes.

Add an alias field by using the Database Dictionary utility

To add an alias field by using the Database Dictionary utility:

1. Click **Tailoring** > **Database Dictionary**.
2. In the File Name field type a file name, and then click **Search**.

   The record opens.
3. On the Fields tab, scroll through the fields and select one.
4. Click **Edit Field/Key**.

   The field window opens.
5. Click **Create Alias**.
6. In the **Name** field, type the name of the alias.
7. Click **OK**.
8. After the record has been updated, click **OK** again.

Add an array

Applies to User Roles:

System Administrator

You can add an array by using System Definition or by using the Database Dictionary utility. Choose one of the following methods to add an array.

Note: You must use the Windows client whenever you need to add a new field/key to a database dictionary table.

Add an array by using System Definition

1. In the System Navigator, click **System Definition** > **Tables**.

   Note: To expand any table in the record list, type the first letter of a table name to jump to that alphabetic section. For example, type o to jump to the oncall table.
2. Double-click **Fields**.

   The Fields and keys definitions for the table opens.
3. Place the cursor on a non-structue field, and click **New array**.

4. Type a name in the **Field Name** field, and click **OK**.

   The array along with an element named after the array is now listed in its alphabetical place in the fields list.

5. Select a data type, and specify other properties for the array field.

6. Click **Save** to save your changes to the table.

Add an array by using the Database Dictionary utility

Example: Add an array to the **unitofmeasure** file.

1. Click **Tailoring** > **Database Dictionary**.

2. In the File Name field, type **unitofmeasure**, and then click **Search**.

3. Place the cursor in the **descriptor** structure.

4. Click **New Field/Key**.

   The field window opens.

5. In the Name field of the field window, type **language**.

   Note: When specifying field names, no blanks are allowed between words. Always use a period as a delimiter.

6. In the Type field, type **array**, or select array from the Type list.

7. Click **Add**.

   Note: HP Service Manager displays this message: *Enter data type of array's element*.

   A new window opens, requesting a data type for the array element you have created.

8. In the Type field, select **character**, and then click **Add** to add the new array element.

9. Use the scroll bar to locate the new array and element at the bottom of the fields list.

   There are two entries in the record for the new field: The first entry defines the field as an array. The second entry defines the data type of the array. Both entries are listed, using the field name

for both entries. However, you can override this if you choose. The two entries are added to the bottom of the list of fields in the database dictionary record. The second entry is also indented from the first entry (this occurs only if the number of fields specified in the database dictionary record is less than 50).

You can now reedit your database dictionary record, update it, or cancel without updating.

Note: You must update your database dictionary record when adding a new field, or the database dictionary record will remain the same. You must also add a new field(s) to the applicable format(s) via Forms Designer. See the Forms Designer Help for additional information about coordinating record changes.

10. Once you are finished adding fields to the database dictionary, click **OK** to close and automatically update the file.

Add fields to a structure

Applies to User Roles:

System Administrator

You can add fields to a structure with the Database Dictionary utility.

Note: You must use the Windows client whenever you need to add a new field/key to a database dictionary table.

To add a field to a structure:

Example: Add a field to the descriptor structure in the device table.

To add fields to a structure by using the Database Dictionary utility:

1. Open a file in the database dictionary, and then position the cursor on the structure **descriptor**.
2. Click **New Field/Key**, and then do the following.
    a. In the **Name** field, type: **logical.example**.
    b. In the Type field, type **character**, or select character from the Type list of valid data types.
3. Click **Add** to add the new field in this structure.

    The field is added to the bottom of the list of fields under the structure.

4. Double-click the new field to provide the remaining SQL information.
5. Click **OK** once more to save your edits and end, or click **Cancel** to quit without recording the edits.

You must also add any new fields to the appropriate formats using Forms Designer to make the field available to users. For additional information, see the Forms Designer Help.

When you add a new field to structures that are blobs such as an image type, you do not need the additional SQL information.

When creating categories in Incident Management, define any additional fields in the newly-created categories within the middle structure in the `problem` dbdict record.

Create a database dictionary record by importing RDBMS columns

Applies to User Roles:

System Administrator

This procedure assumes the HP Service Manager system does not have table create-alter-drop-rights to the RDBMS system, and that you have not already created a database dictionary record for this change. If your system has table create-alter-drop-rights to the RDBMS system, then Service Manager automatically creates any necessary tables, columns, and indexes when you create a database dictionary record. If your Service Manager system does not have RDBMS table create-alter-drop-rights, HP recommends you create the database dictionary record first and provide DDL to the RDBMS administrator to implement the necessary changes. The Service Manager administrator can then activate the database record when the RDBMS tables and columns become available. This procedure provides an alternative means for you to keep your database dictionary records synchronized with an RDBMS.

In order for Service Manager to successfully import columns into a database dictionary record, the RDBMS tables must have the following features:

- The table names must be in uppercase
- The table must be owned by a Service Manager user

To create a database dictionary record by importing RDBMS columns:

1. Have your RDBMS administrator create the new tables and columns you want Service Manager to manage.
2. Click **Tailoring** > **Database Dictionary**.

3. Type the name you want to use for database dictionary record in the **File Name** field, and then click **New**.

   For example: `newtableone`.

   A new blank database dictionary record opens.

4. Click the **SQL tables** tab.

5. In the **Alias** field, type the short name you want fields in this database dictionary record to use to refer to this table. For example, `m1`.

   Note: Alias names are limited to three characters.

6. In the **Name** field, type the name of the RDBMS table to which you want this database dictionary record to map.

   For example: `NEWTABLEONEM1` .

7. In the **Type** field, type one of the following codes to identify the vendor of your RDBMS.
   - `db2` – for supported versions of IBM DB2 databases
   - `oracle` – for supported versions of Oracle databases
   - `sqlserver` – for supported versions of Microsoft SQL Server databases

8. If this database dictionary maps to multiple RDBMS tables, repeat steps 5-7 for each RDBMS table.

9. Click **Import new columns from SQL** to have the Service Manager server log in to the RDBMS and load all the columns from the SQL tables defined for the database dictionary record.

   Service Manager automatically displays new columns as fields in the database dictionary record.

10. Click **OK** to save the database dictionary record.

Designate unique fields in the array of structures

Applies to User Roles:

System Administrator

To designate unique fields in the array of structures:

1. Locate the field you want to designate as unique. In this example, the unique field is **feature.id**.
2. Insert your cursor in the **Type** column for the field you want to designate as unique.
3. Click **Edit**.
4. Click the **Unique in Structured Array** check box.
5. Click **OK**.
6. Repeat steps 1 — 5 for each field you want to designate as unique.

Create an inactive database dictionary record

Applies to User Roles:

System Administrator

This procedure assumes the HP Service Manager system does not have table create-alter-drop-rights to the RDBMS system. If your system does have table create-alter-drop-rights to the RDBMS system, then Service Manager automatically creates any necessary tables, columns, and indexes required by your database dictionary records.

HP recommends you enable the option to create DDL to provide to your RDBMS administrator. You can then activate the database dictionary record after your RDBMS administrator has created the tables and columns specified in the DDL.

Note: You must use the Windows client whenever you need to add a new field/key to a database dictionary table.

To create an inactive database dictionary record:

1. Log in to the Service Manager system with a system administrator account.
2. Click **Tailoring** > **Database Dictionary**.
3. Type the table name in the **File Name** field, and then click **New**.
4. In the Fields tab, click **New Field/Key** to add one field for each RDBMS column you want to add.
5. In the Keys tab, click **New Field/Key** to add a unique key to the database dictionary record.
6. Click **OK**.

   Service Manager creates the database dictionary record and maps each field to a null table alias.

The database dictionary record remains inactive until you activate it and remap the fields to actual RDBMS tables and columns.

Activate a database dictionary record

Applies to User Roles:

System Administrator

This procedure assumes the HP Service Manager system does not have table create-alter-drop-rights to the RDBMS system, and that you have not already created a database dictionary record for this change. If your system has table create-alter-drop-rights to the RDBMS system, then Service Manager automatically creates any necessary tables, columns, and indexes when you create a database dictionary record. If your Service Manager system does not have RDBMS table create-alter-drop-rights, HP recommends you create the database dictionary record first and provide DDL to the RDBMS administrator to implement the necessary changes. The Service Manager administrator can then use this procedure to activate the database record when the RDBMS tables and columns become available.

To activate a database dictionary record:

1. Log on to the Service Manager system with a System Administrator account.
2. Click **Tailoring** > **Database Dictionary**.
3. Type the existing database dictionary record name in the **Filename** field, and then click **Search**.

   The database dictionary record opens.
4. Click the More Actions and choose **Activate Nulltable Rows** to have the Service Manager server log in to the RDBMS and check for new columns in the SQL tables that match the fields in database dictionary record.

   Service Manager automatically displays the new field mappings in the database dictionary record.
5. Click **OK** to save the database dictionary record.

Methods for deleting database dictionary records

Your database dictionary records should match the tables and columns HP Service Manager manages in the RDBMS. HP recommends that you use database dictionary records as your primary data definition source and allow Service Manager to make changes to RDBMS tables, columns, and indexes as needed. If you provide Service Manager with an RDBMS account with table create-alter-drop-rights, any time you

make a change in a database dictionary record, the server automatically makes the necessary changes in the RDBMS.

If you do not provide Service Manager with table create-alter-drop-rights to the RDBMS, an RDBMS administrator must make all the necessary changes each time you add, remove, or update a database dictionary record. In such cases, HP recommends you enable DDL logging to capture database dictionary changes. The Service Manager administrator can then provide the DDL to the RDBMS administrator who can approve or modify it as needed. After the RDBMS administrator has created the necessary tables, columns, and indexes, the Service Manager administrator can import the existing RDBMS columns.

The following table summarizes the methods you can use to delete database dictionary records and the best practices HP recommends.

Action	From the System Definition utility	From the Database Dictionary form	From RDBMS utilities
Delete a database dictionary record	**Best practice**: Provide Service Manager table create-alter-drop-rights to the RDBMS, and delete an existing database dictionary record. Service Manager deletes any tables, columns, and indexes in the RDBMS.	• **Best practice**: Provide Service Manager table create-alter-drop-rights to the RDBMS, and delete an existing database dictionary record. Service Manager deletes the tables, columns, and indexes in the RDBMS. • Enable DDL logging and use the Database Dictionary form to delete an existing database dictionary record. Have the RDBMS administrator delete tables, columns, and indexes based on the DDL.	You cannot delete a Service Manager database dictionary record from RDBMS utilities.
Delete a table	**Best practice**: Provide Service Manager table create-alter-drop-rights to the RDBMS, and delete an existing database dictionary record. Service Manager deletes the table in the RDBMS.	• **Best practice**: Provide Service Manager table create-alter-drop-rights to the RDBMS, and delete an existing database dictionary record. Service Manager deletes the tables in the RDBMS.	Have the RDBMS administrator delete the tables directly in the RDBMS. Service Manager will automatically update any corresponding database dictionary records to map to NULLTABLE the next time the server restarts and there is a query against the changed table.

Action	From the System Definition utility	From the Database Dictionary form	From RDBMS utilities
		• Enable DDL logging and use the Database Dictionary form to delete an existing database dictionary record. Have the RDBMS administrator delete the tables based on the DDL.	
Delete a field	**Best practice**: Provide Service Manager table create-alter-drop-rights to the RDBMS, and delete an existing field in a database dictionary record. Service Manager deletes the column in the RDBMS.	• **Best practice**: Provide Service Manager table create-alter-drop-rights to the RDBMS, and use the Database Dictionary form to delete an existing field in a database dictionary record. Service Manager deletes the column in the RDBMS. • Enable DDL logging and use the Database Dictionary form to delete an existing field in a database dictionary record. Have the RDBMS administrator delete the column based on the DDL.	You cannot delete a Service Manager field from RDBMS utilities.
Delete a column	**Best practice**: Provide Service Manager table create-alter-drop-rights to the RDBMS, and update an existing field in a database dictionary record. Service Manager updates the column in the RDBMS.	• **Best practice**: Provide Service Manager table create-alter-drop-rights to the RDBMS, and use the Database Dictionary form to update an existing field in a database dictionary record. Service Manager updates the column in the RDBMS. • Enable DDL logging and use the Database	Have the RDBMS administrator update the column directly to the RDBMS. Service Manager will automatically update any corresponding database dictionary records the next time the server restarts and there is a query against the changed table.

Action	From the System Definition utility	From the Database Dictionary form	From RDBMS utilities
		Dictionary form to update an existing field in a database dictionary record. Have the RDBMS administrator update the column based on the DDL.	
Update a key	**Best practice**: Provide Service Manager table create-alter-drop-rights to the RDBMS, and delete an existing key in a database dictionary record. Service Manager delete the index in the RDBMS.	• **Best practice**: Provide Service Manager table create-alter-drop-rights to the RDBMS, and use the Database Dictionary form to delete an existing key in a database dictionary record. Service Manager deletes the index in the RDBMS. • Enable DDL logging and use the Database Dictionary form to delete an existing key in a database dictionary record. Have the RDBMS administrator delete the index based on the DDL.	You cannot delete a Service Manager key from RDBMS utilities.
Delete an index	**Best practice**: Provide Service Manager table create-alter-drop-rights to the RDBMS, and delete an existing key in a database dictionary record. Service Manager deletes the index in the RDBMS.	• **Best practice**: Provide Service Manager table create-alter-drop-rights to the RDBMS, and use the Database Dictionary form to delete an existing key in a database dictionary record. Service Manager deletes the index in the RDBMS. • Enable DDL logging and use the Database Dictionary form to delete an existing key in	Have the RDBMS administrator delete an index directly in the RDBMS.

Action	From the System Definition utility	From the Database Dictionary form	From RDBMS utilities
		a database dictionary record. Have the RDBMS administrator delete the index based on the DDL.	

Delete a database dictionary record

Applies to User Roles:

System Administrator

This procedure assumes the HP Service Manager system has table create-alter-drop-rights to the RDBMS system. If Service Manager does not have table create-alter-drop-rights, your database dictionary record changes remain inactive and mapped to null tables until you activate them. If your system does not have table create-alter-drop-rights to the RDBMS, HP recommends you enable the option to create DDL to provide to your RDBMS administrator. You can then activate the database dictionary record changes after your RDBMS administrator has created the tables and columns specified in the DDL.

To delete a database dictionary record:

1. Log in to the Service Manager system with a system administrator account.
2. From the System Navigator, click **System Definition** > **Tables**.
3. Double-click the table name you want to delete.

   Note: Each table name is actually a database dictionary record.

   Note: You can type the first letter of a table name to jump to that alphabetic section. For example, type o to jump to the oncall table.

   An overview of the table opens.
4. In the **Table management** section of the Table tab, click **Delete**.
5. Click **OK** to confirm the deletion.
6. Click **OK** to close the confirmation message.

Remove a field from an existing database dictionary record

Applies to User Roles:

System Administrator

This procedure maps any unwanted fields in your system to NULLTABLE. HP Service Manager does not actually remove data or delete the corresponding columns from the RDBMS.

Important: HP recommends you leave any unwanted columns mapped to NULLTABLE rather than actually deleting them from the RDBMS to preserve any legacy data these columns may still contain and to provide you a means to remap the columns at a later date.

If you choose to remove the column from the RDBMS, Service Manager will recognize the change and update the database dictionary record to map the deleted column to NULLTABLE the next time you restart the server and query the RDBMS table that contained the deleted column.

To remove a field from an existing database dictionary record, add a table nulltable with alias n1, using the dbdict utility:

1. Log in to Service Manager as a System Administrator.
2. Click **Tailoring** > **Database Dictionary**.
3. In the File Name field, enter the required file name, and then click **Search**.
4. When the correct file is displayed, select the **SQL Tables** tab.
5. Add the following line:

Alias	Name	Type
n1	NULLTABLE	sqlserver

6. Click **OK**. A message appears, stating that the record has been updated in the dbdict file.

7. Select the **Fields** tab and double-click the column to be moved to the Null Table. The field window opens.

8. In the SQL Table field, change the entry from **m1** to **n1**.

9. Click **OK** to return to the dbdict record. The 'SQL Table' column value for the changed field is now 'n1.'

10. Click **OK** to save the changes in the specified table. A message appears, stating that the record has been updated in the dbdict file.

When you select the table through the System Definition utility, you will see the change.

Methods for updating database dictionary records

Your database dictionary records should match the tables and columns HP Service Manager manages in the RDBMS. HP recommends that you use database dictionary records as your primary data definition source and allow Service Manager to make changes to RDBMS tables, columns, and indexes as needed. If you provide Service Manager with an RDBMS account with table create-alter-drop-rights, any time you make a change in a database dictionary record, the server automatically makes the necessary changes in the RDBMS.

If you do not provide Service Manager with table create-alter-drop-rights to the RDBMS, an RDBMS administrator must make all the necessary changes each time you add, remove, or update a database dictionary record. In such cases, HP recommends you enable DDL logging to capture database dictionary changes. The Service Manager administrator can then provide the DDL to the RDBMS administrator who can approve or modify it as needed. After the RDBMS administrator has created the necessary tables, columns, and indexes, the Service Manager administrator can import the existing RDBMS columns.

The following table summarizes the methods you can use to update database dictionary records and the best practices HP recommends.

Action	From the System Definition utility	From the Database Dictionary form	From RDBMS utilities
Update a database dictionary record	**Best practice**: Provide Service Manager table create-alter-drop-rights to the RDBMS, and update an existing database dictionary record. Service Manager updates any tables, columns, and indexes in the RDBMS as well as remaps any existing columns.	• **Best practice**: Provide Service Manager table create-alter-drop-rights to the RDBMS, and update an existing database dictionary record. Service Manager updates the changed tables, columns, and indexes in the RDBMS. • Enable DDL logging and use the Database Dictionary form to update an existing database dictionary record. Have the RDBMS administrator update tables, columns, and indexes based on the DDL.	You cannot update a Service Manager database dictionary record from RDBMS utilities.

Action	From the System Definition utility	From the Database Dictionary form	From RDBMS utilities
Update a table	**Best practice**: Provide Service Manager table create-alter-drop-rights to the RDBMS, and update an existing database dictionary record. Service Manager updates the table in the RDBMS.	• **Best practice**: Provide Service Manager table create-alter-drop-rights to the RDBMS, and update an existing database dictionary record. Service Manager updates the tables in the RDBMS. • Enable DDL logging and use the Database Dictionary form to update an existing database dictionary record. Have the RDBMS administrator update the tables based on the DDL.	Have the RDBMS administrator update the tables directly in the RDBMS. Service Manager will automatically update any corresponding database dictionary records the next time the server restarts and there is a query against the changed table.
Update a field	**Best practice**: Provide Service Manager table create-alter-drop-rights to the RDBMS, and update an existing field in a database dictionary record. Service Manager updates the column in the RDBMS.	• **Best practice**: Provide Service Manager table create-alter-drop-rights to the RDBMS, and use the Database Dictionary form to update an existing field in a database dictionary record. Service Manager updates the column in the RDBMS. • Enable DDL logging and use the Database Dictionary form to update an existing field in a database dictionary record. Have the RDBMS administrator update the column based on the DDL.	You cannot update a Service Manager field from RDBMS utilities.
Update a column	**Best practice**: Provide Service Manager table	• **Best practice**: Provide Service Manager table	Have the RDBMS administrator update the

Action	From the System Definition utility	From the Database Dictionary form	From RDBMS utilities
	create-alter-drop-rights to the RDBMS, and update an existing field in a database dictionary record. Service Manager updates the column in the RDBMS.	create-alter-drop-rights to the RDBMS, and use the Database Dictionary form to update an existing field in a database dictionary record. Service Manager updates the column in the RDBMS. • Enable DDL logging and use the Database Dictionary form to update an existing field in a database dictionary record. Have the RDBMS administrator update the column based on the DDL.	column directly to the RDBMS. Service Manager will automatically update any corresponding database dictionary records the next time the server restarts and there is a query against the changed table.
Update a key	**Best practice**: Provide Service Manager table create-alter-drop-rights to the RDBMS, and update an existing key in a database dictionary record. Service Manager updates the index in the RDBMS.	• **Best practice**: Provide Service Manager table create-alter-drop-rights to the RDBMS, and use the Database Dictionary form to update an existing key in a database dictionary record. Service Manager updates the index in the RDBMS. • Enable DDL logging and use the Database Dictionary form to update an existing key in a database dictionary record. Have the RDBMS administrator update the index based on the DDL.	You cannot update a Service Manager key from RDBMS utilities.

Action	From the System Definition utility	From the Database Dictionary form	From RDBMS utilities
Update an index	**Best practice**: Provide Service Manager table create-alter-drop-rights to the RDBMS, and update an existing key in a database dictionary record. Service Manager updates the index in the RDBMS.	• **Best practice**: Provide Service Manager table create-alter-drop-rights to the RDBMS, and use the Database Dictionary form to update an existing key in a database dictionary record. Service Manager updates the index in the RDBMS. • Enable DDL logging and use the Database Dictionary form to update an existing key in a database dictionary record. Have the RDBMS administrator update the index based on the DDL.	Have the RDBMS administrator update an index directly to the RDBMS, and then manage it from RDBMS utilities.

Automatic update of database dictionary records

HP Service Manager can detect changes you make to *existing* columns and data types in your back-end RDBMS and then automatically update the database dictionary to match these changes after a system restart. This mechanism allows the database dictionary to better match the actual columns within the RDBMS. To trigger the automatic update of database dictionary records, you must first restart Service Manager and then make a query to the changed table in the RDBMS. Since several queries are made as part of the log in process, simply logging in may be sufficient to trigger an automatic update. For tables not queried during login, you will need to directly query the changed table in some fashion. You can either search from a form that queries the table or create a query from the Database Manager.

For example, suppose an RDBMS administrator changes the length of the "NUMBER" column in the "PROBSUMMARYM1" table from VARCHAR2(90) to VARCHAR2(100). To pick up the change, the administrator restarts the Service Manager server, logs in, and opens the incident queue form to trigger a query against the probsummary table. This process causes Service Manager to compare the database dictionary record to the RDBMS column definition, recognize the change, and update the database dictionary record to match. Service Manager writes all such changes to the log file. For example:

```
5120( 5872) 02/05/2009 10:57:00 RTE I Change of SQL data type for field 'number'
from 'VARCHAR2(90)'
to 'VARCHAR2(100)' detected has been saved to DBDICT for file 'probsummary'
```

Note: Service Manager can only change existing entries in database dictionary records, which are stored in the dbdict table. If your RDBMS administrator adds new columns intended for Service Manager to the RDBMS, the Service Manager administrator will have to manually add entries to the database dictionary to reflect those changes. If your RDBMS administrator removes columns managed by Service Manager, then Service Manager updates the database dictionary to map the file to a null table.

Note: In the above example, the "number" field is the unique key of the "probsummary" database dictionary, which maps to multiple tables in the out-of-box system. In an Oracle back-end RDBMS for example, the "probsummary" database dictionary maps to the "probsummarym1" and "probsummarya1" RDBMS tables. If you change the SQL data type of a unique key that is shared across multiple tables, be sure to make the same change in all associated tables. In our example above, the RDBMS administrator should also change the "number" column of the "probsummary **a** 1" table from VARCHAR2(90) to VARCHAR2(100). The Service Manager server displays a warning in the log file when it detects a difference between unique key definitions. For example:

```
4784( 4452) 02/05/2009 10:58:34 RTE W getSqldas: SQL data type for column 'NUMBER'
in alias table a1
(VARCHAR2(90)) does not match SQL data type in main table (VARCHAR2(100))
```

Automatic update of RDBMS data when database dictionary records change

The HP Service Manager server can push changes you make in the database dictionary to your back-end RDBMS. If you provide the Service Manager server with table create-alter-drop-rights to the RDBMS, it automatically adds or updates any tables, columns, or indexes needed in the RDBMS. If you do not provide the Service Manager server table create-alter-drop-rights to the RDBMS, HP recommends you enable exporting your database dictionary changes as DDL. This allows you to make the necessary changes to the database dictionary and then have your RDBMS administrator review and implement the DDL on the RDBMS.

To trigger the automatic update of RDBMS data, you add or update a database dictionary record and save it. When you save the database dictionary record, Service Manager consults the mapping format for your RDBMS to create the corresponding tables, columns, and indexes required by your addition or change. Service Manager then connects to the RDBMS with the user account you provided in the configuration file and attempts to add or update the database. If the add or update is successful, you will see the updated mapping for the table, column, or index after you restart Service Manager. If the add or update fails, the Service Manager server changes the database dictionary mappings to map to null tables and writes an error message in the Service Manager log file.

For example, suppose the Service Manager administrator creates a new database dictionary called `testtable`. The testtable database dictionary consists of two fields: a number field called `test.id` and a character field called `test.name`. The administrator makes the test.id field the unique key. After saving the new database dictionary record, the administrator views the database dictionary record to see how they are mapped to tables and columns in the RDBMS. On a Microsoft SQL Server, the testtable database dictionary might map to the following tables and columns.

Sample database dictionary for the test.id number field

Database dictionary object	Value if RDBMS table create-alter-drop-rights granted	Value if RDBMS table create-alter-drop-rights denied
Database dictionary field name	test.id	test.id
Database dictionary data type	number	number
SQL column name	TEST_ID	TEST_ID
SQL data type	FLOAT	FLOAT
SQL alias	m1	N-m1
SQL table name	TESTTABLEM1	NULLTABLE

Sample database dictionary for the test.name character field

Database dictionary object	Value if RDBMS table create-alter-drop-rights granted	Value if RDBMS table create-alter-drop-rights denied
Database dictionary field name	test.name	test.name
Database dictionary data type	character	character
SQL column name	TEST_NAME	TEST_NAME
SQL data type	VARCHAR(60)	VARCHAR(60)
SQL alias	m1	N-m1
SQL table name	TESTTABLEM1	NULLTABLE

When Service Manager has table create-alter-drop-rights to the RDBMS, the database dictionary record contains mapping information to the actual tables and column names. When Service Manager does not

have table create-alter-drop-rights to the RDBMS, the database dictionary record contains mapping information to the proposed table and column names. The RDBMS administrator can use Service Manager's generated DDL to create the proposed table and column names. Until the actual tables and columns exist, the database dictionary record points to a null table mapping. All null table aliases start with an "N-" prefix, and the SQL table name maps to the value NULLTABLE. After the RDBMS administrator has created the tables and columns for the database dictionary record, the Service Manager administrator can use the **Activate Nulltable Rows** option to update the SQL mappings to their new values. Service Manager will use the column names listed in the database dictionary record to match it to the proper table name in the RDBMS.

Activate a database dictionary record

Applies to User Roles:

System Administrator

This procedure assumes the HP Service Manager system does not have table create-alter-drop-rights to the RDBMS system, and that you have not already created a database dictionary record for this change. If your system has table create-alter-drop-rights to the RDBMS system, then Service Manager automatically creates any necessary tables, columns, and indexes when you create a database dictionary record. If your Service Manager system does not have RDBMS table create-alter-drop-rights, HP recommends you create the database dictionary record first and provide DDL to the RDBMS administrator to implement the necessary changes. The Service Manager administrator can then use this procedure to activate the database record when the RDBMS tables and columns become available.

To activate a database dictionary record:

1. Log on to the Service Manager system with a System Administrator account.
2. Click **Tailoring** > **Database Dictionary**.
3. Type the existing database dictionary record name in the **Filename** field, and then click **Search**.

   The database dictionary record opens.
4. Click the More Actions and choose **Activate Nulltable Rows** to have the Service Manager server log in to the RDBMS and check for new columns in the SQL tables that match the fields in database dictionary record.

   Service Manager automatically displays the new field mappings in the database dictionary record.
5. Click **OK** to save the database dictionary record.

Add a field to an existing database dictionary record

Applies to User Roles:

System Administrator

This procedure assumes the HP Service Manager system has table create-alter-drop-rights to the RDBMS system. If the system does not have table create-alter-drop-rights, your database dictionary record remains inactive and mapped to null tables until you activate it. If your system does not have table create-alter-drop-rights to the RDBMS, HP recommends you enable the option to create DDL to provide rights to your RDBMS administrator. You can then activate the database dictionary record after your RDBMS administrator has created the tables and columns specified in the DDL.

Note: If you choose to add a new column directly from the RDBMS, Service Manager will not automatically recognize the new column. You must import the new column into the database dictionary record before Service Manager can manage it.

To add a field to an existing database dictionary record:

1. Log in to the Service Manager system with a system administrator account.
2. Click **System Definition** > **Tables**.
3. Double-click the table name you want to edit.
   Note: Each table name is actually a database dictionary record.
4. Click the **Fields and Keys** tab.
5. In the Fields and keys definitions for the table, do one of the following:
   - Click **New Field** to create a scalar field
   - Click **New Array** to create an array or an array of structure
6. Type a name for the new field.
7. Click **OK**.

   The field name appears in the list of fields.
8. Select a value from the Data Type field.

   Note: Each field must have a data type.

9. Add any optional field properties, such as data policies or editing rules.
10. Click **Save**..

    Service Manager creates a new column in the RDBMS.

Import RDBMS columns into an existing database dictionary record

Applies to User Roles:

System Administrator

This procedure assumes the HP Service Manager system does not have table create-alter-drop-rights to the RDBMS system, and that you have not already created a database dictionary record for this change. If your system has table create-alter-drop-rights to the RDBMS system, then Service Manager automatically creates any necessary tables, columns, and indexes when you create a database dictionary record. If your Service Manager system does not have RDBMS table create-alter-drop-rights, HP recommends you create the database dictionary record first and provide DDL to the RDBMS administrator to implement the necessary changes. The Service Manager administrator can then activate the database record when the RDBMS tables and columns become available. This procedure provides an alternative means for you to keep your database dictionary records synchronized with an RDBMS.

In order for Service Manager to successfully import columns into a database dictionary record, the RDBMS tables must have the following features:

- The table names must be in uppercase
- The table must be owned by a Service Manager user

To import RDBMS columns into an existing database dictionary record:

1. Log in to the Service Manager system with a system administrator account.
2. Click **Tailoring** > **Database Dictionary**.
3. Type the existing database dictionary record name in the **Filename** field, and then click **Search**. The database dictionary record opens.
4. Click **Import new columns from SQL** to have the Service Manager server log in to the RDBMS and check for new columns in the SQL tables defined for the database dictionary record.

Service Manager automatically displays new columns as fields in the database dictionary record.

Caution: Service Manager only supports importing new columns from main tables.

5. Click **OK** to save the database dictionary record.

Modify field types: Character (scalar) to array

Applies to User Roles:

System Administrator

Example: Change a field with existing data, called address, from a character type scalar field to an array. For this example, assume that we created a new file called vendorview from the format vendor.

To modify field types, character (scalar) to array:

1. Select the **address** field name. Only one entry for **address** should appear, as it is currently a scalar field.
2. Click **Edit**.
3. In the **Name** field, type: `old.address`.
4. Leave the **Type** field value as *character*.
5. Click **OK** to close the window and commit the change to the file.
   The address field name is now old.address in the Field Name array.
6. Select the **descriptor** structure, and then click **New**.
7. In the **Name** field, type: `address`.
8. In the **Type** field, type:`array`.
9. Click **Add** to commit the new array to the file.
   When adding an array field, you also need to specify the type of array (for example, character, number, date/time, etc.). Service Manager prompts you for this information now.
   A new **field.window** opens with **address** displayed in the **Name** field and the **Type** field blank. The message *Enter data type of array's element* also appears at the bottom of the format window.
10. Select the **Type** field and type:`character`.

11. Click **Add**.
    Notice two entries for address now exist in the database dictionary record since it has been defined as a character array.
12. Click **OK** to close the window and save the changes.
13. Click OK to end this session with the vendorview file.

The field type has been changed, but this is not the end of the process.

Because data exists in the file, you need to move the contents of the field now called **old.address** to the field called **address**. To accomplish this, perform a **Mass Update** in Database Manager on all records in the file that have data in the old.address field.

Once the data has been moved, HP recommends that you also remove the **old.address** field from the database dictionary record.

The Mass Update instruction string to move the data from the field called **old.address** to the first element in the newly created array called **address** is:

```
1 in address in $file = old.address in $file
```

Modify field types: Single to concatenated

Applies to User Roles:

System Administrator

Example Modify the single key *wdManagerName* in a copy of the **assignment** database dictionary record to make it a concatenated key comprised of the fields *wdManagerName* and *company*.

To modify field types: single to concatenated:

1. Open the assignment dbdict.
2. Make a copy of the assignment dbdict named *assignment2*.
3. Position the cursor on the key type Nulls & Duplicates, above the key model.
4. Click **Edit**.
5. Add company in the second element of the field names array below *wdManagerName*.
6. Click **OK** to close the window.
7. You can now re-edit your database dictionary record, update it, or cancel without updating.

8. Click **OK**, to close and automatically update the file. This returns the message Record updated in the dbdict file.

9. When an IR key is added or updated, then Service Manager prompts you to regenerate the file. You can either click **OK** (regen file) to continue with the regen, click **Cancel** (F3) to abort the regen and restore the database dictionary record to the version prior to the latest changes, or schedule a time for the file regeneration, using Schedule (clock button).

Regen a database dictionary

Applies to User Roles:

System Administrator

You can regenerate an IR index in the database dictionary by using System Definition or by using the Database Dictionary utility.

Choose one of the following methods to regenerate an IR index.

Regenerate an IR index using System Definition

To regenerate an IR index using System Definition:

1. From the System Navigator, click **System Definition** > **Tables**.

2. Double-click the table containing the IR index you want to regenerate. The overview of the table opens.

3. In the **Table management** section, click **Regenerate IR Index**.

   The Regen IR Index window opens.

4. Specify the schedule parameters, as follows:
   - In the **When do you want regen to occur?** field, select whether to start `Now` or `Later`. If you select later, enter when the `mm/dd/yyyy hh:mm:ss` format.

   - In the **Repeat Interval** field, select an option if the database needs to be regenerated on a regular basis. This setting is optional. The interval period starts from the time and date set in the "When do you want regen to occur?" field.
     - Monthly – regenerates once a month.

     - Quarterly – regenerates every three months.

- Semi-annually – regenerates every six months.
- Annually – regenerates once a year.
- Other – regenerates in the specified number of days, at the specified time.

  Use the dd hh:mm:ss format, where dd is the number of days from the initial date and time, and hh:mm:ss is the time of day at which the regen is run. For example, 26 10:00:00 sets the regen to run at 10 a.m., 26 days from the initial regeneration set in the Date/Time to run field.

5. Click **OK**.

Regenerate an IR index using the Database Dictionary utility

Using the Database Dictionary utility to regenerate an IR index enables you to schedule the regen.

To regenerate an IR index using the Database Dictionary utility.

1. Open the database dictionary.
2. Open the More Actions menu and choose **Regen IR**.

   Note: This option does not appear for files that contain no data records.

   You are prompted to confirm this regen action.

3. From the prompt, do one of the following:
   - Click **OK** to confirm this regen and erase all records in this file.
   - Click **Cancel** to quit and return to the database dictionary.
   - Click **Schedule** (calendar) to schedule the time to begin Regen.
     - In the **Date/Time to run** field, provide a date in the DD/MM/YYYY format and time in the HH:MM:SS format.
     - If you choose the optional **Repeat Interval** option to regenerate the database on a regular basis, the interval period starts from the `Date/Time to run` you just set.
       - Monthly – regenerates once a month.
       - Quarterly – regenerates every three months.

- Semi-annually – regenerates every six months.
- Annually – regenerates once a year.
- Other – regenerates in the specified number of days, at the specified time.

  Use the dd hh:mm:ss format, where dd is the number of days from the initial date and time, and hh:mm:ss is the time of day at which the regen is run. For example, 26 10:00:00 sets the regen to run at 10 a.m., 26 days from the initial regeneration set in the Date/Time to run field.

- Click **Schedule**. to confirm this database regeneration action.

4. If you clicked **OK**, the regen performs immediately in the foreground. When completed, Service Manager displays a message confirming the time/date of the regen and the removal of all records.

   Note: The asterisk (*) at the beginning of the message indicates there are additional messages related to this operation.

5. View all messages.
6. Read the messages and identify any errors that occurred during the operation.

Note: When you run an IR Regen, Service Manager needs to clear the current indexes. This causes a delete to the backend database. When Service Manager attempts to delete a large set of records, the database could run out of space for this activity and cause an error. Each supported database platform provides some type of transaction or undo log to back up data changes and allow a rollback in case of errors. If you encounter such an error, please contact your database administrator and ask them to increase the size available for this purpose.

Some examples of the errors you could encounter are:

Oracle:SQL code=30036 message=ORA-30036: unable to extend segment by % in undo tablespace

SQL Server SQL State: 42000-9002 Message: [Microsoft][SQL Native Client][SQL Server]The transaction log for database '%' is full

DB2 SQL State: 57011--964 Message: [IBM][CLI Driver][DB2/NT] SQL0964C The transaction log for the database is full. SQLSTATE=57011

Search for a field in the database dictionary

Applies to User Roles:

System Administrator

The ability to search the dictionary for a specific field name is particularly useful when dealing with a large database dictionary.

You can search for a field in the database dictionary by using System Definition or by using the Database Dictionary utility. Choose one of the following methods to search for fields.

Search for fields by using System Definition

To search for a field by using System Definition:

1. From the System Navigator, click **System Definition** > **Tables** > ***file_name*** > **Fields**

   Note: You can type the first letter of a table name to jump to that alphabetic section. For example, type o to jump to the oncall table. Double-click a table to open it. Double-click the table to open it.

2. Type the first letter(s) of the field you are looking for. The navigator will go to the first field starting with those letters.
   Note: If there is a delay between the letters you type, the Navigator will assume that you want a different search and jump out of the folder to find the matching letters.

Search for fields by using the Database Dictionary utility

Using the Database Dictionary utility when searching for fields enables you do a “contains” search.

To search for a field by using the Database Dictionary utility:

1. Click **Tailoring** > **Database Dictionary**, and then open the database dictionary file.

2. From within the database dictionary record you want to search, open the More Actions menu and click **Field Search**.

   This brings up the Field Search window prompt.

3. Type the field name to search, for example, **one**.
   - Click **Exact match** if you are sure that what you entered is a field in the dictionary record.
   - Click **Starts with** to find all fields starting with the string entered.
   - Click **Contains** to search for fields containing the string entered anywhere in the field name.

(Used in this example.)

- Click **End** to cancel the search.

4. If Service Manager finds a match, it responds by giving a message similar to the following: **Field "phone" found: type=character; level=3; index=2*.

   The message gives the field type, the level (how many structures down the field where it is located), and the index (location within its parent structure).

   For example, **one** appears in:

   - A character type field (phone)
   - In the third level structure (alternate.contact)
   - As the second field in the alternate.contact structure.

5. If multiple fields are found by the search, multiple messages are returned, one message for each field found. The message in the above example displays an asterisk (*), indicating it does have additional messages to display.

6. View all messages related to this operation.

   In this example, these messages reveal that **one** is present in at least two fields in this database dictionary.

   However, since the **Contains** button was used, the letters are found in their specific order, regardless of where they appear in the field name.

   For an exact match on the word **one**, you can use the Exact match or Starts with button. (Using these options against this example returns no positive hits using Exact match or Starts with.)

Update an existing database dictionary record

Applies to User Roles:

System Administrator

This procedure assumes the HP Service Manager system has table create-alter-drop-rights to the RDBMS system. If Service Manager does not have table create-alter-drop-rights, your database dictionary record changes remain inactive and mapped to null tables until you activate them. If your system does not have table create-alter-drop-rights to the RDBMS, HP recommends you enable the option to create DDL to provide to your RDBMS administrator. You can then activate the database

dictionary record changes after your RDBMS administrator has created the tables and columns specified in the DDL.

To update an existing database dictionary record:

1. Log in to the Service Manager system with a system administrator account.
2. Click **System Definition** > **Tables**.
3. Double-click the table name you want to edit.

   Note: Each table name is actually a database dictionary record.
4. From the **Table** tab you can delete, rename, or copy the database dictionary record as well as add triggers.
5. From the **Fields and Keys** tab you can add, delete, or update fields and keys.
6. Make the changes you want to make,, and then click **Save** to update the database dictionary record.

Update the data type of a database dictionary field

Applies to User Roles:

System Administrator

This procedure assumes the HP Service Manager system has table create-alter-drop-rights to the RDBMS system. If Service Manager does not have table create-alter-drop-rights, your database dictionary record changes remains inactive and mapped to null tables until you activate it. If your system does not have table create-alter-drop-rights to the RDBMS, HP recommends you enable the option to create DDL to provide to your RDBMS administrator. You can then activate the database dictionary record changes after your RDBMS administrator has created the tables and columns specified in the DDL.

To update the data type of a database dictionary field:

1. Log in to the Service Manager system with a system administrator account.
2. Click **System Definition** > **Tables**.
3. Double-click the table name you want to edit.

   Note: Each table name is actually a database dictionary record.

4. Click the **Fields and Keys** tab.
5. Click the field name you want to edit.
   The General properties pane opens.
6. Select a new value from the **Data type** field.

   A confirmation window opens.

   Note: You cannot use the drop-down list to change a scalar field to an array or structure. Instead, you must add a new array or structure.
7. Click **OK** to verify the data type change.
8. Click **Save** to update the database dictionary record.

Renaming a database dictionary

After you create a database dictionary, you can do one of the following:

- Change the name of the dictionary.
- Change the name of one or more of the files sent with the base system.

For example, you can rename the *communications* file to *interactions*. This changes the file name but retains all of the field and key definitions in the dictionary.

If records exist in the file, renaming the dictionary associates all of the records with the new dictionary name. This differs from copying a file because it takes only a few seconds to perform this function.

Rename a database dictionary

Applies to User Roles:

System Administrator

Database dictionary names must be one word. No blanks and no special delimiters are allowed.

You can rename a database dictionary by using System Definition or by using the Database Dictionary utility. Choose one of the following methods to rename a database dictionary.

Rename a database dictionary by using System Definition

To rename a database dictionary using System Definition:

1. On the System Navigation, click **System Definition** > **Tables** > ***file_name***.

   Note: You can type the first letter of a table name to jump to that alphabetic section. For example, type o to jump to the oncall table. Double-click a table to open it.

2. Click **Rename**.

3. Type the desired file name in the **Rename To** text box.

4. Click **OK**. HP Service Manager displays the following message: *File renamed from <old name> to <new name>*.

Rename a database dictionary by using the Database Dictionary utility

To rename a database dictionary using the Database Dictionary utility:

1. Open the old dbdict using the Database Dictionary utility.

   Click **Tailoring** > **Database Dictionary**.

2. Open the More Actions menu and click **Copy/Rename**.

3. Type the desired file name in the **New Name** text box.

4. Select the **Rename** option.

5. Click **OK**. The renamed database dictionary opens with the following message in the status bar: *Dbdict successfully renamed*.

6. Click **OK** to exit the record and save the file with the new name.

Resetting a database table

In some scenarios, you may need to reset a database table to remove all records. For example, when moving a table from a development environment to a production environment, you may want to start with an empty production table. You may also need to limit the size of files that continually increase in size, such as syslog or msglog.

Warning: Use caution when choosing this option, as resetting removes all records from the table.

Resetting the database and all records

If the records in a dictionary are no longer needed but the file structure is to be retained for later use, perform a file reset.

The **Reset** option deletes all data records from a dictionary. In most instances, once a system is set up and is in operation, this option is not used. Most often it is utilized when clearing test data from a preimplemented system or test file.

Warning: You should only use this option when you intend to permanently remove data file records. Once a reset is performed, the records cannot be retrieved.

Reset a database table from Database Manager

Applies to User Roles:

System Administrator

In this example we reset the syslog table. Resetting other tables works the same way.

To reset a database table from Database Manager:

1. Open the syslog form in Database Manager.
2. Click **More** or the More Actions icon.
3. Select **Reset**.
   Service Manager displays a prompt, asking you to confirm the action and allowing you to schedule the Reset.
4. From the prompt, you can:
   - Click **OK** to reset the syslog table.
   - Click **Cancel** to leave the table intact and return to the blank syslog format.
   - Click **Schedule** to schedule a time to run the file reset operation, either once or repeatedly at a set interval.

Schedule a reset

Applies to User Roles:

System Administrator

If you chose to schedule a reset by clicking the Schedule button on the reset prompt you are prompted with a schedule dialog, Service Manager displays a schedule form.

To schedule a reset:

1. Provide a date in a DD/MM/YYYY format in the Date/Time to run field.
2. Provide a time in an HH:MM:SS format the Date/Time to run field.
3. Select a Repeat Interval option to reset the file on a regular basis. This setting is optional. The interval period starts from the time and date set in steps 2 and 3.
   - Monthly – reset once a month.
   - Quarterly – reset every three months.
   - Semi-annually – reset every six months.
   - Annually – reset once a year.
   - Other – reset in the specified number of days, at the specified time.

     Use the dd hh:mm:ss format, where dd is the number of days from the initial date and time, and hh:mm:ss is the time of day. For example, 26 10:00:00 schedules the reset to run at 10 a.m., 26 days from the initial reset date and time set in the Date/Time to run field.
4. Click **Schedule** to confirm the reset action.

Keys and indexes in HP Service Manager

HP Service Manager keys are abstract entities that provide a logical view of the indexes in your RDBMS. When you create Service Manager logical keys, the server creates corresponding indexes on the back-end RDBMS. The benefits of using Service Manager logical keys are:

- They allow administrators a means to manage and move indexes from one environment to another (for example, from a test environment to a production environment)
- They allow administrators a means to move indexes from one RDBMS type to another (for example, from a SQL Server Express edition demonstration RDBMS to a supported RDBMS in a development environment)

In RDBMS terms, Service Manager keys provide constraints on column data as well as an index of the records in a table that improves query performance. From the Service Manager side, logical key definitions are one part of a database dictionary record. The full database dictionary record also

includes field and column mappings. Typically, you will use database dictionary records to manage and move both logical keys and field mappings at the same time.

Currently, Service Manager can only push keys as indexes on an RDBMS. It cannot read existing indexes (pull) from the RDBMS and create the corresponding logical keys. Service Manager can still use existing indexes on an RDBMS without the indexes being defined as logical keys. Any query issued against an indexed column still gets a performance benefit. However you can only take advantage of the features of logical keys if you create them from Service Manager.

Service Manager uses the following logic to create indexes:

Service Manager key type	Type of index created on the RDBMS
Primary	Primary
Unique	Index with unique constraint
No Nulls	Index
No Duplicates	Index
Nulls and Duplicates	Index
IR index	None Service Manager directly manages IR Index keys

You can define Service Manager keys from either the System Definition utility or the Table Definition utility.

Key type definitions

Key type definitions allow you to specify the type of key and the names of the fields within the key. There may be more than one field in one key. A key with multiple fields is called a concatenated key.

Key Type	Description
Primary	No fields in the key can be null. The value of the complete key cannot be null and must be unique in the index. A table can have at most one primary key.
Unique	At least one field in the key must not be null. The value of the complete key must be unique in the index.
No Nulls	At least one field in the key must not be null.
No Duplicates	The value of the complete key must be unique in the index or the values of all fields must be null.
Nulls &	All fields can be null. The complete key value can be in the index more than once.

Key Type	Description
Duplicates	
IR key	The fields in the key are indexed by IR Expert. Only one IR key can be used per dbdict record or IR searches on that file do not work. You can concatenate several fields in an IR key.

Key type attributes

The attributes of Unique keys differ by RDBMS type. For example, in a DB2 environment, the constraints for a Unique key are No Duplicates and No Null. In an Oracle environment, the constraints are No Duplicates and Null. Due to the variety of constraints for each RDBMS, HP Service Manager only creates Unique key constraints on the RDBMS that allows Nulls.

You can choose to modify the Unique constraints to primary keys in your RDBMS. If you choose to do so, this will allow No Nulls and No Duplicates if the key is to contain those values.

Key type attributes:

Key Type	No Duplicates	Allows Duplicates
No Nulls	Unique Primary (If primary key mode is enabled)	No Nulls
Allows Nulls	No Duplicates	Nulls & Duplicates

Note: IR keys are indexed separately and are not a part of the key types listed in this table.

Tuning: Designing keys for queries

Fully keyed queries typically offer the best performance. You can design keys to ensure that every query is fully keyed, although it may not be practical in all instances. Running all queries fully keyed requires defining a large number of keys, which in turn can cause performance degradation when using the add, update, and delete operations. In general, the Service Manager System Administrator should discuss query performance with the Database Administrator and tune the system accordingly.

Points to consider when designing keys:

- Design keys for the most used queries.
- You can optionally force users to issue fully keyed queries by not allowing partially keyed and non-keyed queries.
- Specify the fields that are most commonly used at the beginning of the key in the query.

- Specify fields that have many possible values at the beginning of the key. A key with a Boolean field at the beginning is inefficient, unless the majority of your queries return only a small number of records. For example, flag=true in the probsummary table eliminates 90% of all records.
- Do not use the same field in multiple keys. If you update a record and then change the value of that field, update all of the indexes that contain that key.
- Do not define more than 25 keys for one file. The more keys defined for a file, the more time is required to add, update, and delete records in that file. Likewise, if you define too few keys, the operations run faster but the searches on the individual file run slower.
- There is no specific rule of how many keys you should define for a file. Logical consideration of these facts and your specific work environment are the best decision-making factors.

Tuning: Key selection algorithms

A key selection algorithm selects a key to perform a query based on the order of the fields in the query expression, and the order of the fields in the keys defined in the dbdict. The system then assigns a weight to each key based on the order in which the fields appear in the query for the key, and the order in which they appear in the query and key. For example, if the first field in a key matches the first field in the query, Service Manager assigns that key a higher weight than one that has that field as the second field in the key. The system makes the decision for all fields in the query and keys.

Example query and keys:

```
query: a=1 and b=2 and c=3 and d=4
       key1: b,c,d
       key2: a,c,d
       key3: a,b
       key4: a
```

In this example, the key selection algorithm selects key3. The next highest weight is key2, followed by key4 and key1. The new key selection is based on the location database dictionary record with the keys {location}, {location,state}, {location,city}, {location.name,location}, and {location.code}.

Working with primary keys

In versions of Service Manager earlier than 9.32, Service Manager did not support primary keys or `Not Null` constraints in the RDBMS tables. While the Service Manager logical representations did include support for unique keys with a `Not Null` constraint, this support was managed entirely within Service Manager and not in the RDBMS. Because of this lack of functionality, SQL queries generated by Service Manager added an `Is Null` condition in the `JOIN/WHERE` sub-clause as shown in the following example:

```
SELECT m1."RELATIONSHIP_NAME", m1."LOGICAL_NAME"
FROM CIRELATIONSHIPM1 m1 LEFT OUTER JOIN CIRELATIONSHIPA1 a1
ON
  ( ( (m1."LOGICAL_NAME" = a1."LOGICAL_NAME")
 OR (m1."LOGICAL_NAME" IS NULL AND a1."LOGICAL_NAME" IS NULL) )
AND
    ( (m1."RELATIONSHIP_NAME" = a1."RELATIONSHIP_NAME")
OR (m1."RELATIONSHIP_NAME" IS NULL AND a1."RELATIONSHIP_NAME" IS NULL) ) )

WHERE …
```

This mechanism resulted in poor performance when you consider the speed advantages of leveraging the inherent integrity of the database.

In Service Manager 9.32 or later versions of Service Manager, primary keys and `Not Null` constraints on the unique keys are now supported.

Usage

From command line

The `sm -system_addconstraint` command line is used to add either `NOT NULL` or primary key constraint to the existing files in SM.

Note: For details on how to use the `system_addconstraint` command, see the "Modify keys: Convert the first unique key to primary key" on page 126 and the "Modify keys: Add Not Null constraints to the first unique key in a table" on page 127 topics.

There are three levels of operations you can perform by using this command:

1. Add the NOT NULL constraint only

   While Service Manager already has a `No Nulls` key type, this key type is in fact mapped to an normal index in RDBMS. The `NOT NULL` constraint added by `system_addconstraint` command adds a true constraint in the RDBMS (such as the `NOT NULL` constraint in and Oracle database.

   The `NOT NULL` constraint is invisible to end users; End users cannot add, modify or delete it from anywhere in the Service Manager client.

2. Modify the first unique key to primary key

By using this command line, the Service Manager server can convert the first unique key of the Service Manager files to true primary keys in RDBMS. However, if the data in the table is not qualified for the primary key constraint, the conversion from unique key to primary key will fail.

3. Modify the first unique key to primary key more aggressively

   If you can afford the risk of lost data in the RDBMS, you can use this level of operation. At this level, the `system_addconstraint` converts the first unique key to a primary key more aggressively. That is, a temporary table will be created, all the qualified data will be moved to this new table, and the old table is dropped, and the temporary table is renamed to the original one.

   This process is referred to as a full table copy; unqualified data is lost, and qualified data is left.

From SM client

From the Service Manager client, users can add, modify or remove the primary key type just like any other key type.

Considerations

The following behavior is subjected to the primary key feature:

- When there is data in a table, you cannot modify the name of the primary key column.
- If there is a RECORD_KEY column and one unique key column in a table, the RECORD_KEY column is removed, and the first unique key is converted to a primary key when you run the sm –system_ addconstraint command.
- Service Manager automatically creates a hidden RECORD_KEY field when there is no primary key or unique key in the Database Dictionary file. It is recommended that you explicitly define a primary key when creating a new dbdict record or convert the first unique key to a primary key if you think it necessary for better performance or other reasons.
- Once the database is converted to primary key mode by using the system_addconstraint parameter, the first unique key in the Database Dictionary is changed, and a hidden RECORD_KEY field may or may not have been created accordingly. This behavior could potentially impact any integration you may have implemented in HP Service Manager. For example, the legacy SCENTER integration and ODBC driver may rely on the primary key field directly. If such integration does not work as expected after you apply the primary keys, you should review and modify the integration accordingly.
- The type of a newly-created RECORD_KEY field is now VARCHAR2 or VARCHAR in Service Manager. However, any existing RECORD_KEY fields remain as INTEGER.

Retrieve primary key values

Caution: You can enable primary key mode only after you have upgraded to Service Manager 9.32 applications or later. Otherwise, the following procedures may have unexpected results.

You can use the following example to derive your own scripts to read the primary key values:

1. Start Service Manager and then log in as an administrator.
2. Navigate to dbdict.
3. Create a `testkey` table with a primary key defined.
4. Create a detailed form based on the `testkey` table.
5. Insert several data points into the `testkey` table.
6. Navigate to ScriptLibrary.
7. Create a new script by copying the code below:

```
var retValue;
var retCode;
var keyValues = new SCDatum();
var file = new SCFile("testkey");
file.doSelect( "true" );
file.getFirst();
retValue = system.functions.rtecall("getprimary", retCode, keyValues, file );
print( "file testkey is ", file );
print( "The return value of getprimary is ", retValue );
print( "The value of primary key is ", keyValues );
```

8. Execute the script. You should see output that resembles the following:

```
The value of primary key is  [C++ object SCDatum] - {'12/12/12 00:00:00',
"54343"}
The return value of getprimary is  true
File testkey is  [C++ object SCFile] - testkey={['12/12/12 00:00:00', "fgdf", ,
"54343"]}
```

Retrieve primary key mode

Caution: You can enable primary key mode only after you have upgraded to Service Manager 9.32 applications or later. Otherwise, the following procedures may have unexpected results. If you are running Service Manager 9.34p4 or later (for 9.3x) or 9.40 or later (for 9.4x), the primary key feature is enabled by default.

You can run a script to verify whether Service Manager has the primary key feature enabled.

Application developers

Application developers can retrieve primary key mode information by using the `sysinfo.get ("PKMode")` RAD function. To do this, use the following example:

1. Log in to Service Manager as an administrator.
2. Navigate to **Script Library**.
3. Create a new script named `test`.
4. Copy the following code into the text box.

```
var retValue;
retValue = system.functions.sysinfo_get ("PKMode" );
print ( "Current value of PKMode is " + retValue );
```

5. Run the script.

If the primary key feature is enabled, the following message appears in the Messages pane of the SM client:

```
Current value of PKMode is true
```

Client Developers

Client developers can verify whether the primary key feature is enabled; To do this, use the following example:

1. Log in to Service Manager as an administrator.
2. Navigate **System Navigator** > **System Definition** > **Tables**.
3. Double-click any table, and then click the **Fields and Keys** tab.

4. Click any key in the Keys list.
5. Click the **Type** drop-down list in the General section.

If the primary key feature is enabled, you should see **Primary Key** as an option in the **Type** drop-down list.

Note: The legacy listener does not work when the primary key feature is enabled.

Adding a key

A key is an identifying field in a file, used when queries are processed so all fields in a record do not need to be searched for set criteria. A field or fields in a key must be defined as a field or fields in the database dictionary record. A key may be added in a production system without a system restart.

When a database dictionary file is created through **Forms Designer**, HP Service Manager automatically uses the first field defined on the associated format to create a *Unique* key. Manually modify the database dictionary record to add the necessary keys to support on-line and reporting queries, as well as sort sequences. Many other default system files, set up in the base system, may need to be changed, based on your individual search and reporting requirements. This section details the manual addition and manipulation of database dictionary record/file keys.

Note: If records already exist in the file when you add a key, updating the database dictionary record will cause Service Manager to automatically perform an index regeneration.

If you add keys to a file that contains records, be careful when defining key types. For example, do not define a new key as **No Nulls** (no blank values) if records exist in the file that contain no data in the particular field(s) comprising the key.

Each record not containing a value in the **key** field would require an update, populating the field with data. The updates must be performed prior to adding the No Nulls key in order to satisfy the new key definition.

A CREATE INDEX statement is generated when a new key is defined in the dbdict.

Note: If Service Manager determines it does not have rights to CREATE a TABLE it writes all database definition language (DDL) into files in the `<RUN directory>/ddl` folder and accepts changes to the dbdict without performing the CREATE INDEX operation.

Add a key as the first key

Applies to User Roles:

System Administrator

Example: Add the language.key field as a No Nulls key to the **unitofmeasure** database dictionary record.

In this example, you add the language.key field to the **unitofmeasure** record and assign it as the first key with a property of No Nulls. This field is a required field in the record.

To add a key as the first key:

1. Log in to the Windows client.
2. From the System Navigator, click **System Definition** > **Tables** > **unitofmeasure** > **Keys**.
3. In the Keys section, click **New**.
4. In the General section, click the **Type** field. For this example, select **No nulls** in the Type list as the key type.

   The following key types are valid:

   - **Nulls and duplicates** – allows nulls and duplicates in all the fields in the key.
   - **No nulls** – allows duplicates, if selecting No nulls, the key must contain data in at least one of the fields in the key.
   - **No duplicates** – allows null records but not duplicates. Although the data must be unique, but the field may be left blank.
   - **Unique** – must contain unique data and cannot be left blank.
   - **IR key** – files that contain an IR key must also contain another key that is Unique, with No nulls, No duplicates or Nulls, and Duplicates. A Service Manager file can contain only one IR key.
5. Click **Add**.
6. From the **Choose a field to add to the key dialog**, select the **language.key** field, and then click **OK**.
7. In the Keys section, select the key you added (language.key in this example), and click **Up**.

   The key **language.key** now appears as the first key in the list of keys in the database dictionary record. All other keys are moved down the screen.
8. Once you are finished adding keys to the table, click **Save** and then click **OK** to confirm your

changes.

Service Manager automatically updates and regen the file.

9. To remove the key, do the following:
   a. Select the key, and then click **Delete**.

   b. Click **Save**.

Add keys to bottom of the key list

Applies to User Roles:

System Administrator

You can add keys to the bottom of the key list by using System Definition or the Database Dictionary utility. Choose one of the following methods to add keys.

Note: You must use the Windows client whenever you need to add a new field/key to a database dictionary table.

Add keys to the bottom of the key list by using System Definition

To add keys to the bottom of the key list by using the System Definition:

1. In the System Navigator, click **System Definition** > **Tables** > *file_name* > **Database Definition** > **Keys**.

   Note: To expand any table in the record list, type the first letter of a table name to jump to that alphabetic section. For example, type o to jump to the oncall table.

2. Right click **New** > **Key**.

3. Select the key type, and then click **OK**.

   Your new key appears in the key list without a (+) sign to the left, because the key is still empty.

4. Drag the fields you want for that key from the Fields node and drop them on the key.

5. Right-click **Save** on the **Database Definition** node to save your changes.

Add keys to bottom of the key list by using the Database Dictionary utility

Example: Add a concatenated key including the fields **sysmodtime** and **sysmoduser** to the bottom of the key list.

To add keys to the bottom of the key list by using the Database Dictionary utility:

1. Open the file in the database dictionary, and then scroll down the keys until you reach the bottom of the list.

   The last key in the **unitofmeasure** database dictionary record is **language.key**.

   This is a single key, because it contains only one field. If the key is a concatenated key, regardless of how many fields are included in the key, only two fields are displayed in the key list.

   To view all fields contained in a concatenated key, you must use the scroll arrow buttons to the right of the array, or edit the key.

   The blank header represents the beginning of the next key. The number of lines displayed per key depends on the number of fields concatenated in that key.

   Position the cursor on the blank header field located below the key type, and then click **New Field/Key** to display the key definition window.

2. Type **Nulls & Duplicates** in the **Type** field, or select **Nulls & Duplicates** from the Type list.

3. Select the fields array, and then do the following:
   - Type **sysmodtime** in the first field of the array.
   - Type **sysmoduser** in the second field of the fields array.
   - Click **Add** to commit this new key to the database dictionary file.

     HP Service Manager displays the message: *Key added to the **unitofmeasure** file*.

     The concatenated key comprised of **sysmodtime** and **sysmoduser** now appears in the bottom of the list of keys in the **unitofmeasure** database dictionary record.

   - Place the cursor in the applicable position in the keys section of the database dictionary record, and then click **New** to add additional keys to the database dictionary record without closing the window.

4. You can reedit your database dictionary record, update it, or cancel without updating.

   Note: You must update your database dictionary record when adding a new key, or the database dictionary record will remain the same.

5. Press **OK** once you are finished adding keys to the database dictionary record, to close and automatically update the file.

   This returns the message Record updated in the dbdict file.

6. You can continue by doing one of the following:
   - Click **OK** (regen file) to continue with the regen.
   - Click **Cancel** (F3) to abort the regen and restore the database dictionary record to the version prior to the latest changes.
   - Schedule a time for the file regeneration, using **Schedule** (clock button).

7. To remove the key, select it and click **Edit**.

   The edit window appears.

8. Ignore the edit window and instead click **Delete** on the main record format.

9. The next viable key is displayed. If you do not modify this key, click **OK**.

   You are returned to the database dictionary record, minus the key.

10. Click **OK** to save the record.

Insert a key between other keys

Applies to User Roles:

System Administrator

Example: Insert the key company between the keys name and operators in a copy of the assignment file.

To insert a key between other key:

1. Open the assignment dbdict.

2. Make a copy of the assignment dbdict named assignment1.

3. Click on the key type of the second key, Nulls & Duplicates in this example, above the **operators** key.
4. Click **New**.
5. In the new key window, select a valid key in the **Type** field by either typing in the value or selecting it from the drop down menu under the arrow button. Use **No Nulls** for this example.
6. Enter **company** in the field names array.
7. Click **Add** to add this key to the assignment2 database dictionary record.
   The key **company** now appears after the *first* key, in the second place in the list, and the **operators** key has moved down the list of keys in the assignment2 database dictionary record. All other keys are moved down the screen. You can scroll the keys array to view the other keys in the list.
8. You can now reedit your database dictionary record, update it, or cancel without updating.
   You must update your database dictionary record when adding a new key, or the database dictionary record will remain the same.
9. Press **OK** once you are finished adding keys to the database dictionary record, to close and automatically update the file. This returns the message Record updated in the dbdict file.
10. You can either click **OK** (regen file) to continue with the regen, click **Cancel** (F3) to abort the regen and restore the database dictionary record to the version prior to the latest changes, or schedule a time for the file regeneration, using **Schedule** (clock button).
11. To remove the key, select it and click **Edit**.
12. The edit window appears, but ignore the window, and instead click **Delete** on the main record format.
13. The next viable key is displayed, if not modifying this key click **OK**.
    You are returned to the database dictionary record minus the key.
14. Click **OK** to save the record.

Deleting a key

You can delete keys from a database dictionary record. Remember, there is a direct correlation between the number of keys defined in a database dictionary record and the length of time it takes to add a record to that file.

A DROP INDEX statement is generated if a key is removed from the dbdict.

Note: If HP Service Manager determines it does not have rights to CREATE a TABLE it writes all database definition language (DDL) into files in the `<RUN directory>/ddl` folder and accepts changes to the dbdict without performing the DROP INDEX operation.

Delete a key

Applies to User Roles:

System Administrator

You can delete keys from System Definition and from the Database Dictionary utility. Choose one of the following methods to delete a key.

Delete a key by using System Definition

To delete a key by using System Definition:

1. From the System Navigator, click **System Definition** > **Tables** > ***file_name*** > **Fields**

   Note: You can type the first letter of a table name to jump to that alphabetic section. For example, type `o` to jump to the oncall table.

   An overview of the table opens.

2. Select the Fields and Keys tab.

3. In the Keys section, click on a key.

   Caution: When you select a key for deletion, you do not receive a confirmation message. The key is automatically deleted.

Delete a key by using the Database Dictionary utility

Example: Delete the Nulls & Duplicates key for **operators** in the **assignment1** database dictionary record.

To delete a key by using the Database Dictionary utility:

1. Click **Tailoring** > **Database Dictionary**.

2. Open the `assignment` file, and then make a copy of the assignment dbdict named assignment1.

3. Open the assignment1 dbdict record.
4. On the Keys tab, place the cursor in the Nulls & Duplicates section under which the **operators** key is located, and then click **Edit Field/Key**. The key window opens.
5. Click **Delete** from the tool tray.
6. Click **Yes** to confirm the delete.

   The screen redisplays with the key **operators** removed from the list of keys, and the keys below it moved up to take its former place.
7. Click **OK** to close and automatically update the file.

   You receive a message that states the record has been updated in the dbdict file.

Modifying a key

As new users are added to HP Service Manager and additional requirements are specified, you may find the need to modify keys in your database dictionary records in order to optimize performance.

Note: As with adding new keys, if records already exist in the file when you modify a key, Service Manager automatically performs a regeneration of file data when you update the database dictionary record.

A modification to a key may be a change in the field(s) comprising the key or a change in the key type definition.

Note: When you modify keys in a file that contains records, be careful when defining key types.

For example, do not change a key from a **Nulls & Duplicates** type to **No Nulls** if the field(s) comprising the key is **all null** (contains no value). In this case, each record that did not contain a value in the field(s) comprising the key would require an update, putting data in the field before the key could be changed to No Nulls.

A CREATE INDEX statement is generated when an existing key is changed. For example, if the key type is changed or fields are added to the key.

A DROP INDEX statement is generated when an existing key is changed. For example, if fields are removed from the key.

Note: If Service Manager determines it does not have rights to CREATE a TABLE, it assumes it also does not have the rights to perform any of the above operations either. In which case it writes all database

definition language (DDL) into files in the<RUN directory>/ddl folder and accepts changes to the dbdict without performing the CREATE INDEX or DROP INDEX operations.

Modify keys: Nulls and duplicates to unique

Applies to User Roles:

System Administrator

You can modify keys by using System Definition or by using the Database Dictionary utility. Choose one of the following methods to modify keys.

Modify keys by using System Definition

To modify keys by using System Definition:

1. From the System Navigation, click **System Definition** > **Tables** > **Database Definition** > **Keys** > ***key name***
2. Right click **Edit**.
3. Make your changes.
4. Click **OK**.
5. Right click **Save** on the **Database Definition** node to save your changes.

Modify keys by using the Database Dictionary utility

Example: Change the key type for operators in a copy of the assignment file from Nulls & Duplicates to Unique. If data already exists in the file, ensure that a unique value also exists in the operator field for every record in the file. If this condition is not met, you will then have to update each record individually or perform a Mass Update.

To modify keys by using the Database Dictionary utility:

1. Open the assignment dbdict.
2. Make a copy of the assignment dbdict named assignment2.
3. Position the cursor on the key type Nulls & Duplicates above the key field **operators**.
4. Click **Edit**.

5. In the **Type** field, type: `unique`.
6. Click **OK** to close the window and commit the changes to the file.
7. You can now reedit your database dictionary record, update it, or cancel without updating.
8. Click **OK** to close and automatically update the file.
   This returns the message *Record updated in the dbdict file*.

Modify keys: Convert the first unique key to primary key

You can run the `sm` command together with `system_addconstraint` parameter to convert the first unique key of a table to a primary key. When you run this command, Service Manager first modifies the fields of the first unique keys to be `NOT NULL` (as if you ran `system_addconstraint:0` as described in "Modify keys: Add Not Null constraints to the first unique key in a table" on the next page) and then applies a primary constraint to those keys.

To convert the first unique key to a primary key by using the command line, use one of the following methods:

> **Note:** Make sure that Service Manager is offline before you execute this command.

Method 1

Run the following `system_addconstraint` command with a value of `1`. You can use this method with an Oracle or DB2 database. You cannot use this with a SQL Server database.

```
sm -system_addconstraint:<target>:1
```

In this example, *<target>* represents the name of the Service Manager table on which you wish to add the constraint, and `1` indicates that you wish to convert the first unique key of the *<target>* to a primary key.

Method 2

> **Caution:** If Service Manager fails to perform the conversion by using the command in Method 1, or if you have a Microsoft SQL Server database, use the following method. However, you should note the following dangers when using this method:
>
> - Data loss will occur if you run this method on a unique index that contains null values.
> - This method invokes a full-table-copy operation, which may take a long time.

- Any database exception error that occurs during the operation can cause unexpected results.

To use this method, follow these steps:

1. Make sure that you have no null values on any unique index.
2. Execute the following command: `sm -system_addconstraint:<target>:2`

Note: Regardless of which method you use, an attempt to revert from a primary key to a unique key may cause unpredictable results.

Modify keys: Add `Not Null` constraints to the first unique key in a table

To add a `Not Null` constraint on a key by using the command line, enter the following command from the operating system command prompt:

Note:

- Make sure that Service Manager is offline before you execute this command. Otherwise, the "Not Null" constraint may be lost after you modify the dbdict. In this case, you would need run the command again.
- Due to differences in how databases handle "Not Null" constraints, you should note the following behaviors:

  For Oracle and DB2 databases: a "Not Null" constraint is added to the unique index.

  For SQL Server database, a "Not Null" attribute is added to each field of the unique index.

`sm -system_addconstraint:<target>:0`

In this example, *<target>* represents the tables on which you wish to add the constraint, and 0 represents that you wish to add a `Not Null` constraint on the first unique key of the *<target>*.

After you run this script, you will see information that resembles the following in the `sm.log` file.

```
 11508( 12940) 12/11/2012 11:24:59  RTE I Adding NOT NULL constraint for table
'Alertlog'.
 11508( 12940) 12/11/2012 11:24:59  RTE I Adding NOT NULL constraint for mapped
table 'ALERTLOGM1'.
```

```
 11508( 12940) 12/11/2012 11:24:59  RTE I Add NOT NULL constraint for field 'ALERT_
ID' successfully!
 11508( 12940) 12/11/2012 11:24:59  RTE I Add NOT NULL constraint for mapped table
successfully, updated to DBDICT!
 11508( 12940) 12/11/2012 11:24:59  RTE I Adding NOT NULL constraint for table
'Approval'.
 11508( 12940) 12/11/2012 11:24:59  RTE I Adding NOT NULL constraint for mapped
table 'APPROVALM1'.
 11508( 12940) 12/11/2012 11:24:59  RTE I Add NOT NULL constraint for field
'UNIQUE_KEY' successfully!
 11508( 12940) 12/11/2012 11:24:59  RTE I Add NOT NULL constraint for field 'FILE_
NAME' successfully!
 11508( 12940) 12/11/2012 11:24:59  RTE I Add NOT NULL constraint for field 'NAME'
successfully!
 11508( 12940) 12/11/2012 11:24:59  RTE E Error: SQL code=2293 message=ORA-02293:
cannot validate (SM930GA.APPROVALM18C1CC5FB) - check constraint violated
 11508( 12940) 12/11/2012 11:24:59  RTE E API=OCIStmtExecute [in sqociExecOne],
Statement=ALTER TABLE APPROVALM1 ADD CONSTRAINT APPROVALM18C1CC5FB CHECK(
"COMPONENT" IS NOT NULL )
 11508( 12940) 12/11/2012 11:24:59  RTE W Adding NOT NULL constraint for field
'COMPONENT' failed, please change/remove the NULL values in records!
 11508( 12940) 12/11/2012 11:24:59  RTE I Adding NOT NULL constraint for mapped
table 'APPROVALA1'.
 11508( 12940) 12/11/2012 11:24:59  RTE I Add NOT NULL constraint for field
'UNIQUE_KEY' successfully!
 11508( 12940) 12/11/2012 11:24:59  RTE I Add NOT NULL constraint for field 'FILE_
NAME' successfully!
 11508( 12940) 12/11/2012 11:24:59  RTE I Add NOT NULL constraint for field 'NAME'
successfully!
 11508( 12940) 12/11/2012 11:24:59  RTE E Error: SQL code=2293 message=ORA-02293:
cannot validate (SM930GA.APPROVALA153D2D5D4) - check constraint violated
 11508( 12940) 12/11/2012 11:24:59  RTE E API=OCIStmtExecute [in sqociExecOne],
Statement=ALTER TABLE APPROVALA1 ADD CONSTRAINT APPROVALA153D2D5D4 CHECK(
"COMPONENT" IS NOT NULL )
 11508( 12940) 12/11/2012 11:24:59  RTE W Adding NOT NULL constraint for field
'COMPONENT' failed, please change/remove the NULL values in records!
```

Lines in blue mean Service Manager starts to add a `NOT NULL` constraint for a logical table in SM.

Lines in dark yellow mean Service Manager starts to add a `Not Null` constraint for a physical table in SM.

Lines in red mean Service Manager cannot add a `Not Null` constraint to that field, because that field has `NULL` values in records. Therefore, you should remove these `NULL` values manually.

SQL cache

HP Service Manager uses caches for table descriptions and definitions. They can be flushed by using the Refresh SQL Cache utility (**Tailoring > SQL Utilities**). This utility can be executed at any time and may be used to force Service Manager to retrieve the table descriptions again from the relational database management system (RDBMS) in case the database administrator has changed a table while Service Manager was active. Service Manager does not cache information about keys because it does not know much about the keys in the RDBMS.

Regenerating database keys

Database keys provide efficient, organized access to records in a table. The keys define a hierarchical tree of indexes associated with the actual data records. Regenerating database keys discards and then regenerates the existing index tree for a table. The regeneration process examines each data record in the table and then adds indexes to the new index tree for each record. Key regeneration is a time-consuming process that can take up to several hours on large databases and is only necessary for IR keys.

Note: When regenerating the keys to a file, ensure that other users are not accessing the file. The regeneration process will interrupt all activity in progress on the file being regenerated.

Regenerate database keys

In this example we regenerate the keys to the contacts table. Regenerating keys for other tables works the same way.

Applies to User Roles:

System Administrator

To regenerate database keys:

1. Click **Tailoring** > **Database Manager**.
2. Select **Administration mode**.
3. In the **Form** field, type `contacts`, and then click **Search**.
4. Select the form.

5. Open the More Actions menu and choose **Regen**.

   The Confirm Action window opens.

6. From the prompt, do one of the following:
   - Click **OK** to continue with the file/key regen.
   - Click **Cancel** to exit this screen without doing a regen.
   - Click **Schedule** to schedule a time to run the file regen operation, either once or repeatedly at a set interval.

7. If you decide to schedule a time to run the file regen operation, do the following:
   - Provide a date in a DD/MM/YYYY format in the **Date/Time to run** field.
   - Provide a time in an HH:MM:SS format in the **Date/Time to run** field.
   - Select a **Repeat Interval** option if the database needs to be regenerated on a regular basis. This setting is optional. The interval period starts from the time and date set in steps 2 and 3.
     - Monthly – regenerate once a month.
     - Quarterly – regenerate every three months.
     - Semi-annually – regenerate every six months.
     - Annually – regenerate once a year.
     - Other – regenerate in the specified number of days, at the specified time.

       Use the dd hh:mm:ss format, where dd is the number of days from the initial date and time, and hh:mm:ss is the time of day at which the regen is run. For example, 26 10:00:00 sets the regen to run at 10 a.m., 26 days from the initial regeneration set in the Date/Time to run field.
   - Click **Schedule** to confirm this file/key regeneration action.

Note: When you run an IR Regen, Service Manager needs to clear the current indexes. This causes a delete to the backend database. When Service Manager attempts to delete a large set of records, the database could run out of space for this activity and cause an error. Each supported database platform provides some type of transaction or undo log to back up data changes and allow a rollback in case of errors. If you encounter such an error, please contact your database administrator and ask them to increase the size available for this purpose.

Some examples of the errors you could encounter are:

Oracle:SQL code=30036 message=ORA-30036: unable to extend segment by % in undo tablespace

SQL Server SQL State: 42000-9002 Message: [Microsoft][SQL Native Client][SQL Server]The transaction log for database '%' is full

DB2 SQL State: 57011--964 Message: [IBM][CLI Driver][DB2/NT] SQL0964C The transaction log for the database is full. SQLSTATE=57011

Schedule a regeneration

Applies to User Roles:

System Administrator

To schedule from the regen prompt:

1. Click **Tailoring** > **Database Manager**.
2. Select **Administration mode**.
3. In the **Form** field, type the form name. For example, type core, and then click **Search**.
4. Select the form.
5. Open the More Actions menu and choose **Regen**.

   The Confirm Action window opens.
6. Click **Schedule** (calendar).
7. Provide a date in a DD/MM/YYYY format in the **Date/Time to run** field.
8. Provide a time in an HH:MM:SS format in the **Date/Time to run** field.
9. Select a **Repeat Interval** option if the database needs to be regenerated on a regular basis. This setting is optional. The interval period starts from the time and date set in steps 2 and 3.
   - Monthly – regenerate once a month.
   - Quarterly – regenerate every three months.
   - Semi-annually – regenerate every six months.
   - Annually – regenerate once a year.

- Other – regenerate in the specified number of days, at the specified time.

  Use the dd hh:mm:ss format, where dd is the number of days from the initial date and time, and hh:mm:ss is the time of day at which the regen is run. For example, 26 10:00:00 sets the regen to run at 10 a.m., 26 days from the initial regeneration set in the Date/Time to run field.

10. Click **Schedule** to confirm this file/key regeneration action.

Data maps

Data maps are a means of identifying related records among separate tables in HP Service Manager. Service Manager applies data map queries to each record returned in a purge or archive selection query. For example, if during the purge and archive process you select 10 probsummary records, then Service Manager uses the probsummary data map to search for related records on each of the 10 records.

When you enable a data map, Service Manager returns a list of additional related records for each purge or archive record queried. Service Manager purges or archives these related records along with the original records you selected during the purge and archive process. For example, using a data map on 10 probsummary records may result in 30 actual records being purged or archived.

Note: Using data maps can greatly increase the number of records that you purge or archive. Since there is no limit on the number of data maps that you can define, there can potentially be a large number of mappings for any given table.

Tip: To prevent the accidental loss of substantial numbers of records, use the purge and archive function to save copies of your records to an external table before deleting them. This enables you to restore any records that you may have deleted accidentally.

By default, Service Manager has 155 pre-defined data mappings that you can use to purge or archive records and forms. These mappings contain the most commonly purged or exported records and tables. If you add custom tables and fields to your Service Manager system, you may want to create custom data mappings to purge or archive any related records based on these custom tables.

To create data mappings to related records, you must know the primary key fields and join fields used by your Service Manager tables. You must also know how to write Service Manager queries using escaped characters, special operators, and RAD variables.

Alternatives to data maps

Data maps are an optional administration feature that you can use as part of the health and maintenance of your system to identify related records to purge or archive. In addition to data mapping, the following table shows other methods to purge and archive records.

Feature	Description	Differences from data mapping
Unload script utility	Unload records from multiple tables using saved queries.	No restrictions on record queries (data map queries must identify related records). Save multiple queries as one script record (data maps can support only one type of query for each record). List forms that you want to protect from purging (data maps can only purge records).
Entity relationship diagram definition (Erddef)	Define joins by entity relationships.	Define multiple joins (Each data map record can only define one join). Not integrated with the purge/archive function. Cascading deletion of related records (data mapping can only delete records as part of a purge process).

Types of data maps

You can define three types of data maps.

Type	Description	Required fields	Joins
0	A Type 0 data map defines the primary key of a file.	**Filename** — identifies the name of the table queried to find source records. This is typically a table to purge or archive. For example, **format**. **Field Name** — identifies the primary key of the table to query. For example, **name**. **Map Query** — identifies the query to select records. For example, **"name=\""+str($umapdata)+"\""**. **Note:** The map query often uses RAD variables to select information about a record rather than hard code a particular record value. Type 0 data map records are equivalent to the following SQL query:	None

Type	Description	Required fields	Joins
		SELECT <Field Name> FROM <Filename> WHERE <Field Name>=<Value in Map Query>	
1	A Type 1 data map defines a join between two tables using one field to query the external file.	**Filename** — identifies the name of the table queried to find source records. This is typically a table to purge or archive. HP Service Manager uses the records from this table as the selection criteria to identify related records in external tables. For example, **format**. **Field Name** — identifies the join field shared between the primary and external tables. Service Manager uses the records from this field as the selection criteria to identify related records in the external table. For example, **name**. **Query File** — identifies the name of the external table queried with the join field. Service Manager searches for related records in this table. For example, **formatctrl**. Type 1 data map records are equivalent to the following SQL query: SELECT [<Override Qry> | <Field Name>] FROM <Query File> WHERE <Field Name>=<Value in FileName>	Joins the Field Name from the *Filename* to the values in the *Query File*.
2	A Type 2 data map defines a join between two tables using a more restrictive query of the external table.	**Filename** — identifies the name of the table queried to find source records. This is typically a table to purge or archive. Service Manager uses the records from this table as the selection criteria to identify related records in external table. For example, **format**. **Field Name** — identifies the join field shared between the primary and external table. Service Manager uses the records from this field as the selection criteria to identify related records in the external table. For example, **name**. **Map Query** — identifies the additional selection criteria to use when querying the join field to select records from the external target table. For example, **"name=\""+name in $umapfile+".g\" or name=\""+name in $umapfile+".w\""**. **Note:** The map query field uses the Service Manager query format. **Query File** — identifies the name of the external table that Service Manager queries with the join	Joins the *Field Name* from *Filename* to the values returned by the *Map Query* in the Query File.

Type	Description	Required fields	Joins
		field. Service Manager searches for related records in this table. For example, **formatctrl**. Type 2 data map records are equivalent to the following SQL query: SELECT [<Override Qry> | <Field Name>] FROM <Query File> WHERE <Field Name>=<Value in FileName> and <Value in Map Query>	

Add a data map record

Applies to User Roles:

System Administrator

To add a data map record:

1. Click **System Administration** > **Base System Configuration** > **Miscellaneous** > **DataMaps**.
2. Fill in the following fields.

Field	Description
Desc	Type a description of the data map record.
Filename	Type the name of the HP Service Manager table with the primary key field that you want to query. Service Manager purges or archives records from this table. This is a required field for all data map records.
Type	Service Manager automatically populates this field with the data map type 0, 1, or 2. Do not change the value of this field directly.
Field Name	Type the name of the primary key field for the table to create a Type 0 data map, or type the name of the join field to query for Type 1 and Type 2 data maps. This is a required field for all data map records.
Array?	Type `true` or **false** to specify whether the join field to query is an array. By default, this field is false. Choose **true** if the data map is to check every element in the array for related records in the external file. This field is only for Type 1 data maps.
Arrayed Structure?	Type `true` or **false** to specify whether the join field to query is an arrayed structure. By default, this field is false. Choose **true** if the data map is to

Field	Description
	check every element in the **AS Field No.** for related records in the external file. This field is only for Type 1 data maps.
AS Field No	Type the ID number of the join field to query in an Arrayed Structure. This ID number is defined in the dbdict.
Map Query	Type the query to identify source records in the primary table. Adding a query to this field makes the record a Type 0 or Type 2 data map, depending on the **Query File** value. You must write this query in the Service Manager query format, and you can use the following variables: $umapdata — This variable stores the value of the **Field Name** field of the current record. For example, the Field Name was **name**, the value could be an operator such as **System.Admin** or a form name such as **contacts** depending on the table you are querying. $umapfile — This variable reads the value of the **Filename** field in the purge or archive record. $umap.modtime — This variable reads the value of the current modification time in the purge or archive record. $umap.mod.date — This variable reads the value of the last modification date in the purge or archive record. If no value is listed, Service Manager displays the current date and time. $acntr — This variables reads the value of the element in an array or array of structures in the purge or archive record.
Query File	Type the name of the external file to query for related records. Adding an entry to this field makes the record a Type 1 or Type 2 data map, depending on the Map Query value.
Override Qry	Type the query used to replace the Field Name to identify related records in the external file. If this field is blank, Service Manager uses the Field Name as the join field to query for related records. Adding an entry to this field makes the record a Type 1 or Type 2 data map, depending on the Map Query value.
Map Cond	Type true, false, or a Boolean expression that evaluates to true or false to identify when Service Manager should use this data map record. By default, this field is true meaning that Service Manager should use the data map in all conditions.
Add to Map List	Type true, false, or a Boolean expression that evaluates to true or false to identify when Service Manager should add an entry to the data map list. By

Field	Description
	default, this field is true meaning that Service Manager should always add an entry to the data map list..

3. Click **Add**.

Create data maps to related records

Applies to User Roles:

System Administrator

You can create data maps to related records across HP Service Manager tables to use with the purge and archive functions. The purpose of data maps is to identify related records of the records you select to purge or archive. All data maps use the same basic structure to search tables and fields for related records.

To create data maps to related records:

1. Create one Type 0 data map record for each table to purge or archive. The Type 0 data map record typically identifies the primary key field of the table.
   Type 0 data map records are equivalent to the following SQL query:
   `SELECT <fieldname> FROM <filename> WHERE <fieldname>=<Value in Map Query>`

2. Create one Type 1 data map record for each join field to query for related records. The Type 1 data map record uses only the join field to search for related records in the external file.
   - To specify additional selection criteria of records from the primary table, you can add a Map Query, which creates a Type 2 data map record.
   - To specify additional selection criteria for records in the external file, you can add an override query in the **Override Qry** field. Because this query overrides the field listed in the **Field Name** field, you must include the join field within this query.

   Type 1 data map records are equivalent to the following SQL query:
   `SELECT [<override_qry> | <fieldname>] FROM <queryfile> WHERE <fieldname>=<value in filename>`

3. Create one Type 2 data map record for each join field that has additional selection criteria for records from the primary table. Type 2 data map records use the **Map Query** field in addition to the **Field Name** field to identify records from the primary table.

Type 2 data map records are equivalent to the following SQL query:
`SELECT [<override_qry> | <fieldname>] FROM <queryfile> WHERE <fieldname>=<Value in filename> and <Value in Map Query>`

Delete a data map record

Applies to User Roles:

System Administrator

To delete a data map record:

1. Click **System Administration** > **Base Configuration** > **Miscellaneous** > **Data Maps**.
2. Fill in optional search criteria, and then click **Search**.
3. Click the data map to delete.
4. Click **Delete**.
5. Click **Yes** to confirm the deletion.

Update a data map record

Applies to User Roles:

System Administrator

To update a data map record:

1. Click **System Administration** > **Base System Configuration** > **Miscellaneouus** > **Data Maps**.
2. Fill in optional search criteria, and then click **Search**.
3. Click the data map to update.
4. Fill in the following fields.

Field	Description
Desc	Type a description of the data map record.

Field	Description
Filename	Type the name of the HP Service Manager table that contains the primary key field to query. Service Manager purges or archives records from this table. This is a required field for all data map records.
Type	Service Manager automatically populates this field with the data map type 0, 1, or 2. Do not change the value of this field directly.
Field Name	Type the name of the primary key field for the table to create a Type 0 data map, or type the name of the join field to query for Type 1 and Type 2 data maps. This is a required field for all data map records.
Array?	Type `true` or `false` to identify if the join field to query is an array. By default, this field is false. Specify **true** to check every element in the array for related records in the external file. This field is only used for Type 1 data maps.
Arrayed Structure?	Type `true` or `false` to specify if the join field is an arrayed structure. By default, this field is false. Specify **true** if the data map is to check every element in the **AS Field No.** for related records in the external file. This field is used only for Type 1 data maps.
AS Field No	Type the ID number of the join field to query in an Arrayed Structure. This ID number is defined in the dbdict.
Map Query	Type the query to identify source records in the primary table. Adding a query to this field makes the record a Type 0 or Type 2 data map depending on whether you specify a **Query File** value. You must write this query in the Service Manager query format, and you can use the following variables: $umapdata — This variable stores the value of the **Field Name** field of the current record. For example, if the Field Name is **name**, the value could be an operator. $umapfile — This variable reads the value of the **Filename** field in the purge or archive record. $umap.modtime — This variable reads the value of the current modification time in the purge or archive record. $umap.mod.date — This variable reads the value of the last modification date in the purge or archive record. If no value occurs, Service Manager displays the current date and time. $acntr — This variables reads the value of the element in an array or array of structures in the purge or archive record.

Field	Description
Query File	Type the name of the external file to query for related records. Adding an entry to this field makes the record a Type 1 or Type 2 data map depending upon if you type a Map Query value.
Override Qry	Type the query to replace the **Field Name** to identify related records in the external file. If this field is blank, Service Manager uses the Field Name as the join field to query for related records. Adding an entry to this field makes the record a Type 1 or Type 2 data map depending upon if you type a Map Query value.
Map Cond	Type `true`, `false`, or a Boolean expression that is true or false to identify when Service Manager should use this data map record. By default, this field is true, which specifies that Service Manager should use the data map in all conditions.
Add to Map List	Type `true`, `false`, or a Boolean expression that is true or false to identify when Service Manager should add an entry to the data map list. By default, this field is true, which specifies that Service Manager should always add an entry to the data map list.

5. Click **Save**.

View a data map record

Applies to User Roles:

System Administrator

To view a data map record:

1. Click **System Administration** > **Base System Configuration** > **Miscellaneous** > **Data Maps**.
2. Specify the data map search criteria.
3. Click **Search**.

Exporting records

A system administrator can export HP Service Manager records to an external file using the following methods:

- Export an individual record as an unload file
  - As a binary unload
  - As a text unload
  - As a formatted text unload
- Export a record from a record list
  - As an unload file
  - As an Excel file
  - As a text file
- Export a list of records from multiple tables
  - As an unload file

Export descriptor records

You can create an export descriptor record to define the rules for exporting data from HP Service Manager tables to external files. Each export descriptor record describes the following information:

- The Service Manager source table to be exported
- The fields from the table to export
- Formatting information about each field
  - Field start position
  - Field width
  - Number of entries to extract from array fields
  - Date and time format
- The related Format Control record to use, if any, to validate and format data prior to export

Note: You cannot export database dictionary field definitions using an export description record. To export database dictionary field definitions, use the native Service Manager unload format.

When you create an export descriptor record, Service Manager validates the table selected for exporting against the database dictionary. If the validation is unsuccessful, Service Manager displays the following message: `Validation failed`

You can click on the warning message to see the reasons why the data validation failed. Although Service Manager lets you save an invalid export description record, you cannot use this record to export data. You need to correct any validation errors before exporting data. After Service Manager validates the table, it exports each record as a single line entry in the resulting external file.

Unload error messages

HP Service Manager displays an error message if it cannot successfully unload an external file.

Unload error message	Description
Invalid unload append/create parameter.	Valid parameters are append or create for the unload append/create input field.
Select ONE export format: binary, text, or formatted text.	Valid entries are binary, text, and formatted text for the unload mode input field.
Invalid unload dbdict load parameter.	Valid entries are create, replace, and remove for the unload dbdict input field.
Unload mode must be export.	When you specify an export descriptor record, you must set the unload type to export for the unload mode input field.
Invalid unload record load parameter.	Valid entries are update and add only for the record load input field.
No unload filename.	Type an external file name for the unload file.
The unload format must be binary for Purge/Archives.	Select the binary unload format when purging or archiving records.
<*database name*> file unloaded to external file: <*Unix filename*>	Unload processing the named database to the Unix file complete.
External file: <*UNIX.filename*> is locked; try again later.	Another task has exclusive control over the named external file.
The config record (database load/unload) is missing.	The hardware configuration record database load/unload.unix or database load/unload.winnt required for database loads and unloads has been deleted or renamed. It must be replaced.

Unload error message	Description
<*nn*> records unloaded so far from <*database name*> file to <*unload file*>	The unload utility generates this message every hundredth record it attempts to unload.
<*database name*> file cannot be unloaded to external file: <*UNIX/NT.FILENAME*>.	Service Manager could not connect to the named external file.
unable to allocate.	This message occurs in two distinct situations: The unload utility attempted to connect to a non-existent external file causing the message to be generated. However, the unload utility will attempt to dynamically allocate the Unix/NT filename. If the dynamic allocation is successful, confirmation messages will follow. In this situation, it does not indicate an error condition. The unload utility attempted to connect to a non-existent external file causing the message to be generated. The subsequent attempt to dynamically allocate the Unix/NT filename failed causing the message to be generated a second time. Other error messages will follow indicating the error condition.
Application:file.unload has completed with errors; check other messages.	An error exit condition has occurred; check the message log for the error message indicating which exit.
Invalid unload options specified for unload, no records unloaded.	Since the unload options were invalid, no records were unloaded.

Unload files

The native HP Service Manager export format is the unload file. An unload file stores the database dictionary of Service Manager tables in addition to records. The database dictionary embedded in an unload file determines where Service Manager loads records during import. The unload file also specifies whether Service Manager should update or ignore duplicate records when loading records. Service Manager can export unload files in the following file formats:

- Binary — The binary unload file format archives Service Manager tables. You can load binary unload files only from the Database Manager.

- Text — The text unload file format is human readable but only intended to be loaded from the Database Manager.
- Formatted text — The formatted text unload file format is intended to be compatible with external applications, although the actual format of the file is determined by the form you select as a template. You can create export forms to create formatted text unload files compatible with specific applications.

You can export records into an unload file from the following Service Manager utilities:

- Database Manager
- Unload Script Utility
- Directly from some record lists

After you load a .unl file to import data into Service Manager, you must stop and then restart the Service Manager server. This ensures that the table changes are initialized.

Note: Service Manager encodes unload files in the UTF-8 format. Service Manager can import unload files from previous versions of HP ServiceCenter, but earlier versions cannot import unload files created by Service Manager.

Add an export descriptor record

Applies to User Roles:

System Administrator

To add an export descriptor record:

1. Click **Tailoring** > **Database Manager**.
2. In the Form field, type **=export**, and click **Search**.

   A list of existing export descriptor records is displayed.
3. Select a record from the list, and update the following fields.

Field	Description
Export Descriptor Record	Type the unique name for this record.
Service Manager File Name	Type the name of the Service Manager database table exporting records.
Format Control Record Name	Type the name of the optional Format Control record to use with the export process. This Format Control record determines how Service Manager formats exported records.
Field Name	Type the database dictionary name of the fields exporting data. If the field is part of a structure, include the structure name before the field name. If the field is part of an arrayed structure, type only the field name.
Field Type	Service Manager fills this field automatically using the database dictionary definition of the field. You do not need to type any information in this field.
Start Pos	Type the number of space characters between the start of a line and the information Service Manager exports from the selected field. This information determines which column and line number Service Manager uses to export the field information. **Note**: Most external files use a standard of 80 characters per line. If you type a value greater than 80 then each record will use multiple lines.
Fld Lng	Type the number of characters Service Manager has to write information from the exported field into the external file.
Array Type	Service Manager fills in the value of this field automatically based on the database dictionary definition of the field. You do not need to manually type any information in this field.
Array Occ	Type the number of elements in an array that you want Service Manager to export. You can use the Array Occ value to determine the start position of the next field. For example, if you want to export five elements from an array and you allocated 80 character spaces for each element, this field requires a total of 400 character spaces. The next field should begin more than 400 characters spaces from the start of the file.
Date Format	Type the date and time format for date and time fields. You can use the following tokens to represent units of time: dd — two-digit day

Field	Description
	mm — two-digit month yy — two-digit year yyyy — four-digit year hh — hours ii — minutes ss — seconds You can use any character to separate these tokens. For example, mm/dd/yyyy hh:ii:ss exports dates and times with slashes between the month, day, and four-digit year and colons between the hours, minutes, and seconds. **Note**: If you do not include a time format, then Service Manager does not export any time information.

4. Click **Add**.

Delete an export descriptor record

Applies to User Roles:

System Administrator

To delete an export descriptor record:

1. Click **Tailoring** > **Database Manager**.
2. Type `=export` in the **Form** field.
3. Select **Administration Mode**. You must select this option for HP Service Manager to display the **Delete** button.
4. Click **Search**.
5. Click **Search** again to display a list of export descriptor records.

   Note: You may need to click **Document** > **Restore** to see the list of export descriptor records.
6. Select the export descriptor record to be deleted.
7. Click **Delete**.
8. Click **Yes** to confirm the deletion.

Export a record list to a text file

Applies to User Roles:

All users

To export a record list to a text file:

Open a list of records. For example, open a list of open incident records.

1. Click **Incident Management** > **Search Incidents**.
2. Select **Open** status, and then click **Search**.

   A list of incident records opens.

   Note: You can click **More** or the More Actions icon and then click **Modify Columns**, so that only those columns containing data that you intend to export are displayed. You must keep the column that contains the key field.
3. From the record list, click **More** or the More Actions icon, and then choose **Export to Text File**.
4. Fill in the following fields.

Field	Description
Filename	Type, or browse to, the path and name of the new text file.
Comma Separated Value (CSV)	Select this option to use the comma character as a field delimiter.
Semicolon Separated CSV	Select this option to use the semicolon character as a field delimiter.
Tab	Select this option to use the tab character as a field delimiter.
Asterisk	Select this option to use the asterisk character as a field delimiter.
Caret	Select this option to use the caret character as a field delimiter.
Other	Select this option to use a designated character as a field delimiter. Type the character delimiter.

5. Do one of the following to save the file:
   - *For Windows clients:*

     In the **Filename** field, specify a path and name for the text file, and then click **OK**.

   - *For Web clients:*

     Click **OK**. Save the file when a download job starts in your Web browser.

Export a record list to an Excel file

Applies to User Roles:

All users

Note: You must have the Microsoft Excel application program on your computer.

To export a record list to an Excel file:

Windows client:

1. Go to **Window** > **Preferences**, and make sure that the **Client side load/unload** option is selected.
2. Open a list of records. For example, open a list of open incident records.

   Note: You can use **Modify Columns**, so that only those columns containing data that you intend to export displays. You must keep the column that contains the key field.
3. From the record list, click the More Actions icon, and choose **Export to Excel**.
4. In the **Filename** field, specify a path and name for the file, and then click **OK**.

   Note: If you specify a file that does not exist, Service Manager will create it.

   HP Service Manager exports the record list to a CSV file and opens it in Excel.

Web client:

For a Web tier client, export the record list to a text file, and then open the text file in Excel. Do the following:

1. From the record list, click **More** and choose **Export to Text File**.
2. Fill in the following fields.

Field	Description
Filename	Type, or browse to, the path and name of the new text file.
Comma Separated Value (CSV)	Select this option to use the comma character as a field delimiter.
Semicolon Separated CSV	Select this option to use the semicolon character as a field delimiter.
Tab	Select this option to use the tab character as a field delimiter.
Asterisk	Select this option to use the asterisk character as a field delimiter.
Caret	Select this option to use the caret character as a field delimiter.
Other	Select this option to use a designated character as a field delimiter. Type the character delimiter.

3. Open the Excel application, and then click **Open**.
4. Select **All files** to see a list of all files in your directory.
5. Select the new text file of the records list that you just created.
6. Answer the Excel program prompts to import your text file. For example, select **Delimited**, so that characters such as commas or semicolons separate each field.
7. When you are done making your selections, click **Finish**.
8. Click **OK**.

Your record list is imported into the Excel spreadsheet.

Export a record list using an export description record

Applies to User Roles:

System Administrator

To export a record list using an export description record:

1. Click **Tailoring** > **Database Manager**.
2. Type the form or table name that contains the records to be unloaded.
3. Select the **Administration Mode** option. You must select this option to ensure that HP Service Manager displays the unload options in the target form.
4. Click **Search**.
5. Select the form used to search for records.
6. Fill in optional search criteria, and then click **Search**.
7. A record list appears. Click **Mass Unload**.
8. Fill in the following fields.

Field	Description
External File Name	Type the name and path of the file used to save the record list. By default, this file is located in the RUN folder of the Service Manager server.
Member	This field is obsolete.
Unload Type	Select the operating system file type for the external file.
Append to File	Select this option to add records to the end of an external file of the same name. If an external file of the same name does not exist, Service Manager creates a new file. If this field is cleared, then Service Manager overwrites any existing file of the same name.
Use Data Map	Select this option to use a pre-defined data map when unloading the selected records. Using a data map may result in Service Manager unloading more records than you originally selected to unload.
Prompt for Data Map	Select this option to enable a confirmation message to use a data map for each record unloaded. This option is only valid if the **Use Data Map** option is also enabled.
Suppress Messages	Select this option to hide all messages that the unload process generates.
Export Mode — Formatted Text	Select this option to save the external file using an export descriptor record. This format is intended for export to other applications.

Field	Description
Form Name	Type the name of the export descriptor record. This option is valid only when the **Formatted Text** option is selected. If the export descriptor record does not exist, Service Manager prompts you to create it.

9. Click **Export Ext**.
10. If Service Manager displays a list of existing export descriptor records, select the export descriptor record to use.
11. Do one of the following:
    - To change the export descriptor record, click **Edit Record**to edit the fields.
    - To export the record, click **Export**.
      Service Manager saves the select records to an external file.

Export an individual record using an export descriptor record

Applies to User Roles:

System Administrator

To export an individual record using an export descriptor record:

1. Click **Tailoring** > **Database Manager**.
2. Type the form or table name that contains the record to be unloaded.
3. Select the **Administration Mode** option. You must select this option to ensure that HP Service Manager displays the unload options in the target form.
4. Click **Search**.
5. Select the form used to search for the record.
6. Fill in optional search criteria, and then click **Search**.
7. Select the record to unload.
8. Click **More** or the More Actions icon, and then select **Export/Unload**.

9. Fill in the following fields.

Field	Description
External File Name	Type the name and path of the destination file. By default, this file is located in the RUN folder of the Service Manager server.
Member	This field is obsolete.
Unload Type	Select the operating system file type for the external file.
Append to File	Select this option to add records to the end of an external file of the same name. If a external file of the same name does not exist, Service Manager creates a new file. If this field is cleared, then Service Manager overwrites any existing file of the same name.
Use Data Map	Select this option to use a pre-defined data map when unloading the selected records. Using a data map may result in Service Manager unloading more records than you originally selected to unload.
Prompt for Data Map	Select this option to enable a confirmation message to use a data map for each record unloaded. This option is only valid if the **Use Data Map** option is also enabled.
Suppress Messages	Select this option to hide all messages that the unload process generates.
Export Mode — Formatted Text	Select this option to save the external file using an export descriptor record. This format is intended for export to other applications.
Form Name	Type the name of the export descriptor record to use. This option is only valid with the **Formatted Text** option is selected. If the export descriptor record does not already exist, Service Manager prompts you to create it.

10. Click **Export Ext**.

11. If Service Manager displays a list of existing export descriptor records, select the export descriptor record to use.

12. Do one of the following:
    - To change the export descriptor record, click **Edit Record**.
    - To export the record, click **Export**.
      Service Manager saves the select records to an external file.

Export selected records from a table using an export descriptor record

Applies to User Roles:

System Administrator

To export selected records from a table using an export descriptor record:

1. Click **Tailoring** > **Database Manager**.
2. Type the form or table name that contains the records to unload.
3. Select the **Administration Mode** option.

   Note: You must select this option to ensure that HP Service Manager displays the unload options in the target form.
4. Click **Search**.
5. Select the form used to search for records. Do not type or select any search criteria from this form.
6. Click **More** or the More Actions icon, and then select **Export/Unload**.

   Fill in the following fields.

Field	Description
Field Name	Type the name of the field containing the selection criteria.
Field Values	Type a unique value for each record unloaded from the **Field Name**.

7.
8. Click **Unl Records**.
   Service Manager displays a record list of the records you selected.
9. From the record list, select the form, click **More** or the More Actions icon, and then select **Mass Unload**.
10. Fill in the following fields.

Field	Description
External File Name	Type the name and path of the destination file to contain the record list. By default, this file is located in the RUN folder of the Service Manager server.
Member	This field is obsolete.
Unload Type	Select the operating system file type to use for the external file.
Append to File	Select this option to add records to the end of an external file of the same name. If a external file of the same name does not exist, Service Manager creates a new file. If this field is cleared, then Service Manager overwrites any existing file of the same name.
Use Data Map	Select this option to use a pre-defined data map when unloading the selected records. Using a data map may result in Service Manager unloading more records than you originally selected to unload.
Prompt for Data Map	Select this option to enable a confirmation message to use a data map for each record unloaded. This option is only valid if the **Use Data Map** option is also enabled.
Suppress Messages	Select this option to hide all messages that the unload process generates.
Export Mode — Formatted Text	Select this option to save the external file using an export descriptor record. This format is intended for export to other applications.
Form Name	Type the name of the export descriptor record to use. This option is only valid with the **Formatted Text** option is selected. If the export descriptor record does not already exist, Service Manager prompts you to create it.

11. Click **Export Ext**.
12. If Service Manager displays a list of existing export descriptor records, select the export descriptor record to use.
13. Do one of the following:
    - To change the export descriptor record, click **Edit Record**.
    - To export the record, click **Export**.
      Service Manager saves the select records to an external file.

Schedule the exporting of records using an export descriptor record

Applies to User Roles:

System Administrator

To schedule the exporting of records using an export descriptor record:

1. Select the records to export.
2. From the file.prompt.dbu.g form, also known as the HP Service Manager unload/export facility, click **Export Ext**.
3. If Service Manager displays a list of existing export descriptor records, select the export descriptor record to use.
4. Do one of the following:
   - To change the export descriptor record, click **Edit Record**.
   - To schedule the export of records, click **Schedule**.
5. Fill in the following fields.

Field	Description
Date/Time to Run	Specify the date and time to unload the records.
Repeat Interval — Enter Number of Days and Time	Type the number of days, hours. minutes, and seconds to wait before repeating the unload process. Enter the interval with the following format: *<Days><Hours>:<Minutes>:<Seconds>* You can specify a repeat interval or select one of the pre-defined intervals. Any value you type in this field overrides the pre-defined intervals.
Repeat Interval — Monthly	Select this option to repeat the export process every once every month from today's date.
Repeat Interval — Quarterly	Select this option to repeat the export process every once every quarter from today's date.
Repeat Interval — Semi-	Select this option to repeat the export process every once

Field	Description
Annually	every six months from today's date.
Repeat Interval — Annually	Select this option to repeat the export process every once every year from today's date.

6. Click **Confirm**.

Schedule the unloading of records

Applies to User Roles:

System Administrator

To schedule the unloading of records:

1. Click **Tailoring** > **Database Manager**.
2. Type the form or file name that contains the records to unload, and then click **Search**.
3. Select a form, and then click **Search** again.

   A list of records opens.
4. Select a record.
5. Open the More Actions menu and choose **Export/Unload**.
6. Fill in optional search criteria.
7. From the file.prompt.dbu.g form, also known as the unload/export facility, click **Schedule**.
8. Fill in the following fields.

Field	Description
Date/Time to Run	Specify the date and time to unload the records.
Repeat Interval — Enter Number of Days and Time	Type the number of days, hours. minutes, and seconds to wait before repeating the unload process. Enter the interval with the following format:

Field	Description
	<Days><Hours>:<Minutes>:<Seconds> **Note**: You can either type in a repeat interval or select one of the pre-defined intervals. Any value you type in this field overrides the pre-defined intervals.
Repeat Interval — Monthly	Select this option to repeat the unload process every once every month from today's date.
Repeat Interval — Quarterly	Select this option to repeat the unload process every once every quarter from today's date.
Repeat Interval — Semi-Annually	Select this option to repeat the unload process every once every six months from today's date.
Repeat Interval — Annually	Select this option to repeat the unload process every once every year from today's date.

9. Click **Confirm**.

Unload a database dictionary only

Applies to User Roles:

System Administrator

To unload a database dictionary only:

1. Click **Tailoring** > **Database Manager**.
2. Type the form or file name that contains the records to unload.
3. Select the **Administration Mode** option. You must select this option to ensure that HP Service Manager displays the unload options in the target form.
4. Click **Search**.
5. Select the form to use to search for records. Do not type or select any search criteria from this form.
6. Click **More** or the More Actions icon, and then select **Export/Unload**.

7. Click **Unl Dbdict**.

8. Fill in the following fields.

Field	Description
External File Name	Type the name and path of the file used to save the record list. By default, this file is located in the RUN folder of the Service Manager server.
Member	This field is obsolete
Unload Type	Select the operating system file type for the external file.
Append to File	Select this option to add records to the end of an external file of the same name. If a external file of the same name does not exist, Service Manager creates a new file. If this field is cleared, then Service Manager overwrites any existing file of the same name.
Use Data Map	Select this option to use a pre-defined data map when unloading the selected records. Using a data map may result in Service Manager unloading more records than you originally selected to unload.
Prompt for Data Map	Select this option to enable a confirmation message to use a data map for each record unloaded. This option is only valid if the **Use Data Map** option is also enabled.
Suppress Messages	Select this option to hide all messages that the unload process generates.
Export Mode — Binary	Select this option to save the external file in Service Manager's binary unload format. Only the Service Manager Database Manager can read this binary file format.
Export Mode — Text	Select this option to save the external file in Service Manager's text format. This format is human readable but is not intended for export to other applications other than Service Manager's Database Manager.
Export Mode — Formatted Text	Select this option to save the external file in a user-defined text format. This format is intended for export to other applications. If you select this option, you must type a valid Service Manager form name in the **Form Name** field.
Form Name	Type the name of the Service Manager form to use when exporting records as formatted text. This option is only valid with the **Formatted Text** option is selected. If the text format description does not already exist, Service Manager prompts you to create it.

Field	Description
Dbdict Load Options — Create file if not there, and update if present	Select this option if the data saved in this external file should replace existing data in the target Service Manager system.
Dbdict Load Options — Create file if not there, but do not update if present	Select this option to ignore data saved in the external file that already exists in the target Service Manager system.
Dbdict Load Options — Create file if not there, and update if present, including keys	Select this option if the data saved in this external file should replace existing data in the target Service Manager system including the database keys.
Dbdict Load Options — Create file if not there, and update if present, including SQL mapping	Select this option if the data saved in this external file should replace existing data in the target Service Manager system including the database SQL mapping.
Dbdict Load Options — Create file if not there, and update if present, including keys AND SQL mapping	Select this option if the data saved in this external file should replace existing data in the target Service Manager system including the database SQL mapping and the database keys.
Record Load Options — Add/update record	Select this option if the records saved in this external file are to replace any existing records in the target Service Manager system.
Record Load Options — Only add new records	Select this option to ignore records saved in the external file that already exist in the target Service Manager system.

9. Click **Unload/export**.
   Service Manager saves the select records to an external file.

Unload a record list

Applies to User Roles:

System Administrator

To unload a record list:

1. Click **Tailoring** > **Database Manager**.
2. Type the form or file name that contains the records to unload.
3. Select the **Administration Mode** option. You must select this option to ensure that HP Service Manager displays the unload options in the target form.
4. Click **Search**.
5. Click the form used to search for records.
6. Fill in optional search criteria.
7. Click **Search**.
8. Click **More** or the More Actions icon for record list, and then select **Mass Unload**.
9. Fill in the following fields.

Field	Description
External File Name	Type the name and path of the file used to save the record list. By default, this file is located in the RUN folder of the Service Manager server.
Member	This field is obsolete.
Unload Type	Select the operating system file format for the external file.
Append to File	Select this option to add records to the end of an external file of the same name. If an external file of the same name does not exist, Service Manager creates a new file. If this field is cleared, Service Manager overwrites any existing file of the same name.
Use Data Map	Select this option to use a pre-defined data map when unloading the selected records. Using a data map may result in Service Manager unloading more records than you originally selected to unload.
Prompt for Data Map	Select this option to enable a confirmation message to use a data map for each record unloaded. This option is only valid if the **Use Data Map** option is also enabled.
Suppress Messages	Select this option to hide all messages that the unload process generates.
Export Mode — Binary	Select this option to save the external file in Service Manager's

Field	Description
	binary unload format. Only the Service Manager Database Manager can read this binary file format.
Export Mode — Text	Select this option to save the external file in Service Manager's text format. This format is human readable but is not intended for export to other applications other than Service Manager's Database Manager.
Export Mode — Formatted Text	Select this option to save the external file in a user-defined text format. This format is intended for export to other applications. If you select this option, you must type a valid Service Manager form name in the **Form Name** field.
Form Name	Type the name of the Service Manager form to use when exporting records as formatted text. This option is only valid with the **Formatted Text** option is selected. If the text format description does not already exist, Service Manager prompts you to create it.
Dbdict Load Options — Create file if not there, and update if present	Select this option if the data saved in this external file should replace existing data in the target Service Manager system.
Dbdict Load Options — Create file if not there, but do not update if present	Select this option to ignore data saved in the external file that already exists in the target Service Manager system.
Dbdict Load Options — Create file if not there, and update if present, including keys	Select this option if the data saved in this external file should replace existing data in the target Service Manager system including the database keys.
Dbdict Load Options — Create file if not there, and update if present, including SQL mapping	Select this option if the data saved in this external file should replace existing data in the target Service Manager system including the database SQL mapping.
Dbdict Load Options — Create file if not there, and update if present, including keys AND SQL mapping	Select this option if the data saved in this external file should replace existing data in the target Service Manager system including the database SQL mapping and the database keys.
Record Load Options — Add/update records	Select this option if the records saved in this external file are to replace any existing records in the target Service Manager system.
Record Load Options — Only add new records	Select this option to ignore records saved in the external file that already exist in the target Service Manager system.

10. Click **Unload/export**.
    Service Manager saves the select records to an external file.

Unload an individual record

Applies to User Roles:

System Administrator

To unload an individual record:

1. Click **Tailoring** > **Database Manager**.
2. Type the form or file name that contains the record to unload.
3. Select the **Administration Mode** option. You must select this option to ensure that HP Service Manager displays the unload options in the target form.
4. Click **Search**.
5. Click the form used to search for the record.
6. Fill in optional search criteria.
7. Click **Search**.
8. Select the record to unload.
9. Click **More** or the More Actions icon for the record, and then select **Export/Unload**.
10. Fill in the following fields.

Field	Description
External File Name	Type the name and path of the file used to save the record. By default, this file is located in the RUN folder of the Service Manager server.
Member	This field is obsolete.

Field	Description
Unload Type	Select the operating system file type to use for the external file.
Append to File	Select this option to add records to the end of an external file of the same name. If a external file of the same name does not exist, Service Manager creates a new file. If this field is cleared, then Service Manager overwrites any existing file of the same name.
Use Data Map	Select this option to use a pre-defined data map when unloading the selected records. Using a data map may result in Service Manager unloading more records than you originally selected to unload.
Prompt for Data Map	Select this option to enable a confirmation message to use a data map for each record unloaded. This option is only valid if the **Use Data Map** option is also enabled.
Suppress Messages	Select this option to hide all messages that the unload process generates.
Export Mode — Binary	Select this option to save the external file in Service Manager's binary unload format. Only the Service Manager Database Manager can read this binary file format.
Export Mode — Text	Select this option to save the external file in Service Manager's text format. This format is human readable but is not intended for export to other applications other than Service Manager's Database Manager.
Export Mode — Formatted Text	Select this option to save the external file in a user-defined text format. This format is intended for export to other applications. If you select this option, you must type a valid Service Manager form name in the **Form Name** field.
Form Name	Type the name of the Service Manager form to use when exporting records as formatted text. This option is valid only with the **Formatted Text** option is selected. If the text format description does not already exist, Service Manager prompts you to create it.
Dbdict Load Options — Create file if not there, and update if present	Select this option if the data saved in this external file should replace existing data in the target Service Manager system.
Dbdict Load Options — Create file if not there, but do not update if present	Select this option to ignore data saved in the external file that already exists in the target Service Manager system.

Field	Description
Dbdict Load Options — Create file if not there, and update if present, including keys	Select this option if the data saved in this external file should replace existing data in the target Service Manager system including the database keys.
Dbdict Load Options — Create file if not there, and update if present, including SQL mapping	Select this option if the data saved in this external file should replace existing data in the target Service Manager system including the database SQL mapping.
Dbdict Load Options — Create file if not there, and update if present, including keys AND SQL mapping	Select this option if the data saved in this external file should replace existing data in the target Service Manager system including the database SQL mapping and the database keys.
Record Load Options — Add/update records	Select this option if the records saved in this external file are to replace any existing records in the target Service Manager system.
Record Load Options — Only add new records	Select this option to ignore records saved in the external file that already exist in the target Service Manager system.

11. Click **Unload/export**.

Unload records from multiple files

Applies to User Roles:

System Administrator

To unload records from multiple files:

1. Click **Tailoring** > **Unload Script Utility**.
2. Fill in the following fields.

Field	Description
Unload Script	Type a name or label to identify your unload script. This name can include spaces.
Unload?	Select this option to create an external unload file from the records

Field	Description
	queried by this script.
Purge?	Select this option to delete the records queried by this script.
Show Unload Records	Select this option to display the Filename, Query, and Data Map fields. These fields enable you to specify the tables and queries to use when unloading records.
Filename	Specify the name of the table to unload records. This field is visible only if you select **Show Unload Records**.
Query	Type the SQL query to select records for unloading. You can use HP Service Manager operators and variables in this field. This field is visible only if you select **Show Unload Records**.
Datamap	Specify true or false to specify whether Service Manager should use the data map file to unload any associated records. You must have previously defined a data map for the listed file in order to use this feature. By default, Service Manager reads a blank entry in this field as false. This field is only visible if you have select **Show Unload Records**.
Show Protected Formats	Select this option to display the Protected Formats field. This field enables you to specify the forms to protect from purging.
Protected Formats	Type the names of the forms to protect when you purge records from this script.

3. Click **Add**.

Unload selected records from a file

Applies to User Roles:

System Administrator

To unload selected records from a file:

1. Click **Tailoring** > **Database Manager**.
2. Type the form or file name that contains the records to unload.

3. Select the **Administration Mode** option. You must select this option to ensure that HP Service Manager displays the unload options in the target form.
4. Click **Search**.
5. Select the form used to search for records.
6. Click **More** or the More Actions icon, and then select **Export/Unload**.

   Fill in the following fields.

Field	Description
Field Name	Type the name of the field used as the selection criteria. This field contains the values you list in the **Field Values** field.
Field Values	Type a unique value for each record to unload from the **Field Name**.

7.
8. Click **Unl Records**.
9. Select the form, click **More** or the More Actions icon, and then select **Mass Unload**.
10. Fill in the following fields.

Field	Description
External File Name	Type the name and path of the file used to save the record list. By default, this file is located in the RUN folder of the Service Manager server.
Member	This field is obsolete.
Unload Type	Select the operating system file type to use for the external file.
Append to File	Select this option to add records to the end of an external file of the same name. If a external file of the same name does not exist, Service Manager creates a new file. If this field is cleared, then Service Manager overwrites any existing file of the same name.
Use Data Map	Select this option to use a pre-defined data map when unloading the selected records. Using a data map may result in Service Manager unloading more records than you originally selected to unload.

Field	Description
Prompt for Data Map	Select this option to enable a confirmation message to use a data map for each record unloaded. This option is only valid if the **Use Data Map** option is also enabled.
Suppress Messages	Select this option to hide all messages that the unload process generates.
Export Mode — Binary	Select this option to save the external file in Service Manager's binary unload format. Only the Service Manager Database Manager can read this binary file format.
Export Mode — Text	Select this option to save the external file in Service Manager's text format. This format is human readable but is not intended for export to other applications other than Service Manager's Database Manager.
Export Mode — Formatted Text	Select this option to save the external file in a user-defined text format. This format is intended for export to other applications. If you select this option, you must type a valid Service Manager form name in the **Form Name** field.
Form Name	Type the name of the form to use when exporting records as formatted text. This option is valid only when you select the **Formatted Text** option. If the text format description does not already exist, Service Manager prompts you to create it.
Dbdict Load Options — Create file if not there, and update if present	Select this option if the data saved in this external file should replace existing data in the target Service Manager system.
Dbdict Load Options — Create file if not there, but do not update if present	Select this option to ignore data saved in the external file that already exists in the target Service Manager system.
Dbdict Load Options — Create file if not there, and update if present, including keys	Select this option if the data saved in this external file should replace existing data in the target Service Manager system including the database keys.
Dbdict Load Options — Create file if not there, and update if present, including SQL mapping	Select this option if the data saved in this external file should replace existing data in the target Service Manager system including the database SQL mapping.
Dbdict Load Options — Create file if not there,	Select this option if the data saved in this external file should replace existing data in the target Service Manager system

Field	Description
and update if present, including keys AND SQL mapping	including the database SQL mapping and the database keys.
Record Load Options — Add/update records	Select this option if the records saved in this external file are to replace any existing records in the target Service Manager system.
Record Load Options — Only add new records	Select this option to ignore records saved in the external file that already exist in the target Service Manager system.

11. Click **Unload/export**.
    Service Manager saves the select records to an external file.

Update an export descriptor record

Applies to User Roles:

System Administrator

To update an export descriptor record:

1. Select the records to export.
2. From the file.prompt.dbu.g form, also known as the unload/export utility, click **Export Ext**.
3. If HP Service Manager displays a list of existing export descriptor records, select the export descriptor record to use.
4. Click **Edit Record**.
5. Fill in the following fields.

Field	Description
Export Descriptor Record	Type the name used to identify this record uniquely.
File Name	Type the name of the Service Manager table used to export records.
Format	Type the name of the optional Format Control record to use with the export

Field	Description
Control Record Name	process. This Format Control record determines how Service Manager formats exported records.
Field Names	Type the database dictionary name of the Service Manager field used to export records. If the field is part of a structure, include the structure name before the field name. If the field is part of an arrayed structure, you only need to type the field name.
Field Type	Service Manager fills in the value of this field automatically based on the database dictionary definition of the field. You do not need to manually type any information in this field.
Start Pos	Type the number of space characters between the start of a line and the information Service Manager exports from the selected field. This information determines which column and line number used to export the field information. Most external files use a standard of 80 characters per line. If you type a value greater than 80 then each record uses multiple lines.
Fld Lng	Type the number of characters Service Manager has to write information from the exported field into the external file.
Array Type	Service Manager fills in the value of this field automatically based on the database dictionary definition of the field. You do not need to manually type any information in this field.
Array Occ	Type the number of elements from an array to be exported for this field. You can use the Array Occ value to determine the start position of the next field. For example, if you want to export five elements from an array and you allocated 80 character spaces per element, this field requires a total of 400 character spaces. The next field should start more than 400 characters spaces from the start of the file.
Date Format	Type the date and time format for date and time fields. You can use the following tokens to represent units of time: *dd* — two digit day *mm* — two digit month *yy* — two digit year *yyyy* — four digit year *hh* — hours *ii* — minutes *ss* — seconds You can use any character to separate these tokens. For example, mm/dd/yyyy hh:ii:ss exports dates and times with slashes between the month, day, and four-

Field	Description
	digit year and colons between the hours, minutes, and seconds. **Note**: If you do not include a time format, then Service Manager does not export any time information.

6. Click **Save**.

Importing records

A system administrator can import HP Service Manager records from an external unload file using the Database Manager. Service Manager can import records from all three formats of unload files (binary unload, text unload, and formatted text unload) as well as from two formats of text files (fixed-width and character-delimited text files). You can import records into Service Manager as a manual foreground process or as an automated scheduled background process.

How Service Manager imports records depends upon the format of the source file. When importing unload files, Service Manager uses the database dictionary stored in these files to determine where and how to import records into Service Manager. The database dictionary inside an unload file determines whether Service Manager updates or ignores duplicate records in the existing system. When importing text files, Service Manager requires an import descriptor record to tell it where and how to import records. You can define an import descriptor record during the import process or prepare one in advance.

An import descriptor record defines:

- The external file source
- The internal destination table
- The format used to identify fields in the imported file
  - Fixed-width fields
  - Character-delimited fields
- The rules for handling duplicate records
  - Ignore duplicate records
  - Replace duplicate records

- Update duplicate records

- The optional Format Control record to apply
- The names of destination fields for import records
- The date format to use
- The number of elements from an array to import

Import descriptor records

You can create an import descriptor record to define the rules for importing data from external files into HP Service Manager. Each import descriptor record describes the following information:

- The external file source
- The internal destination table
- The format used to identify fields in the imported file
  - Fixed-width fields
  - Character-delimited fields
- The rules for handling duplicate records
  - Ignore duplicate records
  - Replace duplicate records
  - Update duplicate records
- The optional Format Control record to apply
- The names of destination fields for import records
- The date format to use
- The number of elements from an array to import

Note: You cannot import the database dictionary definition of fields from an external file. To import database dictionary field definitions, you must use the native HP Service Manager unload format.

When you create an import descriptor record, Service Manager validates the records in the external file against the Service Manager database dictionary. If the validation is unsuccessful, Service Manager displays the message: `Validation failed`

You can click on the warning message to see the reasons why the data validation failed. Although Service Manager enables you to save an invalid import description record, you cannot use this record to import data. You must correct any validation errors before importing data. After Service Manager validates the file, it imports one record for each entry in the external file.

Load error messages and notifications

HP Service Manager displays an error message if it cannot successfully load an external file. The following list contains load error messages and notifications.

Load error message or notification	Description
<WINDOWS.FILENAME> or <UNIX.FILENAME> **file loaded.**	Service Manager has finished loading the named file.
File(s) loaded from external file: ***<WINDOWS.FILENAME> or <UNIX.FILENAME>***	Service Manager has finished loading the named files.
<Number> **records loaded into** ***<Database name>*** **from external file:** ***<WINDOWS.FILENAME> or <UNIX.FILENAME>***	Service Manager has finished loading *<Number>* records from the named file.
Loading ***<Database name>*** **from external file:** ***<WINDOWS.FILENAME> or <UNIX.FILENAME>*** **.**	Service Manager has started loading *<Database name>* from the named file.
<Number> **records have been loaded so far into the** ***<Database name>*** **file.**	The load utility generates this message for every hundredth record it attempts to load into the named Service Manager database.
The config record (database load/unload) is missing.	The hardware configuration record database load/unload.unix required for database loads and unloads has been deleted or renamed. It must be replaced. Refer to Unload/Load for more information.
The external file ***<WINDOWS.FILENAME> or <UNIX.FILENAME>*** **is being written to and cannot be read at this time.**	The load utility could not access the named external file because another task has a lock on it.
Invalid end of file encountered in external file ***<WINDOWS.FILENAME> or <UNIX.FILENAME>***	The load utility encountered an incomplete file. Typically, this means that the unload process was interrupted before completion.

Load error message or notification	Description
.	
All null keys:file: *<Database name>*;rec no:*<Number>* in external file: *<WINDOWS.FILENAME> or <UNIX.FILENAME>* .	The load utility attempted to add record number <Number> to the named Service Manager database and encountered all null keys.
Invalid null key;file:*<database name>*;rec no:*<Number>* in external file *<WINDOWS.FILENAME> or <UNIX.FILENAME>* .	The load utility attempted to add record number *<Number>* to the named Service Manager database and encountered an invalid null key.
Invalid duplicate key:file:*<Database name>*;rec no:*<Number>* in external file *<WINDOWS.FILENAME> or <UNIX.FILENAME>* .	The load utility attempted to add record number *<Number>* to the named Service Manager database and encountered an invalid duplicate key.
External file: %S is not in Unload Format.	The named file is not in file format required by the load utility.
You have duplicated column headers: <column names>	When duplicate source columns are detected in the external source file, warns the user about the column names.
You have specified data mapping between the target fields and source columns. This will disable RAD expressions.	If the user has specified a correct mapping between target fields and source columns, when the user specifies a RAD expression, warns the user that RAD expressions will not work in this situation.
Column not found in external file, column: <columns names>	When a specified column is not found in the source file, warns the user about the missing column.
Line <line id> is empty. Skipped the line.	When the line being processed is empty, warns the user about the line number.
Error occurs when processing line <line id>.	When an error occurs during the processing of a line, warns the user about the line number.
Record already exists on line<line id>. Updated the record.	When a record already exists during the processing of a line, warns the user about the line number.
<processed rows count> records processed.	When every 100 rows have been processed, notifies the user about the current number of rows processed.

Unload files

The native HP Service Manager export format is the unload file. An unload file stores the database dictionary of Service Manager tables in addition to records. The database dictionary embedded in an

unload file determines where Service Manager loads records during import. The unload file also specifies whether Service Manager should update or ignore duplicate records when loading records. Service Manager can export unload files in the following file formats:

- Binary — The binary unload file format archives Service Manager tables. You can load binary unload files only from the Database Manager.
- Text — The text unload file format is human readable but only intended to be loaded from the Database Manager.
- Formatted text — The formatted text unload file format is intended to be compatible with external applications, although the actual format of the file is determined by the form you select as a template. You can create export forms to create formatted text unload files compatible with specific applications.

You can export records into an unload file from the following Service Manager utilities:

- Database Manager
- Unload Script Utility
- Directly from some record lists

After you load a .unl file to import data into Service Manager, you must stop and then restart the Service Manager server. This ensures that the table changes are initialized.

Note: Service Manager encodes unload files in the UTF-8 format. Service Manager can import unload files from previous versions of HP ServiceCenter, but earlier versions cannot import unload files created by Service Manager.

Add an import descriptor record

Applies to User Roles:

System Administrator

HP Service Manager enables you to add import descriptor records for two types of text files: character-delimited, and fixed-width.

Add an import descriptor record for a character-delimited text file.

To add an import descriptor record for a character-delimited text file, follow these steps:

1. Click **Tailoring** > **Database Manager**.
2. Click **More** or the More Actions icon, and then select **Text Import Wizard**.
3. Fill in the following fields.

Field	Description
File to Import	Type or browse to the path and name of the external text file to import. **Note:** Only Windows Dos text file format is supported.
Member	This field is obsolete.
Destination Table	Type the database dictionary name of the table receiving the imported records.
File Type	Select the source operating system of external file. **Note:** The Winnt file type is recommended.

4. Click **Search**.
   HP Service Manager searches for the external file and displays one of the following messages:
   - If Service Manager finds the external file but there is not existing import descriptor record, Service Manager displays the form to create a new import descriptor.
   - If Service Manager finds the external file and also finds one or more import descriptor records, Service Manager displays a list of import descriptor records.
   - If Service Manager cannot find the external file, it displays an error message.
5. If Service Manager displays a list of import descriptor records, do one of the following:
   - Click an existing import descriptor record.
   - Click **New**.
6. From the **Creating a New Import Descriptor** form, select **Character-delimited** for **Input File Type**.
7. Click **Proceed**.
8. Fill in the following import data definitions.

Field	Description
Import Name	Type the name used to save the import descriptor record.
Import Table	This field lists the destination table you selected earlier in the wizard. This is a read-only field.
Import Mode	Select one of the following options: ◦ **Add**: Select this option to only add new records. ◦ **Add/Replace**: Select this option to both add new records and replace existing records.You may also select this option to update fields that you define in the **Advanced** tab. ◦ **Replace only**: Select this option to only replace existing records. **Note:** If you use field mapping for importing, Service Manager always use the **Add/Replace** import mode regardless of the mode you specified.
Format Control	Type the name of the optional Format Control record used to format the incoming records. **Note:** For the Format Control record, only the following sections are supported: **Calculations**, **JavaScript**, and **Validations**.
Delimiter	Service Manager only displays this field on character-delimited imports. Select one of the following options: ◦ Comma: Select this option if the import file uses a comma (,) between fields. ◦ Tab: Select this option the import file uses a tab character between fields. ◦ Semicolon: Select this option if the import file uses a semicolon (;) between fields. ◦ Other: Select this option to specify the character the import file uses between fields. **Note: Delimiter** is set to **Comma** by default. However, when you add a descriptor for the import file, if your source file uses a comma as the delimiter, you must select other delimiters first and then reselect Comma. Alternatively, you can first click **Add** to add a descriptor, and then click **Back** to return to the descriptor. You need to do so to trigger the retrieval of the source columns from the source file using the specified delimiter. Otherwise, you will see a blank Source Column list.

Field	Description
Text Qualifier	Service Manager only displays this field on character-delimited imports. Specify the text qualifier the import file uses to include special characters such as text delimiters in the record data.
Array Separator	You can specify this field to split array data in your source file. Service Manager splits the array data based on the Array Type field. Select one of the following separators for your array data: ◦ Semicolon: Select this separator if the source file uses a semicolon to split the array data. ◦ Line Break: Select this separator if the source file uses a line break to split the array data. For example, when data is stored as 'va;vb', and **Semicolon** is selected in the **Array Separator** drop-down list, Service Manager parses the value to ['va','vb'].
Field Name	Specify the database dictionary field name receiving the imported record data. Type one field name for each source field in the import file.
Source Column	This is a list of column headers in the source file. Service Manager automatically retrieves the list according to the delimiter you specified. If you specify a wrong delimiter, the source column retrieval fails. You should map each source column to a field in the target table. ◦ If the mapping table contains a field that is not mapped to a source column, all mappings are ignored. Instead, all specified fields are automatically mapped to the source columns in their display order in the source file. ◦ One source column can be mapped to multiple target fields. For example, a Service Catalog item can have the same value for **Name** and **Display Name**. **Name** | **Display Name** PC Hardware | PC Hardware If you are importing Service Catalog items into Service Manager, you can map the **Name** column in your source file to the **name** and **displayName** fields in the **svcDisplay** table in Service Manager, as shown in the following table.

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Field Name</th><th>Source Column</th></tr>
<tr><td>name</td><td>Name</td></tr>
<tr><td>displayName</td><td>Name</td></tr>
</table>

You cannot specify duplicate field names in the mapping table, that is, each field name can only occur once in the mapping table. Service Manager returns an error if it detects a duplicate field name.
Only the specified source columns are imported.

</td></tr>
<tr><td>Field Type</td><td>This is a read-only field and cannot be edited. If Service Manager validates the data from the import file, it displays a number describing the data type.</td></tr>
<tr><td>Date Format</td><td>Type the date and time format used to import date and time fields. You can use the following tokens to represent units of time:

dd: two digit day
mm: two digit month
yy: two digit year
yyyy: four digit year
hh: hours
ii: minutes
ss : seconds

When you specify a mapping between the source columns and target fields, yyyy is not supported.
When you load date type records, the date format is converted to the same as the login user’s. For example, the date format in the external file is "mm/dd/yy" with the value of "07/12/14", and defined in the mapping table. However, the login user uses a date format of "yy/mm/dd". In this case, the date format is converted to "14/07/12" and saved in Service Manager.

</td></tr>
<tr><td>Array Type</td><td>This is a read-only field and cannot be edited. If Service Manager validates any array data from the import file, it displays a number describing the array type.</td></tr>
<tr><td>Occurrenc</td><td>Type the number of array elements to import for this field.</td></tr>
</table>

Field	Description
es	
Query	Type a query used to select an existing record from the destination Service Manager database for updating. If this query returns a single record, Service Manager updates this record with the data from the import file. If the query returns more than one record, then Service Manager uses the **Import Mode** setting to determine what action to take. **Note:** The import utility stores the values of imported fields in the `$L.db` variable. You can use this variable to construct queries. For example: `not null(name in $L.bd)`
Skip	Type a query that identifies records to skip when adding or updating from the import file. If this query returns one or more records, Service Manager ignores adds or updates from records matching the query. To use this field you must also provide an entry in the **Query** field. **Note:** The import utility stores the values of imported fields in the `$L.db` variable. You can use this variable to construct a skip condition query. For example: `null(name in $L.bd)`
Expressions	Type any RAD expressions used to identify records for importing. **Note:** If you have specified a correct mapping between the target fields and source columns, RAD expressions are disabled.
Javascript	Specify Javascript code to do advanced import. The JavaScript expressions are executed against each record in the source file. For example, when importing your Outlook contacts data into the **contacts** table in Service Manager, you can combine the First Name and Last Name, and then import the combined value into the **contact.name** field in the **contacts** table. **First Name** | **Last Name** John | Smith After the combination, a contact record **John, Smith** is inserted into the **contacts** table in Service Manager. To achieve this, you can define the following statement in the **JavaScript** field:

Field	Description
	`target['contact.name'] = source['First Name'] + ',' + source['Last Name'];` where the **target** object represents the Service Manager target field, and the **source** object represents one record in the source file.
Update Field Names	Type the list of field names from the destination table used to uniquely identify a record. If Service Manager finds existing records matching these fields, it updates the record with the data from the import file.
Case Conversion	Select one of the following options to convert record data to a specific case: ◦ None: Select this option if you do not want to convert the case of imported data. ◦ Upper case: Select this option to convert imported data into all upper case. ◦ Lower case: Select this option to convert imported data into all lower case.
Skip Lines	Specify the optional number of lines to skip at the beginning of the file. **Note:** When you specify a mapping between the source columns and target fields, you must set **Skip Lines** to **1** which means the column headers is specified as the first line.

9. Click **Add**.

Add an import descriptor record for a fixed-width text file.

To add a fixed-width import descriptor record, follow these steps:

1. Click **Tailoring** > **Database Manager**.
2. Click **More** or the More Actions icon, the form and select **Text Import Wizard**.
3. Fill in the following fields.

Field	Description
File to Import	Type or browse to the path and name of the external text file to import.
Member	This field is obsolete.
Destination Table	Type the database dictionary name of the table receiving the imported records.

Field	Description
File Type	Select the source operating system of external file.

4. Click **Search**.
   HP Service Manager searches for the external file and displays one of the following messages:
   - If Service Manager finds the external file but there is not existing import descriptor record, Service Manager displays the form to create a new import descriptor.
   - If Service Manager finds the external file and also finds one or more import descriptor records, Service Manager displays a list of import descriptor records.
   - If Service Manager cannot find the external file, it displays an error message.
5. If Service Manager displays a list of import descriptor records, do one of the following:
   - Click an existing import descriptor record.
   - Click **New**.
6. From the **Creating a New Import Descriptor** form, select **Fixed-width** for **Input File Type**.

Field	Description
Fixed-width	Select this option if each field in the file occupies a fixed-width position. For example, the following are two records using fixed width positions. The number field starts at position 1 and ends at position 6. The open.time field starts at position 8 and ends at position 15.

Position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Record 1	I	M	1	0	4	2		1	2	/	2	9	/	0	0
Record 2	I	M	1	0	5	8		1	2	/	2	9	/	0	0

7. Click **Proceed**.
8. Fill in the following import data definitions.

Field	Description
Import Name	Type the name of the new import descriptor record.

Field	Description
Import Table	This field lists the destination table you selected earlier in the wizard. This is a read-only field.
Import Mode	Select one of the following options: ◦ Add: Select this option to only add new records. ◦ Add/Replace: Select this option to both add new records and replace existing records. You may also select this option to update fields that you define in the **Advanced** tab. ◦ Replace only: Select this option to only replace existing records.
Format Control	Type the name of the optional Format Control record used to format the incoming records.
Field Names	Specify the database dictionary field names receiving the imported record data. Type one field name for each source field in the import file.
Field type	This is a read-only field and cannot be edited. If Service Manager validates the data from the import file, it displays a number describing the data type.
Start Position	Service Manager only displays this field on fixed-width imports. Type the column number where data for this field starts.
Field Length	Service Manager only displays this field on fixed-width imports. Type the number columns that the data from this field uses.
Array Type	This is a read-only field and cannot be edited. If Service Manager validates any array data from the import file, it displays a number describing the array type.
Occurrences	Type the number of array elements to import for this field.
Date Format	Type the date and time format used to import date and time fields. You can use the following tokens to represent units of time: ◦ dd: two digit day ◦ mm: two digit month ◦ yy: two digit year ◦ yyyy: four digit year ◦ hh: hours ◦ ii: minutes ◦ ss: seconds You can use any character to separate these tokens. For example, mm/dd/yyyy

Field	Description
	hh:ii:ss imports dates and times with slashes between the month, day, and four-digit year and colons between the hours, minutes, and seconds.
Query	Type a query to select one existing record from the destination Service Manager database for updating. If this query returns a single record, Service Manager updates this record with the data from the import file. If the query returns more than one record, then Service Manager uses the **Import Mode** setting to determine what action to take.
Skip	Type a query to identify records to skip when adding or updating from the import file. If this query returns one or more records, Service Manager ignores adds or updates from records matching the query.
Expressions	Type any RAD expressions used to identify records for importing.
Update Field Names	Type the list of field names from the destination table to uniquely identify a record. If Service Manager finds existing records matching these fields, it updates the record with the data from the import file.
Case Conversion	Select one of the following options to convert record data to a specific case: ◦ None: Select this option if you do not want to convert the case of imported data. ◦ Upper case: Select this option to convert imported data into all upper case. ◦ Lower case: Select this option to convert imported data into all lower case.
Skip Lines	Specify the optional number of lines to skip at the beginning of the file.

9. Click **Add**.

Delete or update an import descriptor record.

To delete or update an import descriptor record, follow these steps:

1. Click **Tailoring** > **Database Manager**.
2. Type `=import` in the **Form** field.
3. Select **Administration Mode**. You must select this option for HP Service Manager to display the **Delete** button.
4. Click **Search**.
5. Click the form where both the **Format Name** and **File Name** equal **import**.
6. Click **Search** again to display a list of import descriptor records.

Note: You may need to click **Document** > **Restore** to see the list of import descriptor records.

7. Select the import descriptor record you want to update or delete.
8. To update the descriptor record, update the fields and then click **Save**.
9. To delete the descriptor record, click **Delete**, and then select **Yes**.

Import a character-delimited text file

Applies to User Roles:

System Administrator

Before you import a character-delimited text file, you need to prepare your source file and add a character-delimited import descriptor record first.

To import a character-delimited text file, follow these steps:

1. Prepare the source file. The following table describes the rules that external source files must observe.

Rule	Note
Only plaint text files are supported.	Be sure to not load binary files, such as .JPG and .XLS files.
If the source data contains non-ASCII characters, the external text file must use UTF-8 or Unicode encoding.	
Column headers in the external file are mandatory and must be unique.	CSV files with duplicate column headers cannot be imported. The wizard displays a warning on the duplicate columns and is not able to proceed. Make sure your source file contains column headers on the first line, and remove or rename duplicate columns as needed.
Column headers in source files should follow the JavaScript Identifier Naming conventions.	Column headers in source files are used as JavaScript identifiers in this feature.

Rule	Note
If the number of values in a row does not match the number of the column headers, the row is skipped. For example, if your source file contains 6 column headers, a row with more or less than 6 values is skipped.	Make sure your source file does not contain invalid rows.
When a source file is huge (more than 15, 000 lines or 10 MB in size), it may take a long time to load the file.	You are recommended to split a huge file to smaller ones before loading.
Digit grouping is not supported and only numbers in raw format can be imported.	For example, "1, 000.000, 3" cannot be correctly recognized. You must change it to "1000.0003".
Unique keys are not automatically generated during the import operation.	Make sure that the records to import match the data constraints that are defined in their dbdict.
Delimiters in external CSV files can be one of the following: ◦ a comma ◦ a tab ◦ a semicolon ◦ a user-defined character other than listed above	When a field value contains the delimiter character, enclose the value in double quotes. For example: **"abc,def"**
The following date formats are supported for external source files: ◦ mm/dd/yy [hh:ii:ss] ◦ yy/mm/dd [hh:ii:ss] ◦ dd/mm/yy [hh:ii:ss] **Note:** [hh:ii:ss] is optional.	Make sure all of your source data uses the correct date format. You will need to specify the date format later when adding an import descriptor for the source file. **Note:** After importing, the imported data is converted to the login user's date format.

2. Resolve necessary validations or dependencies first.

Note: Triggers are enabled during importing. For this reason, you may need to resolve some

validations or dependencies first. For example, before loading a contact record, an operator record should be loaded first. Otherwise, a message, for example, “No operator record to sync”, pops up.

3. Launch the Text Import Wizard, and add an import descriptor record for the text file. For details, see "Add an import descriptor record" on page 174.
4. Click **Test** to test the importing of records.
5. Review the test results on the **Test Import** tab.

   Note: Only the first ten lines in the external file are tested.

6. Click **Proceed**.
7. Click **Run**.
   Service Manager imports the external file.

   - Scheduled import is not supported for character-delimited files.
   - Some joinfiles need to be loaded completely to achieve desired data import results. For example, when loading service catalog items, you need to load the service catalog display records concurrently. Otherwise, the imported records are not displayed correctly in the catalog.

Example: Import Outlook contacts into the contacts table

The following example describes how to import your Outlook contacts into the contacts table.

1. In Outlook 2010, export your contacts to a `.csv` file. To do this, follow these steps:
   a. Click **File** > **Open** > **Export** to open the Import and Export wizard form.
   b. Select **Export to a file**, and then click **Next**.
   c. Select **Comma Separated Values (Windows)**, and then click **Next**.
   d. Select **Contacts**, and then click **Next**.

e. Click **Browse** to specify the location and file name for your .csv file, and then click **Next.**

f. Click **Finish**.

2. In Service Manager, type `db` in the command line field, and then press **Enter**. Database Manager opens.

3. Click **More** or the More Actions menu, and then select **Text Import Wizard**. The Import wizard form opens.

4. In the **File to import** field, browse to your `.csv` file.

5. In the **Destination table** drop-down list, type or select `contacts`.

6. In the **File Type** drop-down list, select **winnt**.

7. Click **Search**.

8. If a descriptor for the contacts table exists, click **New** to create a descriptor for your import. Otherwise, the Creating a New Import Descriptor page automatically opens.

   Note: The **Import Table** field is automatically populated with contacts.

9. In the **Input File Type** field, select **Character-delimited**, and then click **Proceed**.

10. In the **Import Name** field, specify a name for the descriptor (for example, Outlook Contacts Import), and then click **Add**.

    Note: If you want to delete an existing descriptor, see "Add an import descriptor record" on page 174

11. In the **Import Mode** drop-down list, select **Add/Replace**.

12. In the **Delimiter** radio button, select **Other**, and then select **Comma**. This is because your Outlook contacts were exported as Comma Separated Values. If this is not the case, specify another delimiter that was used to export your Outlook data.

    Note: The default Delimiter is Comma. However, you need to select another value and then select Comma again to activate the retrieval of the source columns in your .csv file.

13. In the **Format Control** drop-down list, select **fcContacts**.

14. On the **Fields** tab, complete the field mapping table. The following is an example.

Field	Source Column
title	Job Title
contact.name	First Name
email	E-mail Address
company	Company
manager	Manager's Name
location	Location
dept	Department
portable.phone	Mobile Phone

Note: Only the specified source columns are imported. Make sure that you follow these rules, otherwise, errors occur. You cannot specify a duplicate field in the destination table. However, you can map a source column to multiple fields. A field cannot have an empty source column mapped to it.

15. On the **Advanced** tab, add the following lines to the JavaScript box:

```
target['contact.name']=source['First Name']+'.'+source['Last Name'];
target['company']='HP1';
```

16. Click **Save**.

17. Click **Proceed**.

18. Click **Run**.

    The import starts. When it has finished, a message is displayed indicating how many records have been successfully inserted. You can go to the contacts table to check the import results.

Import a fixed-width text file

Applies to User Roles:

System Administrator

Before you import a fixed-width text file, you need to add a fixed-width import descriptor record first.

To import a fixed-width text file, follow these steps:

1. Add an import descriptor for the fixed-width text file. For details, see "Add an import descriptor record" on page 174
2. Click **Test** to test the importing of records.
3. On the **Test Import** tab, review the test results.

   Note: Only the first ten lines in the external file are tested.
4. Click **Proceed**.
5. Click **Run**.
   Service Manager imports the external file.

Load an unload file

Applies to User Roles:

System Administrator

Tip: If your application version is 9.30 ap3, 9.31 or later, you are recommended to use Unload Manager to load an unload file, because Unload Manager can help you create a backup of your old data and reconcile conflicts during the installation of the unload; if your application version is other than any of these, Unload Manager is not available and you can use Database Manager instead.

To load an unload file using Unload Manager, follow these steps:

1. Go to **System Administration** > **Ongoing Maintenance** > **Unload Manager**.
2. Double-click **Apply Unload**. A wizard opens.
3. Select the unload file you want to apply, also specify a backup file, and then click **Next**. Details of the unload file appear.
4. Double-click a conflicting object in the table to open the merge tool:
   a. Merge the object, and then select the **Reconciled** check box.
   b. Click **Save** to go back to the wizard.

5. Click **Next** after all the conflicting objects are reconciled.
6. Click **Yes** on the confirmation window to apply the unload.
7. Click **Finish**.

   Now, the unload has been applied and at the same time your old data backed up.

To load an unload file using Database Manager, follow these steps:

1. Click **Tailoring** > **Database Manager**.
2. Click **More** or the More Actions menu, and select **Import/Load**.
3. Fill in the following fields.

Field	Description
File Name	Type the name and path of the file to load.
Import Descriptor	Type the name of the optional Import Descriptor record used to import records. Since unloads files do not require an Import Descriptor record, leave this field blank.
File Type	Select the source operating system of the unload file.
Messages Option — All Messages	Select this option to see all messages that HP Service Manager generates loading the file.
Messages Option — Totals Only	Select this option to see only the total number of files Service Manager loads.
Messages Option — None	Select this option to hide all messages that Service Manager generates loading the file.

   Note: You can view the contents of an unload file before importing it by clicking **List Contents**.

4. Click **Load FG**.

Schedule the importing of records using an import descriptor record

Applies to User Roles:

System Administrator

Before you schedule the importing of records, you may want to add an import descriptor record first, refer to "Add an import descriptor record" on page 174.

To schedule the importing of records using an import descriptor record, follow these steps:

1. Open the import descriptor record you want to schedule, and then click **Proceed**.
2. In the **Expiration** field, type the date and time when the scheduled import runs.
3. To repeat at regular intervals, select one of the following options.

Field	Description
Repeat — Monthly	Select this option to repeat the import process every once every month from the Expiration date.
Repeat — Quarterly	Select this option to repeat the import process every once every quarter from the Expiration date.
Repeat — Semi-Annually	Select this option to repeat the import process every once every six months from the Expiration date.
Repeat — Annually	Select this option to repeat the import process every once every year from the Expiration date.

4. Click **Schedule**.

Schedule the loading of an unload file

Applies to User Roles:

System Administrator

To schedule the loading of an unload file:

1. Click **Tailoring** > **Database Manager**.
2. Click **More** or the More Actions icon, and then select **Import/Load**.
3. Fill in the following fields.

Field	Description
File Name	Type the name and path of the file to load.
Member	This field is obsolete.
Import Descriptor	Type the name of the Import Descriptor record used to import records. Because unload files do not require an Import Descriptor record, leave this field blank.
File Type	Select the source operating system of unload file.
Messages Option — All Messages	Select this option to see all messages that HP Service Manager generates loading the file.
Messages Option — Totals Only	Select this option to see only the total number of files Service Manager loads.
Messages Option — None	Select this option to hide all messages that Service Manager generates loading the file.

4. Click **Load BG**.

5. Fill in the following fields.

Field	Description
Expiration Date	Specify the date and time the scheduled process runs. Service Manager loads the file at this date and time. By default, Service Manager uses the current date and time.
Repeat Interval	Select the time period to wait before repeating the load process. Type the interval in the following format: *<days><hours>:<minutes>:<seconds>*. For example, 1 00:00:00 is a repeat interval of 1 day.
External Filename	Type the name and path to the external file to load. By default, Service Manager looks for unload files in the server RUN folder.
Member	This field is obsolete.
Suppress Messages?	Type true or false to show or hide messages Service Manager generates loading the external file.

6. Click **Load BG**.

View the contents of an unload file

Applies to User Roles:

System Administrator

To view the contents of an unload file:

1. Click **Tailoring** > **Database Manager**.
2. Click **More** or the More Actions icon, and then select **Import/Load**.
3. Fill in the following fields.

Field	Description
File Name	Type the name and path of the file you want to load.
Member	This field is obsolete.
Import Descriptor	Type the name of an Import Descriptor record used to import records. Because unload files do not require an Import Descriptor record, you can leave this field blank.
File Type	Select the source operating system of the unload file.
Messages Option — All Messages	Select this option to see all messages that HP Service Manager generates while loading the file.
Messages Option — Totals Only	Select this option to see only the total number of files Service Manager loads.
Messages Option — None	Select this option to hide all messages that Service Manager generates loading the file.

4. From the More Actions menu, click **List Contents**.

Unload script utility

The unload script utility enables system administrators to create HP Service Manager unload files automatically. The unload script utility enhances the standard unload creation process in many ways. With unload scripts, you can:

- Save records from multiple tables into a single unload file.
- Specify a query for each source table that filters the records added to the unload file.
- Purge records during the unload process.
- Specify which formats to protect during a purge process.
- Add related records from the data map file to the unload file.

The unload script utility is available in Tailoring. By default, Service Manager includes a collection of unload scripts that you can use for common unload tasks. You can also use the default unload scripts as templates to create your own customized unload scripts.

To create and use unload scripts effectively, you must be familiar with the Service Manager Database Manager. You can create a query on any field in the file, but querying keyed fields improves system performance and response time. You can create record queries using Structured Query Language (SQL) syntax.

You can use the Purge/Archive utility to unload and purge records from one file; if you use datamaps, you can use Purge/Archive to unload and purge related records as well.

After you load a .unl file to import data into Service Manager, you must stop and then restart the Service Manager server. This ensures that the table changes are initialized.

Note: The Unload Script utility unloads tables using the binary unload file format.

Add an unload script record

Applies to User Roles:

System Administrator

To add an unload script record:

1. Click **Tailoring** > **Unload Script Utility**.
2. Fill in the following fields.

Field	Description
Unload Script	Type a name or label to identify your unload script. This name can include spaces.

Field	Description
Unload?	Select this option to create an external unload file from the records queried by this script.
Purge?	Select this option to delete the records queried by this script.
Show Unload Records	Select this option to display the **Filename**, **Query**, and **Data Map** fields where you can specify the tables and queries for unloading or purging records.
Filename	Specify the name of the tables unloading or purging records. This field is visible only if you select **Show Unload Records**.
Query	Type the SQL query to identify the records to unload or purge. You can use Service Manager operators and variables in this field. This field is visible only if you select **Show Unload Records**.
Datamap	Specify true or false to specify whether Service Manager should use the data map file to unload or purge any associated records. You must have previously defined a data map for the listed file in order to use this feature. By default, Service Manager reads a blank entry in this field as false. This field is only visible if you select **Show Unload Records**.
Show Protected Formats	Select this option to display the Protected Formats field where you can specify the forms to protect from purging.
Protected Formats	Type the names of the forms to protect purging records.

3. Click **Add**.

Delete an unload script record

Applies to User Roles:

System Administrator

To delete an unload script record:

1. Click **Tailoring** > **Unload Script Utility**.
2. Fill in optional search criteria, and then click **Search**.

3. Select the unload script to be deleted, and then click **Delete**.
4. Click **Yes** to confirm the deletion.

Protect forms from the unload script utility

Applies to User Roles:

System Administrator

You can protect a form from being purged by the unload script utility by explicitly listing the form in the protected formats list.

1. Click **Tailoring** > **Unload Script Utility**.
2. Fill in optional search criteria, and then click **Search**.
3. Select the unload script to update.
4. Select the **Show Protected Formats** options.
5. In the **Protected Formats** field, type the names of the forms to protect when you purge records from this script.
6. Click **Save**.

Purge records with the unload script utility

Applies to User Roles:

System Administrator

To purge records with the unload script utility:

1. Click **Tailoring** > **Unload Script Utility**.
2. Fill in optional search criteria, and then click **Search**.
3. Select the unload script to run.
4. Select the **Purge?** option.
5. Select the **Show Protected Formats** options.

6. In the **Protected Formats** field, type the names of the forms to protect when you purge records from this script.
7. Click **Proceed**.
8. Click **Yes**.

Run an unload script

Applies to User Roles:

System Administrator

To run an unload script:

1. Click **Tailoring** > **Unload Script Utility**.
2. Fill in optional search criteria.
3. Click Search.
4. Select the unload script to run.
5. Click **Proceed**.
6. Fill in the following fields.

Field	Description
External File Name	Type the name of the unload file to create. This name can include the path information but must use characters that are valid for the target operating system. By default, HP Service Manager saves the unload file in RUN folder of the server. If users have enabled client-side unloading, then the Service Manager saves the unload file on their Windows desktop.
Append to File?	Select this option to add records to an existing unload file.
When loading records into an existing database dictionary — Use existing database dictionary	Select this option to use the table and field definitions stored in the database dictionary of the target system.

Field	Description
When loading records into an existing database dictionary — Use database dictionary of loaded record	Select this option to use the table and field definitions stored in the unload file. This option will overwrite the existing database dictionary, if any, for the loaded file and records.
When loading records — Add new records and update existing records	Select this option to update existing records in the target system with records from the unload file. The new records will be inserted into the table and the existing records will be updated regardless of case-sensitivity.
When loading records — Add new records only	Select this option to ignore existing records and add only new records from the unload file. The new records will be inserted into the table regardless of case-sensitivity.

7. Click **Proceed**.
   Service Manager unloads the tables and records you selected in the unload script.

After you load a .unl file to import data into Service Manager, you must stop and then restart the Service Manager server. This ensures that the table changes are initialized.

Note: The unload script utility unloads tables using the binary unload file format.

Update an unload script record

Applies to User Roles:

System Administrator

To update an unload script record:

1. Click **Tailoring** > **Unload Script Utility**.
2. Fill in optional search criteria, and then click **Search**.
3. Select the unload script to update.
4. Specify the new unload script settings:

Field	Description
Unload Script	Type a name or label to identify your unload script. This name can include spaces.
Unload?	Select this option to create an external unload file from the records queried by this script.
Purge?	Select this option to delete the records queried by this script.
Show Unload Records	Select this option to display the Filename, Query, and Data Map fields. These fields enable you to specify the tables and queries to unload or purge records.
Filename	Specify the name of the table used to unload or purge records. This field is visible only if you selected **Show Unload Records**.
Query	Type the SQL query to select records for unloading or purging. You can use HP Service Manager operators and variables in this field. This field is visible only if you selected **Show Unload Records**.
Datamap	Specify true or false to specify whether Service Manager should use the data map file to unload or purge any associated records. You must have previously defined a data map for the listed file in order to use this feature. By default, Service Manager reads a blank entry in this field as false. This field is only visible if you have selected the Show Unload Records option.
Show Protected Formats	Select this option to display the Protected Formats field. Specify the forms to protect from purging.
Protected Formats	Type the names of the forms to protect when you purge records from this script.

5. Click **Save**.

Validate an unload script record

Applies to User Roles:

System Administrator

To validate an unload script record:

1. Click **Tailoring** > **Unload Script Utility**.
2. Use search or advanced search to find one or more records.

3. Select the unload script to validate.
4. Click **More** or the More Actions icon for the record, and then select **Validate**.

Purging and archiving records

The purge and archive function lets you save HP Service Manager records to an external file and then delete those records from the database. You can use the purge and archive function to archive data as part of your backup plan or to free up space on your system. You can access the purge and archive function manually or as part of a scheduled process.

You can use the purge and archive function in one of four ways:

- Archive records only
- Purge records only
- Archive and then purge records
- Schedule an archive, a purge, or an archive and purge process

You can also use the purge and archive function in conjunction with data maps. When you define a data map, you declare certain records as having a mapping or connection to other tables and records. If you have created data maps, you can enable them during the purge and archive process to determine what additional records Service Manager should include in the process.

Unload files

The native HP Service Manager export format is the unload file. An unload file stores the database dictionary of Service Manager tables in addition to records. The database dictionary embedded in an unload file determines where Service Manager loads records during import. The unload file also specifies whether Service Manager should update or ignore duplicate records when loading records. Service Manager can export unload files in the following file formats:

- Binary — The binary unload file format archives Service Manager tables. You can load binary unload files only from the Database Manager.
- Text — The text unload file format is human readable but only intended to be loaded from the Database Manager.

- Formatted text — The formatted text unload file format is intended to be compatible with external applications, although the actual format of the file is determined by the form you select as a template. You can create export forms to create formatted text unload files compatible with specific applications.

You can export records into an unload file from the following Service Manager utilities:

- Database Manager
- Unload Script Utility
- Directly from some record lists

After you load a .unl file to import data into Service Manager, you must stop and then restart the Service Manager server. This ensures that the table changes are initialized.

Note: Service Manager encodes unload files in the UTF-8 format. Service Manager can import unload files from previous versions of HP ServiceCenter, but earlier versions cannot import unload files created by Service Manager.

Using data maps to purge and archive records

You can use data maps to purge or archive related records in your database. A data map defines connections between various tables in your database. For example, when you create a record in the probsummary table, you typically create associated records in the problem and screlation tables. By default, HP Service Manager comes with a collection of data maps to common tables. For example, the following tables have default data maps defined in Service Manager:

Table	Table associated by a data map
probsummary	problem, screlation
cm3r	cm3t, screlation
ocmo	ocmalertlog, ocmapprlog, ocml, ocmlpage, ocmopage, ocmphaselog
ocmq	ocmalertlog, ocmapprlog, ocml, ocmlpage, ocmphaselog, ocmqpage

To purge or archive all records associated with a particular table quickly, enable **Use Data Map?** from the Purge and Archive function. You can add or update data maps to create new associations among records.

Archive records

Applies to User Roles:

System Administrator

To archive records:

1. Click **System Administration** > **Ongoing Maintenance** > **System** > **Purge/Archive**.
2. Fill in the following fields:

Field	Description
Enter Format Name	Type the name of the form to archive.
Use Data Map?	Type true or false to enable the data map associated with this form. If enabled, HP Service Manager archives all associated records with the form you select.
Prompt Data Map?	Type true or false to enable a confirmation message to use a data map for each record archived. This option is only valid if the **Use Data Map?** option is also enabled.

3. Click **OK**.
4. Fill in optional search criteria.
5. Click **Query**.
6. Click **Execute Search**.
7. Click **Archive**.
8. Fill in the following fields:

Field	Description
External File Name	Type the name and path of the file used to save the records. By default, this file is in the RUN folder of the Service Manager server.

Field	Description
Member	This field is obsolete.
Unload Type	Select the operating system file format of the external file.
Append to File	Select this option to add records to the end of an external file of the same name. If an external file of the same name does not exist, Service Manager creates a new file. If this field is cleared, then Service Manager overwrites any existing file of the same name.
Use Data Map	Select this option to use a predefined data map when archiving the selected records. Using a data map may result in Service Manager archiving more records than what you originally selected by the query.
Prompt for Data Map	Select this option to enable a confirmation message to use a data map for each record archived. This option is only valid if the **Use Data Map** option is also enabled.
Suppress Messages	Select this option to hide all messages that the purge and archive function generates.
Export Mode — Binary	Select this option to save the external file in Service Manager's binary format. Only the Service Manager Database Manager can read this binary file format. You must select this option for archiving records.
Export Mode — Text	Select this option to save the external file in Service Manager's text format. This format is human readable but is not intended for export to other applications other than Service Manager's Database Manager. You cannot select this option for archiving records.
Export Mode — Formatted Text	Select this option to save the external file in a user-defined text format. This format is intended for export to other applications. If you select this option, you must type a valid text format name in the **Form Name** field. You cannot select this option for archiving records.
Form Name	Type the name of the form to use when exporting records as formatted text. This option is valid only when you select the **Formatted Text** option. If the text form does not already exist, Service Manager prompts you to create it.
Dbdict Load Options — Create file if not there, and update if present	Select this option if the data saved in this external file should replace existing data in the target Service Manager system.
Dbdict Load Options —	Select this option to ignore data saved in the external file that

Field	Description
Create file if not there, but do not update if present	already exists in the target Service Manager system.
Dbdict Load Options — Create file if not there, and update if present, including keys	Select this option if the data saved in this external file should replace existing data in the target Service Manager system including the database keys.
Dbdict Load Options — Create file if not there, and update if present, including SQL mapping	Select this option if the data saved in this external file should replace existing data in the target Service Manager system including the database SQL mapping.
Dbdict Load Options — Create file if not there, and update if present, including keys AND SQL mapping	Select this option if the data saved in this external file should replace existing data in the target Service Manager system including the database SQL mapping and the database keys.
Record Load Options — Add/update records	Select this option if the records saved in this external file are to replace any existing records in the target Service Manager system.
Record Load Options — Only add new records	Select this option to ignore records saved in the external file that already exist in the target Service Manager system.

9. Click **unload/export**.
   Service Manager saves the select records to an external file.

Purge and archive records

Applies to User Roles:

System Administrator

To purge and archive records:

1. Click **System Administration** > **Ongoing Maintenance** > **System** > **Purge/Archive**.

2. Fill in the following fields.

Field	Description
Format Name	Type the name of the form to purge and archive.
Use Data Map?	Type true or false to enable the data map associated with this form. If enabled, HP Service Manager archives all associated records with the form you select.
Prompt Data Map?	Type true or false to enable a confirmation message to use a data map for each record purged and archived. This option is only valid if the **Use Data Map?** option is also enabled.

3. Click **OK**.
4. Fill in optional search criteria to select the records to purge and archive.
5. Click **Query**.
6. Click **Execute Search**.
7. Click **Purge/Arch**.
8. Fill in the following fields.

Field	Description
External File Name	Type the name and path of the file used to save the records. By default, this file is located in the RUN folder of the Service Manager server.
Member	This field is obsolete.
Unload Type	Select the operating system file format for the external file.
Append to File	Select this option to add records to the end of an external file of the same name. If a external file of the same name does not exist, Service Manager creates a new file. If this field is cleared, then Service Manager overwrites any existing file of the same name.
Use Data Map	Select this option to use a predefined data map when archiving the selected records. Using a data map may result in Service Manager archiving more records than what you originally selected by the query.
Prompt for Data Map	Select this option to enable a confirmation message to use a data

Field	Description
	map for each record archived. This option is only valid if the **Use Data Map** option is also enabled.
Suppress Messages	Select this option to hide all messages that the purge and archive function generates.
Export Mode — Binary	Select this option to save the external file in Service Manager's binary format. Only the Service Manager Database Manager can read this binary file format. You must select this option for archiving records.
Export Mode — Text	Select this option to save the external file in Service Manager's text format. This format is human readable but is not intended for export to other applications other than Service Manager's Database Manager. You cannot select this option for archiving records.
Export Mode — Formatted Text	Select this option to save the external file in a user-defined text format. This format is intended for export to other applications. If you select this option, you must type a valid text format name in the **Form Name** field. You cannot select this option for archiving records.
Form Name	Type the name of the text format to use when exporting records as formatted text. This option is only valid with the **Formatted Text** option is selected. If the text format description does not already exist, Service Manager prompts you to create it.
Dbdict Load Options — Create file if not there, and update if present	Select this option if the data saved in this external file should replace existing data in the target Service Manager system.
Dbdict Load Options — Create file if not there, but do not update if present	Select this option to ignore data saved in the external file that already exists in the target Service Manager system.
Dbdict Load Options — Create file if not there, and update if present, including keys	Select this option if the data saved in this external file should replace existing data in the target Service Manager system including the database keys.
Dbdict Load Options — Create file if not there, and update if present, including SQL mapping	Select this option if the data saved in this external file should replace existing data in the target Service Manager system including the database SQL mapping.
Dbdict Load Options —	Select this option if the data saved in this external file should

Field	Description
Create file if not there, and update if present, including keys AND SQL mapping	replace existing data in the target Service Manager system including the database SQL mapping and the database keys.
Record Load Options — Add/update records	Select this option if the records saved in this external file are to replace any existing records in the target Service Manager system.
Record Load Options — Only add new records	Select this option to ignore records saved in the external file that already exist in the target Service Manager system.

9. Click **unload/export**.
10. Click **Purge All**.

Purge records

Applies to User Roles:

System Administrator

To purge records:

1. Click **System Administration** > **Ongoing Maintenance** > **System** > **Purge/Archive**.

   Fill in the following fields.

Field	Description
Format Name	Type the name of the form to purge.
Use Data Map?	Type true or false to enable the data map associated with this form. If enabled, HP Service Manager purges all associated records with the form you select.
Prompt Data Map?	Type true or false to enable a confirmation message to use a data map for each record purged. This option is only valid if the **Use Data Map?** option is also enabled.

2.
3. Click **OK**.

4. Fill in optional search criteria.
5. Click **Query**.
6. Click **Execute Search**.
7. Click **Purge**.
8. Click **Purge All**.

Schedule purge and archive processes

Applies to User Roles:

System Administrator

To schedule purge and archive processes:

1. Click **System Administration** > **Ongoing Maintenance** > **System** > **Purge/Archive**.
2. Fill in the following fields.

Field	Description
Enter Format Name	Type the name of the form to purge and archive.
Use Data Map?	Type yes or no to enable the data map associated with this form. If enabled, HP Service Manager archives all associated records with the form you select.
Prompt Data Map?	Type yes or no to enable a confirmation message to use a data map for each record purged and archived. This option is only valid if the **Use Data Map?** option is also enabled.

3. Click **OK**.
4. Fill in optional search criteria.
5. Click **Query**.
6. Click **Execute Search**.

7. Click **Schedule**.

8. Fill in the following fields.

Field	Description
Date/Time to Run	Specify the date and time to run the archive, purge, or purge and archive process.
Repeat Interval — Enter Number of Days and Time	Specify the number of days, hours, minutes, and seconds to wait, if any, before repeating the archive, purge, or purge and archive process. Type the interval in the following format: *<days><hours>:<minutes>:<seconds>*. For example, 1 00:00:00 is a repeat interval of 1 day.
Repeat Interval — Monthly	Select this option to repeat the archive, purge, or purge and archive process every once every month from today's date.
Repeat Interval — Quarterly	Select this option to repeat the archive, purge, or purge and archive process every once every quarter from today's date.
Repeat Interval — Semi-Annually	Select this option to repeat the archive, purge, or purge and archive process every once every six months from today's date.
Repeat Interval — Annually	Select this option to repeat the archive, purge, or purge and archive process every once every year from today's date.
Archive Only — Ext File	Type the name and path of the external file to contain archive records.
Archive Only — Member	This field is obsolete.
Archive Only — Append to existing file or create new file (append/create)	Select **Append** to add records to the end of an external file of the same name. Select **Create** to overwrite an external file of the same name. With either option, if an external file of the same name does not exist, Service Manager creates a new file.
Use Data Map?	Type yes or no to enable the data map associated with this form. If enabled, Service Manager archives all associated records with the form you select.
Suppress Messages	Select this option to hide all messages that the purge and archive function generates.
Dbdict load options include	Select one of the following options: ◦ Create — Select this option to create a new file if one is not present in the target Service Manager system and to preserve the existing file if there is a duplicate file in the external file.

Field	Description
	◦ Replace— Select this option to create a new file if one is not present in the target Service Manager system and to overwrite any existing files with the entry in the external file. ◦ Remove — Select this option to create a new file if one is not present in the target Service Manager system and to replace any existing file with the entry in the external file.
Record load options include	Select one of the following options: ◦ Add/Update — Select this option to create a new records if they are not present in the target Service Manager system and to overwrite existing records with entries from the external file. ◦ Add Only — Select this option to create a new records if they are not present in the target Service Manager system and to preserve any existing records if there duplicates entries the external file. The records will be updated or inserted into the table regardless of case-sensitivity.

9. Click one of the following options:
   - **Archive**
   - **Purge**
   - **Purge/Arch**

Master data records

Master data are key data that are shared by all Service Manager applications, and are stored as records in support tables. Master data are often provided by functions outside of Information Technology, such as Human Resource Management, Finance, and Facilities.

As a System Administrator, you can create, update, and delete master data records from the **System Administration** > **Base System Configuration** menu.

Note: In ITIL best practices, a Configuration Administrator manages master data. In Service Manager, however, only a System Administrator has rights to master data, because the master data are shared by all Service Manager applications. You may need to contact a System Administrator to implement master data changes or to request permission to perform this function.

Master data record name	Purpose
Companies	Contains contact information and preferences about the companies you work with.
Contacts	Contains contact information about users who contact the service desk to initiate a service desk interaction, incident, or change, or a person who uses components tracked in Configuration Management. This is a generally larger group than the Service Manager operators, but may include them.
Conversion Rates	Contains the conversion rates for international currencies.
Currencies	Contains the ISO codes and formats for international currencies. These records specify whether an individual currency uses European Union Currency (EUR) as its root.
Departments	Contains identifying information and service level agreements for company departments.
Fax	Contains device name and address information for your office fax configurations.
Holidays	Contains holiday dates by group (worldwide, unique to United States, etc.)
Locations	Contains address and organization information about the locations in your company. You can define the site category with the following types: Critical Site, Major Site, Satellite Site, Home Site.
Models	Contains a detailed bill of materials for a part. This includes the hardware, software and services that comprise the item, as well as controls on the part and its components. The record contains information regarding what operators can do with the part, how components of this part are selected, how quote line items turn into orders, and whether to order or to consume from stock.
Standard File	Contains file configuration information, which defines file handling (import, export, etc.) for different operating systems.
Vendors	Contains manufacturer and retail vendor/supplier information. It also holds data on internal and external service providers. A vendor record for a particular manufacturer must exist before items from the manufacturer can be admitted to the Configuration Management and Request Management catalogs.
Work Schedules	Defines the work hours for one or more operators. Service Manager can generate a complex 24x7 schedule that spans multiple time zones, includes all shift and break information, accommodates any regional shift to Daylight Savings time, and automatically accounts for local or national holidays.

Add a master data record

Applies to User Roles:

System Administrator

Note: In ITIL best practices, a Configuration Administrator manages master data. In Service Manager, however, only a System Administrator has rights to master data, because the master data is shared by all Service Manager applications. You may need to contact a System Administrator to implement master data changes or to request permission to perform this function.

After you validate that the data sets for a new master data record request meet your company's specifications, you can add the master data record to Service Manager. For example, you may need to add a master data record for a new company location or a new vendor/supplier.

To add a master data record:

1. Click **System Administration** > **Base System Configuration**, and then double-click the master data category you want to add a record to.

   For example, if you want to add a master data record for a new company location, click **Locations**.

2. Specify the new master data information and preferences.

3. Click **Add**.

   Service Manager adds the new master data record.

Create a data reconciliation report summarizing the data modifications and any reconciliation errors in accordance with your company procedures.

Part of Workflow(s):

Manage master data (ST 3.6)

Applies to User Roles:

Configuration Administrator

Update a master data record

User roles: System Administrator, Configuration Administrator

Note: In ITIL best practices, a Configuration Administrator manages master data. In Service Manager, however, only a System Administrator has rights to master data, because the master data is shared by all Service Manager applications. You may need to contact a System Administrator to implement master data changes or to request permission to perform this function.

After you validate that the data sets for a master data record update request meet your company's specifications, you can update the master data record in Service Manager. For example, you may need to change the cost center for a department or change the status of a master data record to retired or obsolete.

To update a master data record:

1. Click **System Administration** > **Base System Configuration**, and double-click the master data category for the record you want to update. For example, if you need to update a master data record to change the conversion rate for an international currency, click **Conversion Rates**.
2. Fill in optional search criteria.
3. Click **Search**.
   Service Manager returns a list of records matching your search request.
4. Double-click the master data record you want to update.
5. Specify the new master data information. If you need to change the status of the record to indicate that it is retired or obsolete, update the record in accordance with your company procedures. For example, your company procedures might instruct you to add the words retired or obsolete to the record name.
6. Click **Save**.
   Service Manager updates the master data record.
7. Search for related active configuration items to ensure that the record updates do not conflict with the configuration administration.

Part of Workflow(s):

Manage master data (ST 3.6)

Applies to User Roles:

Configuration Administrator

Delete a master data record

User roles: System Administrator, Configuration Administrator

Note: In ITIL best practices, a Configuration Administrator manages master data. In Service Manager, however, only a System Administrator has rights to master data, because the master data is shared by all Service Manager applications. You may need to contact a System Administrator to implement master data changes or to request permission to perform this function.

After you validate that the data sets for a master data record delete request meet your company's specifications, you can delete a master data record in Service Manager. For example, you may need to delete a location record because the site was closed.

Warning: When you delete a master data record, it is possible that some configuration item records will still include the old master data value. Before you delete master data records, HP recommends you conduct a mass update to replace the master data value scheduled for deletion with the new approved value. This prevents obsolete data from remaining in Service Manager CI records.

To delete a master data record:

1. Click **System Administration** > **Base System Configuration**
2. Double-click the master data category for the record you want to delete.

   For example, if you need to delete a master data record for a user who is no longer with the company, click **Contacts**.
3. Fill in optional search criteria, and then click **Search**.

   Service Manager returns a list of records matching your search request.
4. Double-click the master data record you want to delete.
5. Click **Delete**.

   Note: When you delete the contact record, the corresponding operator record will also be deleted.
6. Click **Yes** to confirm the deletion.
   Service Manager deletes the master data record and notifies you that the corresponding operator record has also been deleted.

Database record auditing

Database record auditing allows you to check the HP Service Manager database for modifications to specified fields within a file. When you update records in a file, record auditing tracks the updates when Paging is not available, whereas Paging creates a unique record or page for every update.

Database Manager detects field modifications by comparing the input values in fields from the original version of a record to the values of the same fields in the updated version. When a modification is detected, Database Manager generates an Audit Log entry containing the following information:

- The name of the modified fields.
- The old and new version of the data.
- The date and time of the modification.
- The user ID of the operator who modified the record.

Invoke auditing

Applies to User Roles:

System Administrator

You can invoke auditing from Format Control. If you are using the Change Management application, you can invoke auditing from Database Manager.

Select one of the following options to invoke auditing:

- "Invoke auditing from Format Control" below
- "Invoke auditing from the file in Database Manager (Change Management only)" on the next page

Invoke auditing from Format Control

Applies to User Roles:

System Administrator

The topics below comprise an example of how to use Database record auditing. The steps in these topics must be followed in order. If you skip a step, Database record auditing may not work.

1. "Open the audit specifications table" on page 223.
2. "Add an audit specifications record" on page 223.
3. "Define an audit specifications entry" on page 224.
4. "Invoke auditing" above.

5. Set up event triggers.
6. Add lookup functionality to Format Control.
7. "Test audit lookup functionality" on page 218.

To invoke auditing from Format Control, perform the following steps:

1. Create an Audit Specifications record for a particular file.
2. Open the Format Control record associated with the form and the file for which you created the audit specifications record in step 1.
3. Select the **Save Copy** option.
4. Click **Subroutines** to display the Subroutines form.
   The Subroutines panel opens.
5. Enter the desired format control. For this example, enter:

Field	Enter
Add	`true`
Upd	`true`
Before	`true`
Application	`audit.compare`
Error Message	`Audit Processing could not complete.`
Names	`file` `second.file`
Values	`$file0` `$file`

Invoke auditing from the file in Database Manager (Change Management only)

Applies to User Roles:

System Administrator

This process is only valid for the Change Management application.

To invoke auditing from Database manager:

1. Open the **cm3rcatphase.main** form in Database Manager. A blank **cm3rcatphase.main** form opens.
2. Enter the name of any phase in the form. For this example, enter Analysis.
3. Click **Search**.
4. Set the **Audit Records** field value to true.
5. Click **Save**.

The topics below comprise an example of how to use Database record auditing. Follow the steps in this order. If you skip a step, the example will not work.

1. "Open the audit specifications table" on page 223
2. "Add an audit specifications record" on page 223
3. "Define an audit specifications entry" on page 224
4. "Invoke auditing" on page 215
5. Set up event triggers.
6. Add lookup functionality to Format Control
7. "Test audit lookup functionality" on the next page

Start database record auditing

Applies to User Roles:

System Administrator

This list is an example of the sequence of events that must occur prior to initiating database record auditing. Each of these steps is required for auditing to function properly.

To start database record auditing:

1. "Open the audit specifications table" on page 223.
2. "Add an audit specifications record" on page 223.

3. "Define an audit specifications entry" on page 224.
4. "Invoke auditing" on page 215
5. Set up event triggers.
6. Add lookup functionality to Format Control.
7. "Test audit lookup functionality" below

Test audit lookup functionality

Applies to User Roles:

System Administrator

Some files may not be configured for Audit Lookup. In that case, you must add Audit Lookup functionality before Audit Lookup will work.

HP Service Manager invokes Audit Processing using the definitions provided in the Format Control, when a record is added or updated in a file for which auditing has been set up.

Note: Before using Audit Lookup, be sure that the Audit Specifications include all fields that need auditing.

To test Audit Lookup functionality:

1. Add a record to the contacts file using the values in the table below.

Field	Value
Contact Name	MILLER, JOHN
Employee ID	NEW00003
Last Name	Miller
First Name	John
Company	advantage
Email	john.miller@Advantage.com
Dept Name	Documentation

Note: Service Manager displays the following message: *"Contact Information record added."* The audit process is invoked. Service Manager does not open an audit record until a user changes the current field values.

Service Manager has recorded all fields specified in the Audit Specifications record. All Old values are NULL or contain no data because there is no previous version of the record; this is a new record.

Note: Audit Log entries are created in response to any changes made to the current values in this record. These audit records are displayed in the **audit.summ** form, when accessed using the Format Control option created earlier.

If you check Audit Lookup now, a message informing you that there are no audit records.

2. To complete the example, modify the record created in the previous steps by entering new values in the fields of the form that were specified as Unique A-D fields in the audit specifications file. For this example, change the Employee ID to NEW00004.
3. Click **Save** to commit the changes to the database.

   Service Manager displays the following message: *"Record updated in the contacts file. Audit Record successfully recorded and added."* when the record is saved and an Audit Log is created.
4. Open the More Actions menu and select **Audit Lookup**.
   A record list opens.

   Note: The updates you see may vary, depending on how you have your triggers set up for the file.
5. Select the desired record from the list.

   Note: Only those fields which were modified are recorded. Service Manager displays both the Old and the New versions of each modified field in the audit log record. If none of the fields defined in the Audit Specifications Table have been modified, then Service Manager does not generate an audit log entry for that database dictionary file.
6. For further information regarding the modifications to the **contacts** file, click **Show detail**.

   The same record is re-displayed in an expanded form audit.g), which shows more detailed information for the arrayed fields, as well as scrollable fields.
7. Click **End** to return to the summary form.

8. From the summary form, click the **Prev** and **Next** buttons to review any additional audit records in the log for this contacts record.
9. Click **End** to return to the Contacts record.

The topics below comprise an example of how to use Database record auditing. Follow the steps in this order. If you skip a step, the example will not work.

To use Audit Lookup:

1. "Open the audit specifications table" on page 223.
2. "Add an audit specifications record" on page 223.
3. "Define an audit specifications entry" on page 224.
4. "Invoke auditing" on page 215.
5. Set up event triggers.
6. Add lookup functionality to Format Control.
7. "Test audit lookup functionality" on page 218

Audit processing

Maintenance of the Audit Specifications table used for audit processing follows normal HP Service Manager file maintenance procedures. For some tables, such as the contacts table, the out-of-box Format Control specifies that each time an update occurs, the audit.compare application is called. If you invoke auditing for a table for which Format Control was previously set up to invoke auditing, both trigger.invoke.auditor and Format Control will generate audit records, resulting in two audit records.

Note: If you are invoking Auditing from Format Control or a RAD call, ensure that you invoke Auditing *after* Validity Table processing but *before* the record add or update.

For additional information about regenerating database keys or resetting a database table, see the related topics.

Determine if a file has audit processing

Applies to User Roles:

System Administrator

Some files may not be configured for Audit Lookup. In that case, you must add Audit Lookup functionality before Audit Lookup will work.

To determine whether a file has audit processing:

1. Open the triggers file in Database Manager.
2. Click **Search**.

   A list of all files with triggers opens.
3. Search for the filename on the list.

Audit log file

The Audit Log File is the repository for data gathered during the audit. It displays the old and new values of input field data, as well as who made the revisions and when. The log is a record of all update transactions for the specified source file. The system adds one log record per source record update when you update one or more of the source record input fields defined in the Audit Specifications file.

Note: The following procedure displays a list of audit log records only if you have made modifications to contacts records after creating the audit specifications record. Otherwise, the log record is empty.

Open the audit log file

Applies to User Roles:

System Administrator

To open the audit log file:

1. Use one of the following methods to open the log file.
   - Type `audlog` from the command line and press **Enter** or click **Execute Command**.
   - Open the audit.summ form from Database Manager.
   - Click **Tailoring** > **Audit** > **Audit Log**.

     Service Manager displays a blank audit log form.
2. Click **Search**. If log records exist, Service Manager displays them.

Verify the field name

Applies to User Roles:

System Administrator

The system validates all field names defined in the record (Unique A through D and all Field Names) against the selected database dictionary file. When it finds an invalid field name, Service Manager displays the following message: *"The Field 'field name' is invalid. Select one from the list."*, and a pop-up window opens showing a list of valid field names for the specified database dictionary.

To correct a field name:

1. Double-click on the correct Field name in the fieldname window, and then click **End**.
   The system copies selected field name to the appropriate field.
   - or -
   Copy the correct field name, click **End**, and paste the field name into the field.

   Service Manager displays the following message: *"Record added to the auditspecs file"*, which means that all fields have been validated and that the record has been saved.
2. Re-query the record before attempting to make any modifications.
3. Exit the form.
4. Re-open the **auditspec** form.
5. Query for the contacts specifications record.

Verify the file name

Applies to User Roles:

System Administrator

When defining an audit specifications entry, if the File Name value is invalid, HP Service Manager displays the following message: "`The filename 'filename— is not valid. Select one from the list.`"

A record list opens and displays the valid database dictionary file names.

To correct or modify a file name, select an entry from the list, and then press **Enter**. The system copies the correct file name to the File Name field.

Add an audit specifications record

Applies to User Roles:

System Administrator

To add an audit specification record:

1. Click **Tailoring** > **Audit** > **Audit Specifications**.
2. Type a file name in the **Filename** field.

   In this example, type **contacts**.
3. Enter unique keys in the Unique A, B, C, or D field.

   The Unique key for the contacts file is contact.name. In this example, type **contact.name** in the Unique A field and leave Unique B through D as NULL.
4. Click **Add**.

When Service Manager records an Audit Log record, for example, the contact name Teresa Chan, the Filename field is recorded in the Audit Log as contacts and the Unique A field is recorded as TERESA CHAN.

The contacts Audit Log record then has a unique association with a contacts record. You can generate new Audit Logs based on the data found in the previous log record for a specific device.

Note: A *device* is also referred to as a *Configuration Item* (CI).

Open the audit specifications table

Applies to User Roles:

System Administrator

Use one of the following methods to open the audit specifications table:

- From the System Navigator, click **Tailoring** > **Audit** and then double-click **Audit Specifications**.
- From the Command line, type audspec from the command line and press **Enter** or click **Execute Command**.
- From the System Navigator, click **System Definition** > **Tables** > **auditspecs** > **forms** and then double-click **auditspecs**.

Define an audit specifications entry

Applies to User Roles:

System Administrator

The system validates Filename and Field Name values every time you update an existing record or add a new record. The Audit utility safeguard system prevents records with misspelled and incorrect file names or field names from being processed. Such errors potentially could cause faulty communication within the database.

Note: The name of the file and one field name of this example have been entered incorrectly to illustrate error-correction process built into the Audit utility.

To enter data in the Audit Specifications file:

1. Click **Tailoring** > **Audit** > **Audit Specifications**.

   A blank Audit Specifications Table form opens.

2. To select the file to create the specifications for.

   Enter information in the **Filename** field and **Unique A** through **Unique D** fields, as necessary, to parallel the Unique key in the source file with the Audit Log.

   Note: see the related topics and read the topic "Audit specifications file description" for more information on input fields.

   For this example, enter contacts in the **Filename** field, and then click **Search**.

   Caution: Making changes to the **contacts** file to cause it to invoke auditing will cause two audit records to be generated when a file is updated. To prevent this, use a backup, and restore the original file when finished with the example, or create a different file to practice on.

3. Define those fields you want Service Manager to audit and any aliases. If a field name is invalid, a list will open up allowing you to copy a valid name and use it to replace the invalid name.
   For this example, enter:

Field Name	Alias
contact.name	Contact Name
user.id	Employee ID
first.name	First Name
last.name	Last Name
dept.name	Department
email	Email Address
location	Location

4. Click **Add**. to retain this record and commit it to the database.

The topics below comprise an example of how to use Database record auditing. Follow the steps in this order. If you skip a step, the example will not work.

1. "Open the audit specifications table" on page 223.
2. "Add an audit specifications record" on page 223.
3. "Define an audit specifications entry" on the previous page
4. "Invoke auditing" on page 215.
5. Set up event triggers.
6. Add lookup functionality to Format Control.
7. "Test audit lookup functionality" on page 218.

Audit specifications file

The Audit Specifications (audspecs) file provides instructions on how and when to perform an audit. Use it to define files and fields that you want the Audit application to monitor. There is one specification record for each HP Service Manager database dictionary (dbdict) record.

Audit specifications file description

The Format Control record associated with the **auditspecs** form executes a routine that validates entries in the Filename and Field Name fields. Format Control executes this validation routine when you add or update Audit Specification records. It prevents users from entering invalid fields or file values and controls unpredictable run-time results.

Field	Definition
File Name	The name of a valid HP Service Manager database dictionary upon which Audit is to be performed. This is a required field on the table.
Unique A–D	Used to parallel data records in the Audit File to data records in the source file. Under most circumstances, the fields specified in Unique A through D are the same fields that are defined as Unique keys in the specified database dictionary file. These values need to be unique identifiers for the specific file. Unique A is the only required entry. Additional entries for Unique B through D are optional. Do not use Nulls & Duplicates keys to define Unique A through D, because if the fields in the key are NULL, then Service Manager creates Audit records that do not relate to any records in the Source File. You do not need to define the fields Unique A through D as keys in the Source File. However, you must ensure that any non-key fields are defined as unique identifiers, in order to avoid these errors. When defining Audit Specifications for the Incident and Change Management files, parallel the number field. It is not necessary to parallel the last or page fields. See database dictionary for information on identifying fields in files.
Field Name	Defines the name of the field to be checked for modifications. The fields specified for Auditing can be of any data type except arrayed structures or fields within arrayed structures. You can specify any number of fields; however, Auditing overhead increases as the number of fields increase. Do not specify all fields within a file for Auditing. Rather, analyze the fields within the file to determine which are critical for the management of data records in the file. The critical fields in the **contacts** file are: • contact.name • first.name • last.name

Field	Definition
	• dept.name • email • location The recommended maximum number of fields to audit is 20. Auditing more than 20 fields causes performance degradation. Under most circumstances, the recommended maximum will not impair the management of a file.
Alias	Defines the alias of a field name used in the audit log. When Service Manager records entries in the audit log (audit.summ.g form), the default is to record the actual field name. You can override the default by specifying an alias. For example, if the widgets file has a field named fd.ast.no, it may be more meaningful to define the alias field name fixed asset (configuration item) number. A one-to-one correlation exists between the Field Name and Alias input fields. Due to processing considerations, these are independent arrays. Therefore, you scroll one; you must also scroll the other to keep the definitions synchronized.

The Subroutines form

Consider each field in the Subroutines form in terms of the outcome you are seeking. The following fields are contained in the Subroutines form:

Field	Definition
Add	This field needs to be true, since you want audit data to be written whenever a new record is added.
Upd	(update) This field needs to be true to allow audit data to be written whenever there is a modification to an existing record.
Del	(delete) This field is empty, as that would write an audit record whenever a record is deleted.
Dis	(display) This field is empty.
Before	This setting on the subroutine call can be set to true or false. • If it is true, and the audit detects a key error, then an Audit record exists, but the update never occurs. • If it is false, then you receive the Record Updated message before the Audit Recorded message.
Application	This field states the application that compares the fields in the old and new data records, and if necessary, adds a record to the Audit Log file.

Field	Definition
Error Message	This field provides the error message, which is displayed if the process cannot be completed as planned.
Names	This field states which files are handled by the subroutine. Activate the Save Copy option on the Format Control Maintenance form to make this file variable available • Pass the Old (or original) version of the data record to the file parameter. • Pass the New (or updated) version of the data record to the second.file parameter.
Values	Values in this field correspond to each parameter. These values state to which variables Service Manager passes the data.

IR Expert

IR Expert is an intelligent, concept-based information retrieval engine that searches the HP Service Manager database for similar or related information based on a natural language query. For example, instead of relying on exactly matching keywords to find similar incidents in the Incident Management database, Service Manager uses the IR Expert engine and the description of an incident to locate similar incidents. IR Expert assigns a probability of relevance to the query results, and ranks them by relevance to the original query.

The Knowledge Base tool is a front-end to IR Expert. The Knowledge Base form enables users with specific search and operator restrictions to access solutions, specific databases or files, and other relevant data without compromising system security.

IR Expert has lexical analysis for Chinese and Korean.

For additional information on using IR Expert or Knowledge Management, see the related topics.

What is Knowledge Engineering?

Knowledge Engineering describes the process to develop and enhance the Knowledge Base used by IR Expert. The application provides several knowledge tools to users, including a centralized knowledge database and solution candidates.

The centralized Knowledge Base unifies data in other knowledge bases, such as KnowledgePak, and probcause files, by replicating it into a central knowledge file, called *core*. You can add new knowledge directly to the central Knowledge Base, or add it to the individual knowledge bases. As you add records to individual databases, HP Service Manager automatically replicates their contents into the core knowledge file.

When you close incidents or service desk interaction records, you can mark the records as solution candidates. For information on marking an incident as a solution candidate, see the related topics. Relevant information from these solution candidates is extracted into a candidate file, called *protocore*. Knowledge engineers review the data submitted as candidates. They can discard the data, or modify it and promote an incident and solution into the core knowledge file.

Standard record lists and IR Expert

When you create a query, HP Service Manager checks the target file for IR keys. If you specify data that is part of an IR key, then Service Manager runs an IR query before any other query. By default, IR Expert returns a list of records for each search. The number of records selected is a function of the granularity of the IR parameters and the presence or absence of additional, non-IR query elements.

IR queries can be combined with other queries to help narrow the search for information. For example, in Incident Management you may want to find out about a particular application error code, but only for incidents in the sysops assignment group. If you type sysops in the assignment field, and the application fails with 00023.EXE, and you describe this in the incident description, IR Expert searches for incidents that are like the incident description first. For a match to occur, the description contains words related to the stems of each major component of the example message. Next, Service Manager eliminates any incidents found by the IR search if the assignment field does not begin with sysops.

You can also use an IR search in conjunction with the Find option by defining a link from a field normally used in an IR query in the source file to any field that is part of an IR key in the target file. For example, you can define a link from the `Action.action` field of the `problem` file to the `description.structure.description` field of the `cm3r` file to detect similar change requests when opening or updating an Incident.

Database dictionary and IR Expert

Installing IR Expert creates three changes in the Database Dictionary utility:

- The IR key types are available in the key list.
- The regenerate type IR Regen is available when keys in a file that contains an IR key has been modified.
- The Type field in the datadict.g record reflects the presence of an IR key in a file.

Note: An IR Regen does not necessarily rebuild any indices other than IR indices. If you modify any key other than an IR key, HP Service Manager may not recognize the change until you perform a full regen. The average IR Regen time is 1 minute for each 2,500 records.

Data Policy information (datadict file) is a critical factor in the IR Query application. It stores details on which files contain IR keys and which fields within those files are keyed for IR queries. You can access the datadict file by clicking **Tailoring** > **Data Policy**.

IR Query features

IR Expert offers these advantages when you design queries:

- You can reuse the same query against multiple files.
- You can tune, modify, and resubmit a query.
- You can copy information from a target record into the source field.
- You can use conditional logic to build other files (or IR-specific files) when you update conventional files.

Using IR Expert to create a query

When you access IR Expert, it displays one of the following:

- A search form containing a query if you invoked it from an Incident Management or Service Desk application.
- A form with a list of files to search.

IR Expert requires a specific query to process against the target file that is a plain text query. It returns a record list containing related records sorted by relevance.

If you issue an IR query by opening the More Actions menu and clicking **Search Knowledge** (or clicking **Find Solution** if you have not purchased a Knowledge Management license) from an incident or service desk interaction record, you can record and post information from the IR record back to your source incident or service desk interaction record.

Note: You cannot submit a true query, which is an empty query statement, to retrieve all records in the specified file.

Searching the central Knowledge Base

The central Knowledge Base is the default database for any knowledge search in HP Service Manager. You can search it using the standard Service Manager Knowledge application, or you can choose other Knowledge Bases. However, if you choose the global Knowledge Base, you are searching all knowledge resources at once.

Promoting or discarding a solution candidate

As you close incident records and mark them as solution candidates, the candidate (protocore) file fills. (For information on marking an incident as a solution candidate, see the related topics.) Knowledge engineers examine proposed solutions and promote them to the central Knowledge Base (core) if applicable, or delete them if they are not useful.

If the Autopost Solution Candidates? (auto.post) field is checked (true) in the Incident Management Environment record, IR Expert automatically promotes recommended solution candidates to the core file. The auto.post check box must be cleared (false) for the solution candidates to remain in the candidate protocore file for approval.

IR Expert tasks

IR Expert tasks involve:

- Accessing IR Expert from other applications.
- Managing queries.
- Creating solution candidates.
- Searching the Knowledge Base.

Access IR Expert

Applies to User Roles:

System Administrator

You can access the IR Expert application from many locations within HP Service Manager.

Note: If you have not purchased a Knowledge Management license, the "Find Solution" option is available. This option helps you to find some solutions in a smaller knowledge base. If you have a Knowledge Management license, the "Search Knowledge" option is available instead of "Find Solution".

The following table contains examples.

Application or Tool	To Access IR Expert:
Service Desk	Click **Service Desk** > **Search Knowledge Base**.
	Click **Service Desk** > **Interaction Queue** > **Search > Text Search**.
	1. Click **Service Desk** > **Search Interaction Records** > **Search**. 2. Double-click any interaction record to display it in the service desk interaction form. 3. Open the More Actions menu, and click **Search Knowledge** or **Find Solution**.
	1. Click **Service Desk** > **Search Interaction Records**. 2. Click the **Text Search** tab. 3. Complete the Search Text text field. 4. Click **Search**. The IR query runs automatically against the ir.incidents file first. IR Expert returns a record list of matches to the query.
Incident Management	Click **Incident Management** > **Search Knowledgebase**.
	Click **Incident Management** > **Incident Queue** > **Search > Text Search**.
	1. Click **Incident Management** > **Search Incidents** > **Search**. 2. Double-click any incident to display it in the Incident form. 3. Click **More** or the More Actions icon, and select **Search Knowledge** or **Find Solution**.
	1. Click **Incident Management** > **Search Incidents**. 2. Click the **Text Search** tab. 3. Complete the Search Text field. 4. Click **Search**. The IR query runs automatically against the ir.probsummary file first. IR Expert returns a record list of matches to the query.
Database	1. In the Table field, type **kmquery**.

Application or Tool	To Access IR Expert:
Manager	2. Click **Search**.
Tailoring	Click **Tailoring** > **View Knowledge Base**.

Enable IR search for a file

Applies to User Roles:

System Administrator

To enable IR search for a file (table), you must create an IR key for the file, and then regenerate the IR index for the file.

You can define an IR key using the Database Dictionary utility. As an example, the following procedure describes how to enable IR search for the `device` file.

1. Log in to the Service Manager Windows client.

   Note: The web client does not allow you to add a new field/key for a Database Dictionary record.

2. Click **Tailoring** > **Database Dictionary**.

3. In the File Name field, type **device**.

4. Click **Keys**, place the cursor in the first blank key line, and click **Edit Field/Key**.

5. In the key window, do the following:
   a. In the Type field, select **IR key**.

   b. In the Fields section, type two fields: **comments**, and **title**. These are the names of the fields that you want to use as IR key fields.

      Note: For the rules that you must follow when selecting IR key fields, see "Building IR keys" on page 246.

   c. Click **OK**.

      Service Manager creates the IR key for the device file.

6. Click **OK** to save the file.

Service Manager displays a message, stating that you are about to Regen the file and asking if you want to continue.

7. Click **OK** to run the IR Regen now.

   Note: You can also click the Schedule button to run the IR Regen at a later time.

8. Wait for the IR Regen to complete.

   A message displays, stating that the IR Regen has completed.

9. (Optional) Verify that IR search is now available for the device file.
   a. Click **Configuration Management** > **Resources** > **Search CIs**.

   b. Do a true search, and locate two CIs whose Title or Description field is not empty. Make note of two words that display in any of these fields in the two CIs, one word for each CI.

   c. Click **Cancel**.

   d. Click **Text Search**.

   e. In the Search Text field, type the above two words, and select **Any of these words**.

   f. Click **Search**.

      The two CIs should appear in the returned search results.

Regenerate IR keys

Applies to User Roles:

System Administrator

You need to run an IR Regen only in the following cases:

- If you are changing an IR key (adding or removing a field);
- If you are changing global IR configurations, for example, the stop words file, the techterms file, and language definition files;
- If you have found an IR index corruption.

When needed, perform an IR Regen against a specific file or against all IR indexed files in the system.

Note: You can perform an IR Regen by using the System Definition or the Database Dictionary utility.

To regenerate the IR index for a table using System Definition:

1. Log in to the Service Manager Windows client.
2. Click **System Definition** > **Tables**.
3. Double-click the name of the table containing the IR key you want to regenerate.
4. From the **Table management** section, click **Regenerate IR index**.
   Note: This option is only available for tables that contain an IR key.
5. Specify the regeneration parameters, and then click **OK**.

To regenerate the IR index for a table using Database Dictionary:

1. Log in to the Service Manager Windows or web client.
2. Click **Tailoring** > **Database Dictionary**.
3. In the Table field, type the table name (for example, **incidents**), and click **Search**.

   The table details form opens.
4. From the table details form, click **More** or the More Actions icon, and select **Regen IR**.
5. Click **OK** to run the IR Regen immediately or click the Schedule button to run it at a later time.

Note: When you run an IR Regen, Service Manager needs to clear the current indexes. This causes a delete to the back-end database. When Service Manager attempts to delete a large set of records, the database could run out of space for this activity and cause an error. Each supported database platform provides some type of transaction or undo log to back up data changes and allow a rollback in case of errors. If you encounter such an error, contact your database administrator and ask them to increase the size available for this purpose.

Some examples of the errors you could encounter are:

Oracle: SQL code=30036 message=ORA-30036: unable to extend segment by % in undo tablespace

SQL Server SQL State: 42000-9002 Message: [Microsoft][SQL Native Client][SQL Server]The transaction log for database '%' is full

DB2 SQL State: 57011--964 Message: [IBM][CLI Driver][DB2/NT] SQL0964C The transaction log for the database is full. SQLSTATE=57011

Start IR Asynchronous mode

Applies to User Roles:

System Administrator

Asynchronous IR means that instead of every session directly updating IR indexes as data is updated, the IR index changes are logged to an internal table (irqueue), and a background process performs all the IR index updates separately.

To start asynchronous IR:

1. Stop all HP Service Manager servers running a horizontal scaling or classic load manager implementation.
2. Open the Service Manager initialization file (sm.ini) on each system and add the following line.
   `ir_asynchronous:1`
   Note: You must add this parameter to each system running in a horizontally scaled implementation so that each server knows that the IR process is running asynchronously.
3. Save the initialization file.
4. Open the Service Manager configuration file (sm.cfg) on the system running the load balancer process and add the following line.
   `sm -que:ir`
5. Save the configuration file.
6. Restart the Service Manager servers.

Load data files with IR Expert keys

Applies to User Roles:

System Administrator

IR keys are created when loading data into HP Service Manager, such as when moving files from a test system to a production system. Any files containing one or more IR keys will require an IR Regen of the new system.

If the file already exists in the new system and the file contains at least one record, you can add the IR key and perform the IR Regen before loading the records from your old system. Then Service Manager updates the IR indices as it adds each record during the load.

Copy information from a query to a source record

Applies to User Roles:

System Administrator

To copy information from a query to a source record:

1. From an incident or service desk interaction record, click **More** or the More Actions icon, and then click **Search Knowledge** or **Find Solution**.

   Note: The Find Solution option is available instead of Search Knowledge, if you have not purchased a Knowledge Management license.

2. Double-click a record from the list returned by the IR query.

3. Click **Use Solution**.

   Note: The Use Solution option is available only when the record's Solution field is not empty.

   HP Service Manager returns to the source record. This action copies the Solution information from the selected record to the source record. The copied information replaces the original information in the source file.

Create an IR Expert query

Applies to User Roles:

System Administrator

To create an IR Expert query:

1. Click **Tailoring** > **View Knowledge Base**.

2. Type specific data in the fields to gather the information you want. For example, specify similar solutions used in previous incident records of the same category, for the same configuration item or related piece of equipment, or new solution candidates.

3. Click **Search**.

4. After you submit the query, HP Service Manager generates a record list and displays the first record in the list. Each of the records in the record list represents a match in the IR Expert search.

5. To view a record, double-click the record in the record list.

Enable editing of Knowledge learning records

Applies to User Roles:

System Administrator

To enable editing of Knowledge learning records:

1. Click **System Administration** > **Base System Configuration** > **Miscellaneous** > **System Information Record**.
2. Click the **General** tab.
3. Select the **Allow Editing of Learning Record** option.
4. Click **Save**.
   HP Service Manager displays the message: Information record updated.

Enable learning for Knowledge records

Applies to User Roles:

System Administrator

Enabling learning for Knowledge records grants users access to the **Use Resolution** option when adding a solution to a knowledge record.

To enable learning for Knowledge records:

1. Click **System Administration** > **Base System Configuration** > **Miscellaneous** > **System Information Record**. The info.company.g form opens.
2. Click the **General** tab.
3. Select the **Use learning for Knowledge Engineering** option.
4. Click **Save**.
   HP Service Manager displays the message: Information record updated.

Mark an incident record as a solution candidate

Applies to User Roles:

System Administrator

When you recommend a solution, you are suggesting that the solution is a good candidate for the Knowledge Base. This option is available only when you close an incident record.

To mark an incident record as a solution candidate:

1. Click **Incident Management** > **Search Incidents**.
2. Type the incident number and then click **Search**.
   You can also click **Search** to generate a record list of existing incidents. Double-click a record to display it in the Incident form.
3. Click **Close Incident**.
   The status changes to Closed.
4. In the Closure Code field, click Fill to select a code.
5. In the Solution field, type a description of the solution to the incident record.
6. If the resolution should be saved to the HP Service Manager knowledge base, select the **Knowledge Candidate** checkbox.
7. Click **Save**.

The system automatically replicates relevant data from the incident record into the Knowledge Base candidate (protocore) file.

Modify an IR Expert query

Applies to User Roles:

System Administrator

To modify an IR Expert query:

1. Create and run an IR Expert query.
2. Open theMore Actions menu.
3. Click **Modify Query**.
4. IR Expert displays the query again. Change the search criteria or change the IR file (Knowledge Base) to be searched.
5. Press Enter to run the new query, or run the original query against the new file.

Promote or delete records in the protocore

Applies to User Roles:

System Administrator

In this example, we create a Problem Management solutions candidate and promote it to the central Knowledge Base (core table).

1. Log in to the Service Manager Windows client.
2. Open a new problem and add a solution.
   a. From the System Navigator, click **Problem Management** > **Problem Control** > **Create New Problem**.
   b. Open a new problem, input the desired text in the problem description field, complete other required fields, and then click **Save**.
   c. Click **Next Phase**.
   d. Provide a Closure Code, a Root Cause Description, and Suggested Workaround.
   e. Open the **Activities** section and update the New Update Type and New Update fields.
   f. Click **Save** to save the Problem, and then click **Close Problem** to close it.
   g. Click More or the More Actions icon, and select **Post to Knowledge**.
   The server displays a message stating that the problem record has been successfully posted as a solution candidate.
3. Click **Tailoring** > **Knowledge Engineering** > **Solution Candidates**.
   IR Expert launches the Solution Candidate Editor.
4. Search and open the solution candidate posted from the problem record. For example, on the Source Key/Info tab, you can type the problem record number (such as PM10001), and then click Search.
5. To move the record into the central Knowledge Base, click **Promote**.
   The server removes the promoted knowledge item from the list.
6. Click **OK** to exit.
7. From the System Navigator, click **System Definitions** > **Tables** > **Core**.

8. In the Table management section, click **Regenerate IR index**.
9. Select **Now** to re-index the file immediately, or select **Later** to do so at a later time.
10. Wait until the IR Regen is complete, and click **View all records in the table**
11. Scroll down to the bottom part of the list, and double-click the core table record to see the newly-promoted knowledge item.

Search the central Knowledge Base

Applies to User Roles:

System Administrator

To search the central Knowledge Base:

1. Click one of the following:
   - **Incident Management** > **Search Knowledgebase**.
   - **Problem Management** > **Problem Control** > **Search Knowledge Base**.
   - **Service Desk** > **Search Knowledge Base.**
2. In the **Search In** section, make sure that all the knowledge base sources are selected.

   Note: All the knowledge base sources are selected by default.
3. Type your search text in the **Search Knowledgebase** text box.
4. (Optional) Click **Advanced**, and enter your search criteria in the **Filter by...** section.
   - All of these words: Search for documents that contain every one of these words.
   - This exact phrase: Search for documents that contain this phrase exactly. These words are not expanded with a thesaurus.
   - Any of these words: Search for documents that contain any of these words.
   - None of these words: Narrow down the search to include only documents that do not contain any of these words.

5. (Optional) Narrow the search by specifying one or more fields in the Knowledge Library, Known Errors, Problems, Incidents, and Interactions sections.
6. Click **Search**.

   Service Manager displays the search results.

Special considerations for using IR Expert

If you are an IR Expert administrator, it is important to know that IR Expert relies heavily upon shared memory.

In HP Service Manager, all IR index data is in the scirexpert table. If necessary, Service Manager creates scirexpert automatically the first time you run an IR Regen.

How IR Expert evaluates documents for relevance

IR Expert queries return results based on relevance to the query. To do this, IR Expert looks at each term used in an IR query and gives a ranking to the term, based on how often it appears in the stored documents.

A term found in many documents has a lower rank than a term found in a few documents. For example, if all of the incidents that customers report involve the Windows operating system, then the term “Window” is in almost every document and has a very small ranking.

After IR Expert assigns ranking, it gives each stored document a weight based on how many of the terms used in the query are in the document, and on how often a term is in the document. A document that contains a term twice has a greater weight than a document that contains the term once. Next, it compares the terms in the document with the terms in the query to see if there is a “phrase” match. If so, IR Expert gives that document a higher weight. Finally, IR Expert considers the most recently updated document to be the most relevant.

Operations on terms

IR Expert enforces term operations on all queries run on the HP Service Manager database, enabling automatic indexing based on key terms. These include:

Lexical analysis
The process of converting an input stream of characters into a stream of words or tokens.

Stemming

The process of combining related words, based on the reduction of words to their common root, stem, or form.

Stop words

A list of words determined to have no indexing value, used to eliminate potential terms appearing in query statements. If a word from the stop list appears in a query, IR Expert ignores that word in the search.

Spelling correction

The automated process of verifying and correcting the spelling of query terms.

Lexical analysis

Symbol	Description
Digits	Numbers do not make good index terms, and are not usually included as tokens. In some instances, query statements consist of only digits, such as a record number. Therefore, IR Expert indexes digits along with alphabetic characters.
Hyphens	Words broken at the end of lines, or including hyphens, can result in multiple word fragment tokens. IR Expert considers hyphenated terms as a single token and does not break them apart.
Other punctuation	Other punctuation, including periods (.), commas (,), and underscores (_) are often used as parts of terms. IR Expert allows apostrophes (’), dashes (-), and periods (.) to appear within, but not at the beginning or end of a token.
Case	Case distinctions are important in some cases, such as programming languages. IR Expert is case-insensitive. It converts all HP Service Manager database terms to lower case.

Stemming

Stemming enables the user to find the variants of a term, while reducing the size of the index file. Because single stems typically correspond to several full terms, storing stems instead of full terms enables a compression factor of over 50 percent.

Stop words

The stop words file naming convention in IR Expert eliminates the need for an extensive list of English words. Both the most frequently occurring and least useful words for intelligent data retrieval are identified.

Spelling correction

The primary issue with spelling correction is to identify when an input string is significantly close to one of a set of given strings. When a user enters a query, after lexical analysis and the other conditions are performed, IR Expert attempts to identify words in the index that are close to the unrecognized word. The requirement is to attain good selectivity while still exploring large databases in a timely manner. To do this, IR Expert uses two tests.

First, IR Expert looks for a shallow match. It compares all existing index terms to any unidentified query terms, taking into account order of letters. Finding shallow matches relies upon the identification of same letters in both words, ideally seeking a distance of zero (0) between the words/letters. IR Expert passes words with a distance of two (2) or less to the second test.

Secondly, IR Expert does a deep match test to prune obviously different words with merely similar letter arrangement, for example, bushland for husband. Deep matching verifies letter order within words. IR Expert uses the words with the lowest distance as corrections for unrecognized words.

Using IR Expert and Adaptive Learning

IR expert allows users to take search phrases and relate them to solution phrases in a table so that when the same query is used, the usage relevance score for the solution phrase increases. Additionally, IR Expert allows administrators to manually change the relevance score of individual phrases and words.

The accuracy of this feature is based on a query term and how it may have been used, or incorrectly used. The data is useful only if the adaptive learning data is monitored by a System Administrator.

The largest value that can be applied to a query or incident record is 65000.

IR Expert scirexpert file

HP Service Manager stores all IR index data in a single database table named scirexpert to ensure database independence. Service Manager creates the scirexpert file in the RDBMS.

When you run IR Regen, Service Manager checks the scirexpert file for an index. If the index is missing, it uses a flat file. After regenerating all IR indexes, all IR data resides in the scirexpert file. No flat files (ir.*) exist. When you back up the database, you will back up the IR index (scirexpert) file.

Other IR procedures, such as regenerating IR indices and creating IR searches, remain the same.

Creating an IR file

An IR file contains information that duplicates data in other HP Service Manager tables, such as `problem`, `cm3r` and `cm3t`. Service Manager uses IR files for information retrieval.

You create the IR file as you would create any other Service Manager table. It must contain an IR key in addition to the key used in the link query. The field names do not have to be identical, but a number field must be the unique key in the IR files accessed by the `probsummary`, `cm3t`, and `cm3r` files.

There is a sample `ir.probsummary` table in the out-of-box Incident Management application. This table, and all `ir.<filename>` tables, are in the Service Manager`Data` directory.

Note: Service Manager creates these files automatically when you perform an IR Regen on database dictionary tables that contain IR keys, and when users add or update records in tables containing IR keys.

Updates to IR files

You can update IR files in two modes:

Synchronous mode

The default mode. The system immediately writes All IR Expert updates into the IR Expert files. The IRQUEUE processor is not used.

Asynchronous mode

The system places changes to the IR key in a queue. They are not processed immediately. A separate background process (IRQUEUE), writes the accumulated updates into the IR index files. The changes are then available for searches. Query response time is faster when using Asynchronous IR. In asynchronous mode, updates to files that have an IR key do not have to wait for the completion of IR queries that are executing at the same time.

Note: If the IRQUEUE process is not running for any reason, changes to IR index data are not available to users. IR files do not reflect the newest IR index data and therefore will not retrieve newly added data. However, searches will work against existing IR data.

IR keys and non-IR keys

An HP Service Manager file can contain only one IR key, and must contain a non-IR key. All files that contain an IR key must also contain another key that is Unique, with No Nulls, No Duplicates, or Nulls &

Duplicates. An attempt to build an index for a file that contains only an IR key results in a regen error. The system builds the external IR indices, but removes the IR key from the Service Manager index.

Building IR keys

Applies to User Roles:

System Administrator

Define IR keys using the standard Database Dictionary utility in HP Service Manager. IR keys are composed of one or more array or scalar text fields. IR keys that combine array and scalar fields should define an array field as the first element of the key.

Use only text fields in an IR key. A text field is an array or scalar field that contains arbitrary information that not used for traditional queries. For example, the device file contains a scalar field called description. This field contains descriptive text about a device, such as:

- Does not support the new drawing package.
- This modem needs ventilation on the top and sides.

The assignment file contains an array field called operators. The operators array contains the exact login name of Service Manager operators, and is not a good candidate for an IR key since IR searches are relevance searches.

Follow these rules when defining IR keys:

- If a query contains a field that is part of an IR key, Service Manager always performs an IR search. IR Expert selects and presents records based upon relevance.
- If a query contains a field that is part of an IR key, your sort criteria are honored. IR queries always sort by relevance.

You can combine IR queries with traditional queries to specifically limit the answer set.

Multiple files containing IR keys

The current release of HP Service Manager uses shared memory, and does not specify a practical limit on the number of files that can contain IR keys. The system caches the most frequently referenced data, making the number of files that have IR keys less of an issue.

Using multiple fields within the IR keys causes no real impact to IR performance. Service Manager takes all fields defined as part of the IR key and concatenates them for IR processing. It takes more time to concatenate five fields than two fields, but the difference is negligible. Multiple files with IR keys

compete for use of the shared memory cache. You should allocate more shared memory as you increase the number of files with IR keys.

Consider using the Knowledge Base that enables you to store all corporate knowledge data into a single file (core) for IR processing.

Find Solution

Find Solution enables you to find information in the Knowledge Base. Use Find Solution to input, retrieve, and flag information for specific importance. You can use fields that are predefined in the link file when you create a Find Solution query.

Edit queries for Find Solution

Applies to User Roles:

System Administrator

Find solution enables you to find information in the Knowledge Base. You can use fields that are predefined in the link file when you create a Find Solution query. For information on understanding links, including how to add or modify a link, see the related topics.

To edit queries for Find Solution:

1. Click **Tailoring** > **Database Manager**.
2. Type **link** in the Form field, and then click **Search**.

   HP Service Manager displays a list of all forms named link.
3. Double-click the form **link**.
4. In the Name field, type **kn**, and then click **Search**.

   A list of all link records that begin with kn opens.
5. Select a link record.
6. Open the More Actions menu and choose **Select Line**.
7. Edit any field to change the query. For example:

Field	Data
Field (From/Source)	Unique key in core file.
File (To/Target)	Name of the table Service Manager should fill from or to.
Field (To/Target)	Unique key in change file.

8. **Save** the query.
9. Click **Back** to exit.

Customizing IR Expert

IR Expert already supports English, German, Chinese, Japanese, and Korean languages without customization. However, you can configure IR Expert to perform efficiently in Spanish, French, Italian, and Portuguese or any other language with the use of stop words, stem, and suffix dictionaries by using these files:

- The stop word file: *[ir_languagefiles_path]<language>*.stp
- The stem dictionary: *[ir_languagefiles_path]language*.STM
- The suffix file: *[ir_languagefiles_path]language*.SUF

Note: IR search is not supported for the Arabic and Hebrew languages.

For Chinese, Japanese, or Korean, see "Chinese, Japanese, and Korean language analyzer" on page 250.

Implement IR searches

Applies to User Roles:

System Administrator

In this example we use <language>. Substitute the language you need to implement wherever you see “<language>”.

To enter language files into the IR system:

1. For English, Spanish, French, Italian, Portuguese, and Japanese, create the files:
   - <language>.stp (stop words)
   - <language>.stm (stem dictionary)
   - <language>.suf (suffix dictionary)
   - <language>.nor (normals dictionary)

   For Chinese use the following files:
   - irlang/cma_options.utf8
   - irlang/cma/*

   For Japanese, use the following files:
   - irlang/jma_options.utf8
   - irlang/jma/*

   For Korean, use the following files:
   - irlang/kma_options.utf8
   - irlang/kma/*

   Note: These file names may vary for different languages or platforms.
2. Place those files in a unique directory. In this example, we use the directory DICT_PATH.
3. Insert the following parameters into the sm.ini file.

   `ir_language:<language>`

   `ir_languagefiles_path:..\irlang\>`

   `ir_opt_path:..\irlang\cma_options.utf8` for Chinese

   or

   `ir_opt_path:..\irlang\jma_options.utf8` for Japanese

   or

`ir_opt_path:..\irlang\kma_options.utf8` for Korean

Chinese, Japanese, and Korean language analyzer

The language analyzer is a text tokenization analysis tool. It uses statistical-based, rule-based, and dictionary-based techniques to quickly determine the correct segmentation of Chinese, Japanese, or Korean text.

In addition to tokenization, the language analyzer offers normalization, part-of-speech information, morphological analysis, and reading information.

Searching for Chinese or Japanese and English words in a Chinese or Japanese environment

For Chinese

- English stemming and branching are not supported.
- A mixture of Chinese and English characters is not supported when the characters are separated by a blank space.
- A mixture of Chinese and English characters is supported when the characters are not separated by a blank space.
- Multiple Chinese keyword search is not supported.

For Japanese

- English stemming and branching are not supported.
- A mixture of Japanese and English characters is not supported when the characters are separated by a blank space.
- A mixture of Japanese and English characters is supported when the characters are not separated by a blank space.
- Multiple Japanese keyword search is not supported.

Searching for Chinese or Japanese and English words in an English environment

For Chinese

- English stemming and branching are supported.
- A mixture of Chinese and English characters is supported when the words are separated by a blank space.
- Multiple Chinese keyword search is supported when the words are separated by a blank space.
- Chinese stop words are not supported.
- Chinese stemming is supported.

For Japanese

- English stemming and branching are supported.
- A mixture of Japanese and English characters is supported when the words are separated by a blank space.
- Multiple Japanese keyword search is supported when the words are separated by a blank space.
- Japanese stop words are not supported.
- Japanese stemming is supported.

IR Expert file descriptions

The following table summarizes information about the IR Expert files.

Description	Purpose	Naming Convention
stop words	Required user file maintained by Service Manager administrator. Contains words that have little or no value to the information retrieval process. For example, prepositions are stop words. You can add or delete stop words as necessary. Changes take effect when you restart HP Service Manager and regenerate the indexes.	*[ir_ languagefiles_ path] language*.stp where *ir_ languagefiles_*

Description	Purpose	Naming Convention
		path and *language* correspond to start-up parameters.
stem dictionary	Required system file for languages other than English and German. Contains word stems from which derivative words are formed, allowing IR Expert to match closely related words. Maintained exclusively by IR Expert.	*[ir_ languagefiles_ path] language*.stm where *ir_ languagefiles_ path* and *language* correspond to start-up parameters.
suffix dictionary	Required system file for languages other than English and German. Contains suffix templates used in stemming. Maintained exclusively by IR Expert.	*[ir_ languagefiles_ path] language*.suf where *ir_ languagefiles_ path* and *language* correspond to start-up parameters.
normals dictionary	Required if the language employs special keyboard characters. You can add or delete Normalization characters as necessary. Changes take effect when you restart Service Manager and regenerate the indexes. The excerpt below shows a typical normalization file. The first two characters of each line become substitutions for the following character or comma-separated characters (in decimal notation). ae 132,142 oe 148,153 ss 225 ue 129,154	*[ir_ languagefiles_ path] language*.nor where *ir_ languagefiles_ path* and *language* correspond to start-up parameters.

Normals dictionary

The normals dictionary, [ir_languagefiles_path]language.NOR, is involved only when there are characters in the language that IR Expert transforms into other characters. For example, in the German language IR Expert changes the umlauted characters. So it changes “ä” into “ae”. You may want to do this to make setting up the stem (.STM) and suffix (.SUF) dictionary files easier.

Stem dictionary

The stem dictionary, [ir_languagefiles_path]language.STM, contains the stem, which is the part of a term used in the IR indices. IR Expert considers each word to have a stem (defined in the .STM file) and many possible suffixes (defined in the .SUF file). For example: For the words go and going, “go” is the stem and “ing” is the suffix. Entries in the .STM file consist of the stem word (go) followed by a blank, and then an index entry in the suffix file (.SUF) would be “go 1“. This index indicates which suffix values are acceptable for the stem word.

Stemming example

In this example, a user wants stemming only for the words take, ride and walk.

- The acceptable forms of take are take, taken, taking.
- The acceptable forms of walk are walk, walking, walked.
- The acceptable forms of ride are ride, ridden, riding.

The stem dictionary (.STM file) might contain the following setup:

- tak 1 (words with this stem will use the first suffix option)
- rid 1
- walk 2 (words with this stem will use the second suffix option)

The suffix dictionary (.SUF file) would contain:

- e, en, ing, den
- ing, ed

Based on the stem and suffix dictionaries:

Take, taken, taking would result in tak.

Walk, walking, walked would result in walk.
Ride, ridden, riding would result in rid.

These files are not perfect. For example, IR Expert would change taken to tak because suffix index tak and rid are the same.

You could change the configuration so that the stem dictionary (.STM file) contained:

- tak 1
- rid 3 (words with this stem will use the third suffix option)
- walk 2

and the suffix dictionary (.SUF file) contained:

- e, en, ing
- ing, ed
- e, den, ing

Note: Setting up these language support files requires a considerable amount of time. It should only be undertaken by someone fluent in the language, and knowledgeable of word components and pronunciation.

Stop word file

The stop word file, *[ir_languagefiles_path]<language>*.stp, contains a list of words used too frequently to index because that would hinder the identification of documents. Edit this list as you would a text file. Type each word on a separate line.

The words in the file go through a stemming process, which eliminates the need to specify all the forms for the word. For example, in English, you do not have to enter both "go" and "going" into the stop words file because the stemming algorithm changes "going" to "go." The only word that must be entered in the .STP file is the word "go."

Suffix file

The suffix file, [ir_languagefiles_path]language.SUF, contains a series of lines, each a list of valid suffix values. The .STM file indicates which line in the .SUF file should be used as the possible suffixes for any given stem word. For example, the stem go suffixes might be “ing”, “es”, or “ne.”

Record retrieval

HP Service Manager allows you to search for and retrieve database records based on particular user-provided search criteria (query). Searching is case-sensitive by default, unless your database has been set up for case-insensitive searching. For information on how to improve query speed, see the Help topic Tuning: Improving query speed.

There are three methods of querying records from a Service Manager database:

- Simple queries are defined using the record list method.
- More complex queries are defined using the query window method.
- Other queries can be performed using IR Expert information retrieval.

The Database Manager utility

The HP Service Manager Database Manager utility runs in two modes, standard and administrative.

- In standard mode, behavior is determined by whatever security you have in place. Request Management, for example, uses Request security while the standard database uses Format Control. In standard mode, an administrator does not necessarily see all options. Options that are potentially troublesome for Service Manager, mass-updates, regenerating keys, and so on, are kept out of sight to prevent accidental use.
- Administration Mode is similar to root privileges on Unix systems. Administration Mode is powerful, in that you can make changes that affect Service Manager as a whole. In Administration Mode, (with the Administration Mode check box selected), a System Administrator will have rights to ALL options unless a different displayscreen is used based on the value of the Administration Mode check box. The value of the Administration Mode check box affects which displayscreen is used for certain files. The link file will use the link.view displayscreen when Admin Mode is disabled and db.view when Admin Mode is enabled. The globallists file will use the global.view displayscreen when Admin Mode is disabled and db.view when Admin Mode is enabled. When Admin Mode is enabled for the link and globallists files, the SysAdmin no longer has access to the Select Line and Rebuild Global List options respectively.

Note: Do day-to-day administration in standard mode.

Two capability words control whether a user administrator can use standard or administrative mode.

- Users with the *SysAdmin* capability word defined in their operator file will see the Administration Mode checkbox.
- Users with the *AlwaysAdmin* capability word defined in their operator file will be in Administration Mode by default.

Determining System Navigator menu options

The function and contents of the items displayed in the More Actions menu for the System Navigator are determined by a combination of the following:

- Additional Options
- Display settings
- Format Control
- The RAD application currently running

Additional options are only available in select applications. You can use Format Control to create additional capabilities for users that allow extended user rights and additional access to data gathering, such as additional views and use of the Find and Fill functions.

Icons

More Actions menu: ▾
Find function: 
Fill function: 

For additional information about Additional Options and Format Control, see the related topics.

Open a record using Database Manager

Applies to User Roles:

System Administrator

To open a record using Database Manager:

1. Click **Tailoring** > **Database Manager**.
   Note: Alternatively, you can open Database Manager from the command line, by typing db, and then pressing **Enter** or clicking **Execute Command**.

2. Select a form or file by typing the name of a form in the **Form** field or the name of a file in the **Table** field.

   If one or both fields are left blank, the record list returns all records that match your criteria.

3. Click **Search**.

4. Enter your search criteria. For example, you can enter known information in key fields, such as names of operators, names of devices, or incident numbers.

   Note: Executing a search that includes a field that is not fully keyed slows response time.

5. Click **Search**.

   - If found, a record that matches your criteria opens.
   - If more than one match is found, a record list opens.

     For multiple returns, click a record from the record list to display it.

Note: A *device* is also referred to as a Configuration Item (CI).

Record list

A record list is a specific form displaying pertinent field values from each record found in the search. If you have not created a customized record list form for this list, HP Service Manager will create one (<*filename*>.qbe) based on key fields defined in the database dictionary for the file.
Select a record from the list by positioning the cursor on the desired record and double-clicking or pressing Enter.

- If the search criteria uniquely defines only one record, that record opens using the form used to define the search or the one defined by a form name in the record, depending on format control.
- If the search criteria does not find any records, Database Manager returns the message: *No records found to satisfy record list search argument(s)*.

Refresh a record list

Applies to User Roles:

System Administrator

To refresh a record list:

1. Open the More Actions menu.
2. Select **Refresh List**.

Edit a record

Applies to User Roles:

System Administrator

To edit a record:

1. Click **Tailoring** > **Database Manager**.
2. Place your cursor in the Table Name field, and then click **Search**.
3. Double-click a record to view or change it. Service Manager displays each record in its appropriate form.
4. Update any field within the record.
5. If you make changes, click **Save**.

Note: The record node is a convenient way to view a record, but you may want to make changes by accessing the record through the appropriate application.

Multiple-record functions

Multiple-record functions allow you to perform add, update, delete, and print functions on multiple records within a database. The Mass functions (Mass Add, Mass Update, Mass Delete) become available whenever a query produces a record list, that is, the desired function acts upon the records in the record list.

Note: The Mass functions may require use of the Administration Mode check box when starting Database Manager.

The topics below comprise an example of how to use multiple-record functions. Follow the steps in this order. If you skip a step, the example will not work.

1. "Add multiple records using a literal value" on page 263
2. "Add multiple records using a variable value" on page 266
3. "Update multiple records using a literal value" on page 270
4. "Update multiple records using a variable value" on page 271
5. "Avoid invalid duplicate or null key errors" on page 268
6. "Delete multiple records" on page 272
7. "View multiple records using an alternate form" on page 274

Adding multiple records

After opening a record list from an HP Service Manager file, the Mass Add function allows you to add a block of records to the database. The new records are exact duplicates of those in the original record list. You can also execute format control triggers and macros by setting the **add** field to *true*. Mass Add processing statements use the $file parameter to access fields on records in the current record list. For example, *name* in the $file parameter or the number one (*1*) in the $file parameter.

You can choose to add exact duplicates only to databases where keys are defined as Nulls & Duplicates or there are No Nulls in the database dictionary. Adding duplicate records in any other manner causes Database Manager to issue an invalid duplicate key error message for each record. To avoid this error when executing a Mass Add function against a file with Unique or No Duplicates keys, execute processing statements during the Mass Add function that will manipulate the contents of the new records.

Mass Update

Mass Update enables a user (ADMIN or SYSADMIN) to update one or several fields in a set of records. Mass Update consists of three related functions, template Mass Update, Simple Mass Update, and Complex Mass Update. In a template Mass Update, a template displays a list of fields that can be updated. The user enters a value for each field to be updated to complete the process. In a Simple Mass Update, Service Manager displays a record form in which the user can enter values for selected fields. These values are then applied to all of the records selected for the Simple Mass Update. In a Complex Mass Update, Service Manager displays a form in which the user enters RAD expressions containing variables and concatenated fields to populate another field in the selected records.

The Mass Update function can be used with the following applications:

- Change Management
- Configuration Management
- Incident Management
- Problem Management
- Service Desk

The Database Manager also uses Mass Update to update the following tables:

- contacts
- company
- department (dept)
- location
- operator

The Mass Update functions in Database Manager also let you modify the contents of each record in the list of records accessed from Database Manager, by entering one or more processing statements, which Service Manager executes during adds, updates, and deletes. When add is set to true, Service Manager also executes Format Control, triggers, and macros, if present for the form.

Mass add/update function errors

If Service Manager encounters an error in any record during a Mass Add or Mass Update, it does not make any changes to the record. Instead, it prompts the user to do one of the following:

- Fix the errors in the record and **Retry** the add or update function
- **Skip** to the next record
- **Force** updates to the record

Update multiple records with Mass Update

Applies to User Roles:

System Administrator

You can update multiple records with Mass Update by using the Mass Update template form, Simple Update, or Complex Update. Choose one of the following methods to update multiple records.

Update multiple records with Mass Update by using the Mass Update template form

To update multiple records with Mass Update by using the Mass Update template form:

1. Click **System Administration > Base System Configuration > Companies**.
2. Click **Search**.
3. Select the records in the record list you want to update.
4. Click **Mass Update**.
5. Type the value for the field in the text box or use the Find feature to display a list of potential values for the field.
   Note: The value(s) you enter in the form are propagated to all of the records you selected.
6. Click **Next**.
7. Continue updating fields and click **Execute** when you complete your updates.

Note: If you exit the Mass Update template form or Complex Update form without pressing **Execute**, all of your changes are cleared on the respective form.

Update multiple records with Mass Update by using Simple Update

To update multiple records with Mass Update by using Simple Update:

1. Click **System Administration > Base Configuration > Companies**.
2. Click **Search**.
3. Select the records in the list you want to update.
4. Click **Mass Update**.
5. Click **Simple Update**.
6. Type new values into the appropriate fields on the form.
7. Click **Execute**.

Update multiple records with Mass Update by using Complex Update

To update multiple records with Mass Update by using Complex Update:

1. Click **System Administration > Base Configuration > Companies**.
2. Click **Search**.
3. Select the records in the list you want to update.
4. Click **Mass Update**.
5. Click **Complex Update**.
6. Type the RAD code in the Complex Update form.
7. Click **Execute**.

Template mass update and required fields

Template mass update processes one record at a time, updating the record with the values that you enter in the template form. If a record is missing required fields, then Service Manager asks for the missing field(s), one at a time.

For example, the change queue tasks out-of-box are missing required fields Initial Impact Assessment and Urgency. When you use template mass update to modify these records and you do not enter a value for Initial Impact Assessment and Urgency, Service Manager displays "The Initial Impact Assessment is required". This message pertains to the present record that is being updated. If you enter an Initial Impact Assessment in the template form, then Service Manager prompts you for the missing Urgency for that record.

If there are three out-of-box change queue task records to modify using template mass update, and the second and third records are missing two required fields (for example, the Initial Impact Assessment and Urgency field values) and you enter a new Downtime End in the template form, Service Manager responds by:

1. Updating the value of the Downtime End field you entered in the first record.
2. Prompting for the missing field(s) for each record. When the Initial Impact Assessment and Urgency field values are entered in the template form, the second record is updated with the Downtime End, Initial Impact Assessment, and Urgency.

3. Prompting for the missing fields for the next record. When the Initial Impact Assessment and Urgency are added to the template, the third record is updated with the Downtime End, Initial Impact Assessment, and Urgency.

In summary, the data you originally added, the Downtime End, is never lost. The third record might have a different Initial Impact Assessment and Urgency from the second record, and since you entered no Initial Impact Assessment and Urgency in the template at first, Service Manager asks you for the Initial Impact Assessment and Urgency for the second and for the third record, one at a time.

Updating multiple records

The Mass Update functions let you modify the contents of each selected record in the list of records accessed from Database Manager. You enter one or more processing statements, which Service Manager executes during adds, updates, and deletes.

Mass Update processing statements use the “$file” parameter to access fields on records in the current record list. For example, “name in $file” or “1 in action in $file”.

Add multiple records using a literal value

Applies to User Roles:

System Administrator

Example: Duplicate all contact records for advantage. Retain the same information in each record, with the exception of changing the **Location** to North America. Since Contact Name (contact.name) is a Unique key and Employee ID (user.id) is a No Duplicates key in the contacts file, you must assign unique contact.name and user.id values to the new records. For additional information about key type definitions, see the related topics. In this example, append the value *-NA* to the contact.name and user.id values to maintain uniqueness.

To mass add records using a literal value:

1. Open the form in Database Manager. Ensure that you select the Administration Mode check box. In this example, open the following form: `contacts`. A blank contacts form opens.

2. Enter the desired value in the **Company** input field. For this example, select advantage from the Company list.

   Click **Fill**. to open the list.

3. Click **Search**.

   Database Manager performs the equal to search and displays the matching records, if any.

   If the query produces no matching records, Service Manager displays a message.

4. To display the Mass function buttons, use a query that returns more than one record.

   Select one of the listed records, and then select **Mass Add**. from the tool bar to start the process of duplicating the records shown in the list.

   Database Manager displays the Mass Add/Update Instruction screen.

   Type the applicable statements in the input fields.

   - Statements in the first input field (**Instructions to be executed ONCE at the beginning of Mass Add/Update**) are executed once at the beginning of the add.
   - Statements in the subsequent input fields (**Instructions for action on EACH RECORD**) are executed against every record as it is added.

   For this example, leave the first input field blank and enter the following assignment statements in the **Instructions for action on EACH RECORD** input fields.

Statement	Purpose
contact.name in $file=contact.name in $file +"-NA"	Appends -NA to the end of all contact.name values.
location in $file="North America"	Changes the location value to North America.
user.id in $file=user.id in $file +"-NA"	Appends -NA to the end of all user.id values.

5.

6. Click **Execute**,

   Database Manager performs the Mass Add in the foreground. Your session is devoted to the add task until it is completed. If you are updating a large number of records, this can take a while.

   Upon completion, Service Manager returns terminal control to you and displays the following message: *<nnn>records added to the contacts file* where *nnn* is the number of records added.

7. Click **View Messages** to check the messages for errors.

- A blue icon indicates a required action.
- A yellow icon indicates an informational or warning message.
- A red icon indicates an error message.

Close the Messages window to return to the contacts form.

- Click the More Actions icon and choose **Refresh**.

  In the new records, the Contact Name (contact.name) and Employee ID (user.id) values are the same as the original records with -NA appended. The Location (location) values are now North America.

  The original records listed before the Mass Add remain in the contacts file with no changes to the original data.

- Click **OK**. to return to the contacts form.

- In the contacts form, do the following:
  - Type **North America** in the Location field.
  - Click **advantage** in the Company field.
  - Type **Search**.

  Database Manager performs the equal to search and displays the matching record(s), if any, in the contacts.qbe form.

The topics below comprise an example of how to use multiple-record functions. Follow the steps in this order. If you skip a step, the example will not work.

1. "Add multiple records using a literal value" on page 263
2. "Add multiple records using a variable value" on the next page
3. "Update multiple records using a literal value" on page 270
4. "Update multiple records using a variable value" on page 271
5. "Avoid invalid duplicate or null key errors" on page 268
6. "Delete multiple records" on page 272

Add multiple records using a variable value

Applies to User Roles:

System Administrator

Example: Duplicate all "contacts" records with a company.name value of advantage. Include the same information in each record, except for changing the Company to NewAdvantage, creating sequential user.id values, and creating a Unique key for each record.

Since Contact Name (contact.name) is a Unique key and Employee ID (user.id) is a No Duplicates key in the contacts file, you must assign unique contact.name and user.id values to the new records. For additional information about key type definitions, see the related topics. For this example, suffix the user.id values in the new records with sequential numbers, starting with 1000 (1000, 1001, and so on.)

Note: Mass Add processing statements reference $file to access fields on records in the list. For example, name in $file or 1 in action in $file.

To add records using a variable value:

1. Open the form in Database Manager. Ensure that you select the Administration Mode check box. For this example, open the following form: `contacts`. A blank contacts form opens.

2. Enter the desired value in the **Location** field.

   For this example, type **North America**, and then click **Search**.

   Database Manager performs the equal to search and displays the Record List of matching record (s), if any. If the query produces no matching records, Service Manager displays a message.

3. To display the Mass function buttons, use a query that returns more than one record.

   Select **Mass Add** from the tool bar to start the process of duplicating the records shown in the Record List.

   Database Manager displays the Mass Add/Update Instruction screen.

4. Type the following assignment statement in the **Instructions to be executed ONCE at the beginning of Mass Add/Update** input field.

Statement	Purpose
$sequential.number=1000	Sets the start point for the variable at 1000 and will be executed only once for the entire function.

5. Enter the following statements on the **Instructions for action on EACH RECORD** input fields:

Statement	Purpose
user.id in $file="NewAdvantage"+str ($sequential.number)	Sets the User Id to NewAdvantage with a sequential number as a suffix.
$sequential.number+=1	Causes one (1) to be added to all future numbers created from this variable.
contact.name in $file=first.name in $file + " " + last.name in $file	Makes the contact name the same as the first and last name.

6. Click **Execute**.

   Database Manager performs the Mass Add function in the foreground. Your session is devoted to the add task until it is completed. If you are updating a large number of records, this can take a while.

   Upon completion, Service Manager returns terminal control to you and displays the following message:*<nnn>records added to the contacts file* where *nnn* is the number of records added.

7. Click **View Messages** to check the messages for errors.
   - A blue icon indicates a required action.
   - A yellow icon indicates an informational or warning message.
   - A red icon indicates an error message.
8. Close the Messages window to return to the contacts form.

The topics below comprise an example of how to use multiple-record functions. Follow the steps in this order. If you skip a step, the example will not work.

1. "Add multiple records using a literal value" on page 263
2. "Add multiple records using a variable value" on the previous page

3. "Update multiple records using a literal value" on page 270
4. "Update multiple records using a variable value" on page 271
5. "Avoid invalid duplicate or null key errors" below
6. "Delete multiple records" on page 272

Avoid invalid duplicate or null key errors

Applies to User Roles:

System Administrator

Records cannot have duplicate Unique or No Duplicates keys. This example shows one such error and how to fix it.

To handle an invalid duplicate of NULL key error:

1. Open the form in Database Manager. For this example, open the following form:`contacts`. Be sure to select the Administration Mode check box.

   A blank contacts form opens.

2. Type **NewAdvantage** in the Employee ID field., and then click **Search**.
3. Select **Mass Update**. from the tool bar to begin updating the records shown in the Record List.

   Database Manager displays the initial form again.

4. On the displayed form, type **NewAdvantage** in the Contact Name field.
5. Click **Simple Update**.
   - The system accepts the first update, because the contact name is unique.
   - The system rejects the second update, since changing it would create a duplicate contact name.

   Database Manager opens the record for modification and displays an error message.

6. Click **View Messages** to check the messages for errors.

- A blue icon indicates a required action.
- A yellow icon indicates an informational or warning message.
- A red icon indicates an error message.

7. At this point, there are several options concerning the outcome of this process.
   - Enter a unique key and click **Retry** for each item on the list separately, to cause Service Manager to reexecute everything, including the Format Control.
   - Click **Skip** to cause Service Manager to skip the record.
   - Click **Force** to cause Service Manager to update the record with the information you entered, regardless of Format Control.

   For this example, modify the **Contact Name** field value to make it unique for each item. (NewAdvantage2, NewAdvantage3, NewAdvantage4, and so on.)

   If the update was successful, Service Manager displays the following message: *"<n>records updated in the contacts file."*

8. Type **NewAdvantage** in the Employee ID (user.id) field, and then click **Search**.

   Database Manager displays the records you have just added in a record list.

   The Employee ID values consist of Advantage + 1000, 1001, and so on, and the Contact Name values the first and the last names. The original records listed before the Mass Add remain in the contacts file, with no changes to the original data.

The topics below comprise an example of how to use multiple-record functions. Follow the steps in this order. If you skip a step, the example will not work.

1. "Add multiple records using a literal value" on page 263
2. "Add multiple records using a variable value" on page 266
3. "Update multiple records using a literal value" on the next page
4. "Update multiple records using a variable value" on page 271
5. Avoid invalid or null key errors
6. "Delete multiple records" on page 272

Update multiple records using a literal value

Applies to User Roles:

System Administrator

Example: modifying the Service Contract field value to HP VIP Service for the contacts records for all advantage employees in North America.

To update multiple records with a literal value:

1. Log in to the Service Manager Windows client.
2. Open the contacts form in Database Manager. Ensure that you select the Administration Mode check box.
3. Enter the desired value in the **Company** input field. For this example, select advantage from the list.
4. Enter the desired value in the **Location** input field. For this example, type North America.
5. Click **Search**.

   Service Manager displays all matching records in a record list. If the query produces no matching records, Service Manager displays a message. To display the Mass function buttons, use a query that returns more than one record.
6. Select **Mass Update** from the tool bar to start the process of updating the records shown in the record list.
   Database Manager re-displays the initial form with new option buttons.
7. Select **Simple Update** from the toolbar.
8. Enter the desired changes. For this example, select HP VIP Service from the **Service Contract** drop-down list.
9. Click **Execute**.
   The system returns terminal control to you and displays the newly updated records in the record list, along with the message *"nnn records updated in the contacts file"*, where *nnn* is the number of records updated.

The topics below comprise an example of how to use multiple-record functions. Follow the steps in this order. If you skip a step, the example will not work.

1. "Add multiple records using a literal value" on page 263
2. "Add multiple records using a variable value" on page 266
3. Update multiple records using a literal value
4. "Update multiple records using a variable value" below
5. "Avoid invalid duplicate or null key errors" on page 268
6. "Delete multiple records" on the next page

Update multiple records using a variable value

Applies to User Roles:

System Administrator

Example: Change the email address for all Advantage contacts records for people located in North America to follow the pattern: firstname.lastname@advantage.com.

To update multiple records with a variable value:

1. Open the contacts form in Database Manager. Ensure that you select the Administration Mode check box.
2. Enter the desired value in the **Company** input field. For this example, select advantage from the Company list.
3. Enter the desired value in the **Location** input field. For this example, type `North America`.
4. Click **Search**.
   Database Manager performs the full search and displays the Record List of matching records, if any, in the **contacts** form.
5. Select **Mass Update** from the tool bar to start the process of updating the records listed.
   Database Manager re-displays the initial form with different option buttons.
6. Do not enter values in any field. Click **Complex Update**.
   Database Manager displays the Mass Add/Update Instruction screen.
7. Enter the following assignment statement in the **Instructions for action on EACH RECORD** input field, as shown:

Statement	Purpose
email in $file = first.name in $file + "." + last.name in $file + "@advantage.com"	Sets the email address to firstname.lastname@advantage.com.

The system executes this instruction for each record updated. Then it returns terminal control to you, and displays the message *"nnn records updated in the contacts file"*, where *nnn* is the number of records updated.

The topics below comprise an example of how to use multiple-record functions. Follow the steps in this order. If you skip a step, the example will not work.

1. "Add multiple records using a literal value" on page 263
2. "Add multiple records using a variable value" on page 266
3. "Update multiple records using a literal value" on page 270
4. "Update multiple records using a variable value" on the previous page
5. "Avoid invalid duplicate or null key errors" on page 268
6. "Delete multiple records" below

Delete multiple records

Applies to User Roles:

System Administrator

Example: Delete the contacts for advantage with a variant of NewAdvantage as the contact name.

To delete all records with a specified value:

1. Click **Tailoring** > **Database Manager**.
2. Select **Administration mode**.
3. For this example, open the `contacts` form.
4. Enter the desired value. For this example, type **NewAdvantage*** in the Contact Name field.
5. Click **Search**.

Database Manager performs the search and displays the record list of matching records, if any, in the contacts.qbe form.

If the query produces no matching records or only one matching record, then Service Manager does not display a record list.

Note: To display the Mass function buttons, use a query that returns more than one record.

6. Select **Mass Delete** from the tool bar to remove the records shown in the record list from the database.

   Database Manager confirms the delete request with a prompt screen.

7. Do one of the following:
   - To cancel the delete, click **No**.
   - To confirm the delete, click **Yes**.

     Database Manager deletes all records in the list.

     Note: Service Manager performs the Mass Delete function in the foreground, which means your terminal is devoted to this function until complete. If you are deleting a large number of records, this can take a while.

Upon completion, Service Manager returns terminal control to you. A blank contacts form opens with the message: *"nnn records deleted from the contacts file"*, where *nnn* is the number of records deleted.

The topics below comprise an example of how to use multiple-record functions. Follow the steps in this order. If you skip a step, the example will not work.

1. "Add multiple records using a literal value" on page 263
2. "Add multiple records using a variable value" on page 266
3. "Update multiple records using a literal value" on page 270
4. "Update multiple records using a variable value" on page 271
5. "Avoid invalid duplicate or null key errors" on page 268
6. "Delete multiple records" on the previous page

View multiple records using an alternate form

Applies to User Roles:

System Administrator

Windows client:

To view a list of records using a predefined alternate form:

1. From the Window Menu select **Preferences...**.
2. Select **HP Service Manager** > **Appearance**.
3. Verify that the option **Show detail pane with record list** is unchecked. If it isn't, uncheck it. Log out and log back in.
4. From a list of records click the **More Actions** menu.
5. Select **Alternate Forms**.
6. Select a form to use to display the list.
   Note: This option will be available only when there are additional forms specified for the list of records on the formatcontrol and the condition(s) specified are met.

Web client:

To view a list of records using an predefined alternate form:

1. Locate the web.xml document belonging to your Service Manager installation. For example: C:\Program Files\Apache Software Foundation\[tomcat directory]\webapps\[ServiceManager web directory]\WEB-INF\web.xml
2. Open it for edit.
3. Set the parameter "viewrecordlist" to false.
4. Restart the Apache Tomcat service (**Windows Control Panel** > **Administrative Tools** > **Services**)
5. From a list of records click the **More Actions** menu.
6. Select **Alternate Forms**.

7. Select a form to use to display the list.
   Note: This option will be available only when there are additional forms specified for the list of records on the formatcontrol and the condition(s) specified are met.

On the Web client, you can also:

1. Logout.
2. Add the following parameter to the end of the URL:
   `?viewrecordlist=false`
3. Example:

   ```
   http://localhost:8080/ [ServiceManager web directory]
   /index.do?viewrecordlist=false
   ```

Mass Close

The Mass Close function enables a user to select and then close a set of incident or interaction records simultaneously. For the out-of-box systems, the ADMIN and SYSADMIN profiles include the Mass Close capability. In Service Desk, you can use Mass Close for interaction lists and interaction queues. In Incident Management, you can close multiple incident records.

Mass close interaction records

Applies to User Roles:

System Administrator

For the out-of-box systems, the ADMIN and SYSADMIN profiles include the Mass Close capability. You can use Mass Close for interaction lists and interaction queues. Choose one of the following methods to mass close interactions.

Mass close interaction records after searching for interactions

To mass close interaction records after searching for interactions:

1. Click **Service Desk** > **Search Interaction Records**.
2. Use Search to display a list of interaction records.
3. Select the records in the record list you want to close.

4. Click **Mass Close**.
5. Edit the text present for the resolution of the interaction or accept the text.
6. Click **Finish**.

Mass close interaction records from the Interaction Queue

To mass close interaction records from the Interaction Queue:

1. Click **Service Desk** > **Interaction Queue**.
2. Specify the Queue and the View.
3. Select the records in the record list you want to close.
4. Click Mass Close in the More Actions menu.
5. Specify the Resolution Code. Use the Fill icon to display a list of available codes.
6. Click **Next**.
7. Accept or edit the text in the Resolution text box.
8. Click **Finish**.

Determine the number of records in a record list

To determine the number of records in a record list:

1. Open the `contacts` form in Database Manager.
2. Select the contacts file from the record list.

   A blank contacts record opens.
3. Click **Search**.

   The contacts format opens with a record list at the top and the **Count** option in the toolbar and in the More Actions menu.
4. From the More Actions menu, choose **Count** to get the record count.

HP Service Manager displays the record count on the **Count** button.

Note: If you select **Count** from the tool bar, a prompt opens asking whether all records in the list should be counted. Confirm by clicking **Yes**.

The system counts the records and displays the following message: *"There are<n>records in this list"*.

Print multiple records

Applies to User Roles:

System Administrator

Example: Print all contact records.

Open the contacts form in Database Manager, do not enter any values and then click **Search**. Database Manager performs the true search and displays a complete list of all the contact records.

Note: If the query produces no matching records or only one matching record, then Service Manager does not display a record list.

Follow the instructions in Print options to print in the windows client or the web client.

Refresh a record list

Applies to User Roles:

System Administrator

To refresh a record list:

1. Open the More Actions menu.
2. Select **Refresh List**.

Options for Mass Update

The following are the available options for Mass Update if validations on format control fail:

Option	Action
Retry	Reruns the mass update function on the selected records. HP Service Manager once again validates each record in the mass update against its format control.
Skip	Skips the current record without making any changes.
Force	Updates the current record with whatever information that you entered, regardless of format control.

Single-record functions

Single-record functions include adding, updating, deleting, and printing individual records within a database.

Add a record

Applies to User Roles:

System Administrator

Example: Add a record for the new contact, Bob Hoskins, to the contacts file.

To add a contact:

1. Open a form in Database Manager. For this example, open the `contacts` form.
2. Select the contacts file from the record list, and then click **Search**.
3. Enter the required values in the Business Information section of the Business tab, and other values as needed. For this example, enter:

Field	Record value
Contact Name	HOSKINS,BOB
Company	Advantage

4. Click **Add**..

Database Manager adds the record, retains the input from the screen displayed and responds with the following message: *"Contact Information record added."*

Clear an initial record

Applies to User Roles:

System Administrator

If you need to redo an incorrect record or need otherwise to clear all data from an initial record form, you can remove this data quickly and completely using the Clear command.

To clear all data from an initial screen:

1. Open the form in Database Manager. For this example, open the `contacts` form.
2. Select the record from the record list.
3. Open the More Actions menu.
4. Select **Clear**.

Delete a record

Applies to User Roles:

System Administrator

Example: Delete the contact record you added for Richard Butler by following the steps in the task"Duplicate an existing record" below.

To delete a record:

1. Open a form in Database Manager. In this example, open the **=contacts** form.
2. Search for the existing record by entering **BUTLER, RICHARD** in the Contact Name field and clicking **Search**.
3. Click **Delete**.

   Confirm the action at the prompt.
4. Click **Yes** to continue with the record deletion. If successful, the following message displays: *Contact Information record deleted.*

Duplicate an existing record

Applies to User Roles:

System Administrator

Example: Add a new record to the contacts table that is a near duplicate of an existing record. For this example, all information except the Primary Configuration item (CI) and Contact Name will be the same.

To duplicate an existing record:

1. Open the form in Database Manager. For this example, open the `contacts` form.
2. Select the contacts file from the record list.

   A blank contacts record opens.
3. Click **Search**.
4. Select CHAN, TERESA from the returned list.
5. Select the Business tab.
6. Type a new name in the **Contact Name** field.

   For this example, type **BUTLER, RICHARD**.
7. Enter a new **Primary Configuration Item** for this record.
8. Delete the **Employee ID**.

   Add your own at this point, or leave this field blank.
9. Click **Add**.

Database Manager adds the record, retains the input from the screen displayed and responds with the following message: Contact Information record added.

Print a record

Applies to User Roles:

System Administrator

Example: Printing an existing contacts record for Richard Butler.

To print a record:

1. Open the contacts form in Database Manager.
2. Select the contacts file from the record list, and then press Enter. A blank contacts file opens.
3. Click **Search**.
4. Select Richard Butler from the returned list.
5. Follow the instructions in Print options to print in the windows client or the web client.

Search for newly added records

Applies to User Roles:

System Administrator

To search for newly added records:

1. Click **Tailoring** > **Database Manager**.
2. Place the cursor in the **Table** field, and then click **Search**.
3. Click the **Last Updated** column header to sort the table of records returned by date and time.

Update an existing record

Applies to User Roles:

System Administrator

Example: Update the existing contacts record for Teresa Chan, whose primary Configuration Item (CI) has an identification number of *167*, and change the value to *adv-nam-desk-267*.

In this example, we use the **contacts** form to change the identification number of the Teresa Chan's primary CI, from *adv-nam-desk-167* to *adv-nam-desk-267*.

Note: Audit Specification Records must be defined for Configuration Items prior to modifying.

To update an existing record:

1. Click **Tailoring** > **Database Manager**.
2. Type `=contacts` in the **Form** field, and then click **Search**.
3. On the Contact Information form, place the cursor in the Contact Name field, and then click **Search**.

4. Select **CHAN, TERESA** from the record list returned and select the **Business** tab.
5. In the **Primary Configuration Item** field, type `adv-nam-desk-267` to replace the current value.
6. Click **Save**.

View a single record using an alternate form

Applies to User Roles:

System Administrator

To view a record using predefined alternate form:

1. From a detail record click the **More Actions** menu.
2. Select **Alternate Forms**.
3. Select a form to use to display the record.

Note: This option will be available only when there are additional forms specified for the record on the formatcontrol and the condition(s) specified is met.

Queries

Queries are logical expressions that evaluate to either true or false against the records queried. Click **Tailoring** > **Database Manager** to search for records matching your query. For records that evaluate to true, those records display in a table of matching records. You can click on each record for details. For records that evaluate to false, no records are returned and an error message is displayed.

Simple query expressions

HP Service Manager offers several methods to perform queries. Simple queries using the Query Window method are most useful in the following situations:

- On forms where a record list query cannot be performed.
- When performing range queries.
- When a field exists in the database dictionary record of the file, but is not displayed on the form.

- When the input field on the form is not long enough to contain the number of characters needed for the value you want to query.

Note: The Query Window Application form, named query.window, is used in Database Manager, Incident Management, and Configuration Management. In addition to the query string, you can optionally specify sort fields to further filter your results. You must have the query.window capability word enabled in your operator record to open the Query Window.

Complex query expressions

HP Service Manager offers several methods to perform complex queries. These include:

- Logical operators, such as AND, OR, and NOT
- Relational Operators, such as greater than, less than, and equal to
- RAD functions, such as index() and tod()
- IR Expert searches

Logical operators

Logical operators evaluate Boolean expressions and determine whether the entire expression is true or false based on the operator used.

You can use logical operators to do the following:

- Construct simple queries that return true, false, or unknown results using Boolean logic.
- Perform queries using combinations of logical operators that are not available to use with the Record list query method, use the Retrieving records method from the query window.

Operator	Symbol	Usage	Operation
AND	&	*value* AND *value*	Returns TRUE if both component conditions are TRUE. Returns FALSE if either is FALSE. Otherwise returns UNKNOWN.
OR	|	*value* OR *value*	Returns TRUE if either component condition is TRUE. Returns FALSE if both are FALSE. Otherwise returns UNKNOWN.
NOT	~	not value	Returns TRUE if the following condition is FALSE. Returns FALSE if it is TRUE. If it is UNKNOWN, then it remains UNKNOWN.

Logical operators take precedence in the following order: (1) NOT, (2) AND, (3) OR.

Results of applying the NOT operation to an expression:

--	**TRUE**	**FALSE**	**UNKNOWN**
NOT	FALSE	TRUE	UNKNOWN

Results of combining two expressions by using AND:

AND	**TRUE**	**FALSE**	**UNKNOWN**
TRUE	TRUE	FALSE	UNKNOWN
FALSE	FALSE	FALSE	FALSE
UNKNOWN	UNKNOWN	FALSE	UNKNOWN

Results of combining two expressions by using OR:

OR	**TRUE**	**FALSE**	**UNKNOWN**
TRUE	TRUE	TRUE	TRUE
FALSE	TRUE	FALSE	UNKNOWN
UNKNOWN	TRUE	UNKNOWN	UNKNOWN

Relational operators

Relational operators compare values and return True, False, or Unknown. Whenever a comparison is executed against a null value, HP Service Manager returns Unknown. For example, if you are searching for all contact records matching the query `shift="day"` all contact records without a value in the shift field will return Unknown.

The most efficient relational operator in a query is the equal to (=) relational operator, for which exact results are returned. Although queries using this operator are valid for all field types, it is seldom used for queries to date or time fields. For date and time fields, the greater than (>) and less than (<) operators are more efficient because the values returned when using these operators include seconds.

Relational Operator	**Definition**
#	starts with
~#	does not start with
=	equal to

Relational Operator	Definition
~=	not equal to
<>, ><	not equal to
<	less than
<= or =<	less than or equal to
>	greater than
>= or =>	greater than or equal to
isin	is element in
like	is similar to

Starts with relational operator

In a search screen, the starts with (#) relational operator is the default query type, unless you use a wildcard character. If the value you enter contains wildcard characters, the like relational operator is assumed. To use the starts with operator in conjunction with wildcard characters, enter the starts with (#) operator as the first character in your query.

In the query window, everything is interpreted literally as you type it in.

Equal to relational operator

The equal to (=) relational operator is the most efficient relational operator and should be used when possible. Queries that use the equal to relational operator are valid on all field types although it is not useful to query date and time fields because the results are literal.

Less than relational operator

The less than (<) operator can be used to query any field type. It is most frequently used to query date and time fields because the values returned include seconds, which makes using the equal to (=) operator for date and time fields impractical.

Greater than relational operator

The greater than (>) relational operator can be used to query any field type. It is most frequently used to query date and time fields because the values returned include seconds, which makes using the equal to (=) operator for date and time fields impractical.

Example Scenario:

Suppose you want to find records opened at 11:00 AM on June 1. Since it is not likely that the record was opened at *exactly* at 11:00 AM and zero seconds (11:00:00). You are more likely to get the results you want by using a range of times. For example, if you type `>06/01/2008 11:00` in the **open.time** field, all records opened after 11:00 AM, starting with 11:00:01, are returned.

Using ranges in queries

A range is defined by using a combination of less than (<) and greater than (>) relational operators between the *and* logical operator or the *or* logical operator.

To query of a range of data, use the Query Window.

Operator	Usage	Examples
AND **and**	><*value* and <*value*	*field_name* > *lesser_value* and *field_name* < *greater_value* For example, open.time>*'09/15/2008'* and open.time<*'09/27/2008'* The *and* logical operator is not case-sensitive. You can use (&) as a shortcut for the *and* logical operator.
OR **or**	<*value* or >*value*	*field_name* < *lesser_value* or *field_name* > *greater_value* For example, open.time<*'09/15/2008'* or open.time>*'09/27/2008'* The *or* logical operator is not case-sensitive. You can use (|) as a shortcut for the *or* logical operator.

Like relational operator

The like relational operator allows you to search for character strings similar to the values entered. The like operator is only valid on character fields.

Wildcard characters

By default, HP Service Manager uses the question mark (?) as the wildcard that represents a single character, the asterisk (*) to represent multiple characters, and the backslash (\) as the escape character. Therefore, for a literal search on the * or ? character, you must immediately precede that character with a backslash (\), such as `\?`.

Note: You can modify the wildcard characters and the escape character by using the **wildcardcharacters** parameter in the Service Manager server **sm.ini** file.

Example Scenario:

Suppose you want to find all records containing last names where the second character is *'e'*.

To search for any character followed by the letter e followed by the asterisk (*) character, you would type `?e\*` in the **Last Name** field to generate a *like* query. Although the question mark (?) is a wildcard character, it verifies that the last name ends with a literal *.

The query would retrieve and display records with last names that begin with:

```
Ae*
ae*
?e*
*e*
...
```

Create a query using not in statements

Applies to User Roles:

System Administrator

Example: Retrieve all contacts records where the value of the manufacturer is not *advantage* and the not symbol (~) is unavailable on the keyboard.

In this example, we use the *not* relational operator to perform an Advanced Search on records in the **contacts.qbe** form and return a list of all contacts that are not employees of Advantage.

To create a query using not in statements:

1. Click **System Administration** > **Base System Configuration** > **Contacts**
2. On the Contact Information form, open the More Actions menu.
3. Select **Expert Search**.

   The Query window opens.
4. In the **Query** field, type `not (company="advantage")`, and then click **Search**.
5. Click any record in the list of returned records to view the details.

Create a query using an or statement or an and statement

Applies to User Roles:

System Administrator

Example: Retrieve all *contacts* records where the Company is either *DEFAULT* or *advantage*, and the value in the **Contact Name** field begins with the letter *B*.

In this example, we perform a complex query using a combination of logical operators to query the **contacts.qbe** form.

To create a query using an or statement or an and statement:

1. Click **Tailoring** > **Database Manager**.
2. Type `contacts` in the **Form** field, and then click **Search**.
3. Double-click the **contacts** form from the list of returned records.
4. On the **Contact Information** form, open the More Actions menu.
5. Select **Expert Search**.
6. In the **Query** field, type `(company="advantage" or company="DEFAULT") and contact.name#"B"`, and then click **Search**.
   This query may result in a partially keyed search. If so, you can optionally enter a time limit in your query prior to executing.
7. Click any record in the list of returned records to view the details.

Note: If you type the same expression without using parentheses, such as `company="advantage" or company="DEFAULT" and contact.name#"B"`, the *and* operator takes precedence over the *or* operator and returns a different set of records. This query is less restricted and returns records with a value of either *DEFAULT* or *advantage*.

Use like queries to search for a string of characters

Applies to User Roles:

System Administrator

Example: Use a like query in combination with a field function to retrieve all records for vendors/suppliers that begin with the character string *DEFAULT*.

In this example, we use the like relational operator on the **vendor.qbe** form to query the **vendor** field for vendors/suppliers that begin with *DEFAULT*, followed by any character or character string.

To use like queries to search for a string of characters:

1. Click **Tailoring** > **Database Manager**.
2. Type `=device` in the **Form** field, and then click **Search**.
3. On the Asset Information form, open the More Actions menu.
4. Select **Expert Search**.
5. In the **Query** field, type `vendor like "DEFAULT*"`, and then click **Search**.
6. Click any record in the list of returned records to view the details.

Create a query using an or statement or an and statement

Applies to User Roles:

System Administrator

Example: Retrieve all *contacts* records where the Company is either *DEFAULT* or *advantage*, and the value in the **Contact Name** field begins with the letter *B*.

In this example, we perform a complex query using a combination of logical operators to query the **contacts.qbe** form.

To create a query using an or statement or an and statement:

1. Click **Tailoring** > **Database Manager**.
2. Type `contacts` in the **Form** field, and then click **Search**.
3. Double-click the **contacts** form from the list of returned records.
4. On the **Contact Information** form, open the More Actions menu.
5. Select **Expert Search**.
6. In the **Query** field, type `(company="advantage" or company="DEFAULT") and contact.name#"B"`, and then click **Search**.
   This query may result in a partially keyed search. If so, you can optionally enter a time limit in your query prior to executing.
7. Click any record in the list of returned records to view the details.

Note: If you type the same expression without using parentheses, such as `company="advantage" or company="DEFAULT" and contact.name#"B"`, the *and* operator takes precedence over the *or* operator and returns a different set of records. This query is less restricted and returns records with a value of either *DEFAULT* or *advantage*.

Create a query using array fields

Applies to User Roles:

System Administrator

Example: Retrieve all operator records where the operator has ability to execute tasks with the capability words of SysAdmin or problem management.

In queries using array fields, the position of an element within an array is not relevant. The logical operator *or* is used in array field queries and all records with any of the values entered in the array are retrieved.

In this example, we use the *or* logical operator to search for records in the **operator.g** form that contain the execute capability words of *SysAdmin* or *Problem Management*.

To create a query using array fields:

1. Click **Tailoring** > **Database Manager**.
2. Type `=operator` in the **Form** field, and then click **Search**.
3. Click **Search** again.
4. On the Operator Record form, click **Startup** tab.
5. Type `=SysAdmin` in the first **Execute Capabilities** array field.
6. Type `=problem management` in the second **Execute Capabilities** array field, and then click **Find**.

Click any record in the list of returned records to view the details.

Create a query using more than one field

Applies to User Roles:

System Administrator

Example: Retrieve all device records where the **logical.name** field in the Manage CIs form begins with the letter *H* and the **IP address** field starts with *196.76.209*.

In this example, we perform a search on the **Location** and **IP Address** fields in the **device** form and return a list of device records matching these criteria. Database Manager forms record list queries on scalar, or *non-arrayed* fields, with the *and* logical operator. Searches are case-sensitive, unless you first set up your database for case-insensitive searching.

To create a query using more than one field:

1. Click **Tailoring** > **Database Manager**.
2. Type **=device** in the **Form** field, and then click **Search**.
3. On the Search Configuration Records form, do the following:
   - Type **H** in the **Location** field.
   - Select the Advanced Filter tab, and then click **Add New Filter Criteria**.

     The Add Filter Criteria wizard opens.
   - In the **Field in Configuration Item** field, select the **IP Address** field.
   - In the **Comparison** field select **Starts With**.
   - In the **Value** field, type **196.76.209**.
   - Click **Next**.

     You are returned to the Search Configuration Records form.
4. Open the More Actions menu and choose **Expert Search**.

   You will see your query in the **Query** field.
5. Click **Search**.

   When a partially keyed query is executed, HP Service Manager searches the file for a specified time interval, or until the screen buffer is filled. When either threshold is met, it stops and displays the records retrieved up until that point.

   A warning then displays, indicating an attempt to initiate a partially keyed query. If you receive a warning message of this type, you have limitations in your operator or user profile that prevent the execution of inefficient queries.

6. Add a time interval to your query, and then click **Search** to execute the query again.
7. Click any record in the list of returned records to view the details.

Note: A *device* is also referred to as a *Configuration Item* (CI).

Perform IR Expert queries

Applies to User Roles:

System Administrator

You can perform an *intelligent* query using an information retrieval engine called the *IR Expert utility*. With this utility, you can accurately define queries for a list of helpful topics filtered from your search criteria.

To perform an IR Expert query on the probsummary table:

1. Click **Tailoring** > **Database Manager**.
2. Type `=probsummary` in the **Form** field, and then click **Search**.
3. On the incident search form, click **Text Search**.
4. In the Search Text area, type your search word(s).

   Note: Be sure to insert a comma (,) or white space between your search words if you want to search for any of them. For example, type "**computer, reboot**" or "**computer reboot**".
5. Specify a search type:
   - All of these words - search for records that contain all of your search words.

     For example, if you entered "web browser hangs" in the Search Text field on the incident search form, selecting this option will only search for incident records whose Description or Title field contains "web browser hangs".
   - Any of these words - search for records that contain any of your search words.

     For example, if you entered "web browser hangs" in the Search Text field on the incident search form, selecting this option will search for incident records whose Description or Title field contains "web", or "browser" or "hangs".

- Search only these words - search for your search words only;
- Also search related words (slower) - search for your search words and also their related words.

6. Click **Search**.

   Service Manager displays the search results.

Use the equal to relational operator

Applies to User Roles:

System Administrator

Example: Retrieve all device records where the value of the **Location** field is equal to *DEFAULT HQ*.

In this example, we execute an equal to (=) search from the **device** form and then displays a record list containing all exact matches.

To use the equal to relational operator:

1. Click **Tailoring** > **Database Manager**.
2. Type `=device` in the **Form** field.
3. Click **Search**.
4. Type `=DEFAULT HQ` in the **Location** field on the device form.
5. Click **Find**.
6. Click any record to view the details.

Note: A *device* is also referred to as an *Asset* or *Configuration Item* (CI).

Use the greater than relational operator

Applies to User Roles:

System Administrator

Example: Retrieve all device records where the value of the **logical.name** field is greater than *d*.

In this example, we execute a greater than (>) search from the **device** form and then displays a record list containing all matches with a value greater than the letter *d*.

To use the greater than relational operator:

1. Click **Tailoring** > **Database Manager**.
2. Type `=device` in the **Form** field.
3. Click **Search**.
4. Type `>d` in the **Asset Tag** field on the device form.
5. Press **Enter**.
6. Click any record to view the details.

Note: A *device* is also referred to as an *Asset* or *Configuration Item* (CI).

Use the less than relational operator

Applies to User Roles:

System Administrator

Example: Retrieve all device records where the value of the **IP Address** field is less than *196*.

In this example, we execute a less than (<) search from the **device** form and then displays a record list containing all matches with a value less than the number *196*.

To use the less than relational operator:

1. Click **Tailoring** > **Database Manager**.
2. Type `=device` in the **Form** field.
3. Click **Search**.
4. Type `<196` in the **Serial number** field on the device form.
5. Press **Enter**.
6. Click any record to view the details.

Note: A *device* is also referred to as an *Asset* or *Configuration Item* (CI).

Use the like relational operator

Applies to User Roles:

System Administrator

Example: Retrieve all contact records that contain the letter *e* as the second letter of the last name.

In this example, we execute a like query where the **full.name** field is like *?e** and then displays a record list containing all matching records.

To use the like relational operator:

1. Click **System Administration** > **Base System Configuration** > **Contacts**.
2. Type `?e*` in the **Full Name** field.
3. Press **Enter**.
4. Click any record to view the details.

Note: This query contains two wildcard characters, which triggers Database Manager to automatically generate a like query. A like query does not automatically generate a starts with query. For example, to see all records that start with any character followed by *e** with any number of characters following, type `?e\**` in the **last.name** field. The returned records include those where the **last.name** is like *Ae**, *he*e*, *fe**, *Be*bcdefg*, and so on.

Use the not symbol with logical or relational operators

Applies to User Roles:

System Administrator

Example: Retrieve all device records, excluding those that begin with the letter *C*, in the **Vendor Name** field.

Note: If your database has not first been set up to enable case-insensitive searching, this query returns records that begin with a lowercase letter *c*.

In this example, we use the not relational operator to exclude records in the **device.qbe** form.

To use the not symbol with logical or relational operators:

1. Click **Tailoring** > **Database Manager**.
2. Type `=device` in the **Form** field, and then click **Search**.
3. With the **device** form open, type `C` or `~C` in the **Vendor Name** field, and then click **Find**.

   Note: Using the tilde (~) character is platform-dependent.
4. Click any record in the list of returned records to view the details.

Note: A *device* is also referred to as an *Asset* or *Configuration Item* (CI).

Use the starts with relational operator

Applies to User Roles:

System Administrator

Example: Retrieve all capability words that begin with the letter *p*.

In this example, we execute a starts with query for all capability records where the **capability** field on the Capability Word form begins with the letter *p*.

To use the starts with relational operator:

1. Click **Tailoring** > **Database Manager**.
2. Type `capability` in the **Form** field, and then click **Search**.
3. Double-click the **capability** form in the records returned.
4. Type `p` in the **Capability** field on the Capability Word form, and then press **Enter**.
5. Click any record to view the details.

Note: Results returned are case-sensitive and return only matches for the case used in your query string, unless your database is first set up for case-insensitive searching.

Create a query using a simple query expression

Applies to User Roles:

System Administrator

In this example, a query is performed on a specific assignee for all contracts in Configuration Management. Once stored, operators or groups of operators can access the query and automatically generate a list of records without reentering the detailed search criteria.

Note: You must have the capability word **QueryAdmin** or **query.stored.mod** operator record to add or edit stored queries.

To create a query using a simple query expression:

1. Click **Configuration Management** > **Contracts** > **Contracts**.
2. Use the Fill function in the Assignee field to choose an assignee. For this example, Carlton Hulman was chosen.
3. Open the More Actions menu and choose **Expert Search**.

   The Query form opens. The new query is displayed in the **Query** field. For this example, the query for Carlton Hulman is displayed as follows: `assignee#"Carlton.Hulman"`
4. On the Query form, click **Keys**.

   Note: Do not modify the automatically-generated query string, `assignee#"Carlton.Hulman"`, as it appears in the **Query** field.
5. On The Key Window (keylist form), type a key number to select the number of positions for the **assignee.name** key in the query. For example, type the number `3` to select *3* positions.
6. Click **OK**.

   The Sort Fields field in the Query Window is now modified to agree with the key definition you selected.
7. On the Query form, click **Store** to create the **querystored** file.
8. Fill in the following fields:
   - In the **Access List** field, type in the Query Groups or Operator Names that you want to have access to this query. If you want all users to have access, leave the list blank.
   - In the **Name** field, type a unique name for your query.
   - In the **Description** field, type an explanatory description.
9. Click **Add**.

The following message confirms your entry was successful: `Query added to querystored file`.

10. Continue to further refine your search criteria, or click **Search**. You can also click any record in the list of returned records to view the details.

Create a query using isin in statements

Applies to User Roles:

System Administrator

Example: Retrieve all contacts records where the Company name is either *DEFAULT* or *advantage* and the Contact Name begins with the letter *B*.

In this example, we perform a complex query on the **contacts.qbe** form using the *isin* relational operator. An Advanced Search is performed and records are returned where the value of the **Company** field is either *DEFAULT* or *advantage* and the value of the **Contact Name** field begins the letter *B*.

To create a query using isin in statements:

1. Click **System Administration** > **Base System Configuration** > **Contacts**.
2. On the Contact Information form, open the More Actions menu.
3. Select **Expert Search**.
4. In the **Query** field, type `company isin {“advantage”, “DEFAULT”} and contact.name#“B’`, and then click **Search**.
5. Click any record in the list of returned records to view the details.

Use the index function in a query

Applies to User Roles:

System Administrator

Example: Retrieve all records where the character string *DEFAULT* occurs anywhere in the device name and the device name ends in the number string *002*.

In this example, we use a like query and wildcard characters with the **index** function to perform an Advanced Search on records in the **device.qbe** form.

To use the index function in a query:

1. Click **Tailoring** > **Database Manager**.
2. Type `=device` in the **Form** field, and then click **Search**.
3. On the Asset Information form, open the More Actions menu.
4. Select **Expert Search**.
5. In the **Query** field, type `logical.name like "*002" and index("DEFAULT", logical.name) >0`, and then click **Search**.

   Note: Wildcard characters are only valid in like queries. In other types of queries, the characters are literal interpretations.
6. Click any record in the list of returned records to view the details.

Note: A *device* is also referred to as an *Asset* or *Configuration Item* (CI).

Use the lng function in a query

Applies to User Roles:

System Administrator

Example: Retrieve operator records where the length of the operator name is longer than five (5) characters.

In this example, we use the lng function to specify the length of character strings in the **operator.qbe** form and then performs an Advanced Search for records matching that criteria.

To use the lng function in a query:

1. Click **Tailoring** > **Database Manager**.
2. Type `=operator` in the **Form** field, and then click **Search**.
3. On the Operator Record form, open the More Actions menu.
4. Select **Expert Search**.
5. In the **Query** field, type `lng(name)>5`, and then click **Search**.

Note: This lng query is the same as typing `name like “??????*”`

6. Click any record in the list of returned records to view the details.

Use the tod function in a greater than or less than query

Applies to User Roles:

System Administrator

Example: Retrieve all operator records where the password was updated within the past *100* days, or where the password has not been updated in the past *100* days.

In this example, we use the greater than (>) or less than (<) operator, in conjunction with the **tod** function, to perform an Advanced Search on records in the **operator.qbe** form.

To use the tod function in a greater than or less than query:

1. Click **Tailoring** > **Database Manager**.
2. Type `=operator` in the **Form** field, and then click **Search**.
3. On the Operator Record form, open the More Actions menu.
4. Select **Expert Search**.
5. In the **Query** field, type `password.date>tod() - '100 00:00:00'`, and then click **Search**.
6. Click any record in the list of returned records to view the details.

Note: Typing `password.date<tod() - '100 00:00'` in Step 5 returns the opposite result, where the password has *not* been changed in 100 days or more.

Retrieving records by using the record list method

The record list is a standard feature for accessing records in a database. You will use it most frequently at prompts where a form or file name is required. You enter the necessary information in the form and Database Manager processes the query by looking for similar records based on the values you entered. Search values from record lists can be made more efficient through the use of relational operators. Database Manager displays the records it finds in a record list.

Retrieve all records in a table

Applies to User Roles:

System Administrator

Example: Retrieve all records in the Manage CI form.

In this example, we execute a *true* query, which returns all records contained in the **device.g** form.

To retrieve all records in a table:

1. Click **Tailoring** > **Database Manager**.
2. Type `=device` in the **Form** field, and then click **Search**.
3. On the **Manage CI** form, click **Search**.
4. Click any record in the list of returned records to view the details.

Retrieve records within a range

Applies to User Roles:

System Administrator

Example: Retrieve all device records where the value of the **Part Number** field is greater than a character value of *500* and less than a character value of *1000*.

In this example we execute a greater than (>) and less than (<) search on the *device.g* form to search part numbers where the value is greater than the number *500* and less than the number *1000*.

Note: For a range query on numeric fields, use the Query window.

To retrieve records within a range:

1. Click **Tailoring** > **Database Manager**.
2. Type `=device` in the **Form** field, and then click **Search**.
3. On the Manage CI form, type `>500 AND <1000` in the **Part Number** field.
4. Click **Find**.
5. Click any record in the list of returned records to view the details

Note: A *device* is also referred to as a *Configuration Item* (CI).

Retrieving records using the Query window method

The Query Window method allows you to retrieve records by typing a logical expression directly into the query field. When you press Enter, Database Manager searches for records that match your query and displays the list of matching records in a table. Additionally, you can define the querywindow capability word in your operator record to add the ability to specify sort fields in the Query Window application.

When should I use the Query Window instead of the record list query method?

Use the Query Window instead of the record list query method in the following situations:

- When performing simple queries in situations where a record list query cannot be used.
- For queries that include ranges, such as when the field to query exists in the database dictionary record of the file, but is not displayed on the form.
- When the input field on the form is not long enough to contain the desired value.

Excluding records from a query

To exclude specific records from a query, use the not symbol with logical or relational operators.

Using functions in a query

Use the Query window to perform queries that involve functions, such as **index**, **tod**, or **lng**.

Open the Query Window

Applies to User Roles:

System Administrator

To open the Query window from the operator record:

1. Click **Tailoring** > **Database Manager**.
2. Type `=operator` in the **Form** field, and then click **Search**.
3. On the Operator Record form, open the More Actions menu.

4. Select **Expert Search**.
5. The Query form is displayed. The display options on the form are dependent on the execution capability words assigned for the user, as follows:
   - **Query Window access** – The Query window is displayed at the bottom of your screen.
   - **Stored query access** – If stored queries are defined, a list is displayed.
   - **Both** – The Query window with buttons defining query options is displayed.

Use the Query Window

Applies to User Roles:

System Administrator

Example: Retrieve all device records where the logical name begins with the character *A*.

In this example, we use the equal to (=) relational operator to perform an Advanced Search on records in the **device.qbe** form and returns all values in the **logical.name** heading that begin with the letter *A*.

To use the Query Window:

1. Click **Tailoring** > **Database Manager**.
2. Type `=device` in the **Form** field, and then click **Search**.
3. On the Asset Information form, open the More Actions menu.
4. Select **Expert Search**.
5. In the **Query** field, type `logical.name#"A"`, and then click **Search**.
6. Click any record in the list of returned records to view the details.

Note: A *device* is also referred to as an *Asset* or *Configuration Item* (CI).

Stored queries

Stored queries enable users to retrieve and display current information efficiently by using predefined search parameters. Depending on permissions, users can run predefined queries, modify predefined queries before running them, or store a personal list of modified queries and run a query from their list.

Administrators can use stored queries to display lists of specific records or populate dynamic display objects such as charts and marquees. They can add new queries and fine tune the out-of-box queries to decrease response time or implement best practices in their environment.

Stored query option definitions

Option key	Function	Definition
F1	Append using the & character	Appends an additional query to the end of your original query using the word AND to separate the two queries.
F2	Append using the | character	Appends an additional query to the end of your original query using the word OR to separate the two queries.
F3	Insert using the & character	Inserts an additional query at the beginning of your original query using the word AND to separate the two queries.
F4	Insert using the | character	Inserts an additional query at the beginning of your original query using the word OR to separate the two queries.

Add a stored query

Applies to User Roles:

System Administrator

You can add, edit, or run stored queries from the Query window, which is accessible from the form in Database Manager or from the Application search form. Data entered into the form is automatically entered into the Query window to form the query. Depending on your permissions, you can edit and run that query, or select a query from a list of stored queries.

In this example, a query is performed on a specific assignee for all contracts in Configuration Management. Once stored, operators or groups of operators can access the query and automatically generate a list of records without reentering the detailed search criteria.

Note: You must have the capability word **QueryAdmin** or **query.stored.mod** operator record to add or edit stored queries.

To add a stored query:

1. Click **Configuration Management** > **Contracts** > **Contracts**.
2. Use the Fill function in the Assignee field to choose an assignee. For this example, Carlton Hulman was chosen.

3. Click the More Actions icon and choose **Expert Search**.

   The Query form opens. The new query is displayed in the **Query** field. For this example, the query for Carlton Hulman is displayed as follows: `assignee#"Carlton.Hulman"`

   Note: In Service Manager, you can use field name information in field help to identify the field names you want to use in your stored queries. To display field name information in field help, make sure one of the following conditions is fulfilled:

   - In the Windows client, the "Show context-sensitive help debug information" preference is enabled.
   - In the Web client, "viewcontexthelp=true" is appended to the URL before you log in.

4. On the Query form, click **Keys**.

   Note: Do not modify the automatically-generated query string, `assignee#"Carlton.Hulman"`, as it appears in the **Query** field.

5. On The Key Window (keylist form), type a key number to select the number of positions for the **assignee.name** key in the query. For example, type the number `3` to select *3* positions.

6. Click **OK**.

   The Sort Fields field in the Query Window is now modified to agree with the key definition you selected.

7. On the Query form, click **Store** to create the **querystored** file.

8. Fill in the following fields:

   - In the **Access List** field, type in the Query Groups or Operator Names that you want to have access to this query. If you want all users to have access, leave the list blank.
   - In the **Name** field, type a unique name for your query.
   - In the **Description** field, type an explanatory description.

9. Click **Add**.

   The following message confirms your entry was successful: `Query added to querystored file`.

10. Continue to further refine your search criteria, or click **Search**. You can also click any record in the list of returned records to view the details.

Update a stored query

Applies to User Roles:

System Administrator

You can add, edit, or run stored queries from the Query window, which is accessible from the form in Database Manager or from the Application search form. Data entered into the from is automatically entered into the Query window to form the query. Depending on your permissions, you can edit and run that query or select a query from a list of stored queries.

In this example, we update and save a stored query for Incident Management (the probsummary file).

To update a stored query:

1. Open the query window.
   - To open the query window from an HP Service Manager application:
     i. Open the application Search form. For this example, click **Incident Management** > **Search Incidents**.
     ii. Open the More Actions menu.
     iii. Select **Expert Search**.
   - To open the query window from a Database Manager search form:
     i. Open the form in Database Manager. For this example, type `=probsummary` in the **Form** field, and then press Enter.
     ii. Open the More Actions menu.
     iii. Select **Advanced Search**.

     The Query window opens.
2. Click **Select** to display the list of stored queries for the associated file.
3. Double-click a query to select it. For this example, select **status.resolved**.

The Stored Query record opens.

4. Click **Select** to edit the query as desired. In this example, change the query to `flag=true and problem.status<>"Resolved"`.

   Note: In Service Manager, you can use field name information in field help to identify the field names you want to use in your stored queries. To display field name information in field help, make sure one of the following conditions is fulfilled:

   - In the Windows client, the "Show context-sensitive help debug information" preference is enabled.
   - In the Web client, "viewcontexthelp=true" is appended to the URL before you log in.

5. Specify any other controls you want.
6. Click **Store** to save your edits to the selected query.

   The Stored Query Maintenance form opens.

7. Give the query a unique **Name**. For this example, `not.resolved`.
8. Add a **Description**. For this example, type `Unresolved records`.
9. Click **Add.**

   Service Manager returns you to the Query window and displays the message: *"Query added to querystored file"*.

You can click **Execute Search** to run the saved query at this point, or click **Back** to exit.

Append stored queries

Applies to User Roles:

System Administrator

You can add, edit, or run stored queries from the Query window, which is accessible from the form in Database Manager or from the Application search form. Data entered into the from is automatically entered into the Query window to form the query. Depending on your permissions, you can edit and run that query or select a query from a list of stored queries.

Note: You must have the capability word **QueryAdmin** or **query.stored.mod** operator record to add or edit stored queries.

In this example, two stored queries for Incident Management (the probsummary file) are joined and saved.

To append stored queries:

1. To open the query window from a HP Service Manager application:
   a. Open the application Search form.

      For this example, click **Incident Management** > **Search Incidents**.

   b. Open the More Actions menu.

   c. Choose **Expert Search**.

      The query window opens.

2. To open the query window from a Database Manager search form:
   a. Open the form in Database Manager.

      For this example, type **=probsummary** in the **Form** field, and then press Enter.

   b. Open the More Actions menu.

   c. Choose **Expert Search**.

      The Query window opens.

3. Click **Select** to display the list of stored queries for the associated file, and then double-click a query to select it.
   For this example, select **deadline**. The Stored Query record opens.

4. Click **Select**.
   Service Manager returns you to the list of queries.

5. Select another query from the list.
   For this example, select **pri.1**. The Stored Query record opens with additional buttons on the toolbar.

6. Click one of these buttons to use this query in addition to the one you selected previously.
   - Click **Append using &** to add this query after the previous one using the *and* (&) operator.

   - Click **Append using |** to add this query after the previous one using the *or* (|) operator.

- Click **Insert using &** to add this query before the previous one using the *and* (&) operator.
- Click **Insert using |** to add this query before the previous one using the *or* (|) operator.

For this example, click **Append using &**.

7. You can click **Execute Search** to run the query without saving it or click **Store** to save your modified query for future use.
   For this example, click **Store**. The Stored Query Maintenance form opens.

- The value in the **Name** field is that of the last appended or inserted stored query, pri.1.
- The value in the **Query** field is the new query you created. (flag=true and status#"DEADLINE" and problem.status~="Cleared" and flag=true and priority.code="1").
- For this example, enter the following information:
  - Give the query a unique **Name**. For this example, type **pri.1&deadline**.
  - Add a **Description**. For this example, type **Priority 1 records in deadline alert**.
  - Edit the **Query**, if necessary. For this example, delete the second `and flag=true`.
  - Specify any other controls you want.
- Click **Add.**

  Service Manager returns you to the Query window and displays the message: *"Query added to querystored file"*.
- You can click **Execute Search** to run the saved query at this point, or click **Back** to exit.

Run a stored query

Applies to User Roles:

System Administrator

Depending on your permissions, you can add, edit, or run stored queries from the Query window, which is accessible from the form in Database Manager or from the Application search form. Data entered into the from is automatically entered into the Query window to form the query. Depending on your permissions, you can edit and run that query, or select a query from a list of stored queries.

Example: Selecting and running a stored query.

In this example, a stored query for Incident Management (the probsummary file) is selected and run.

1. Open the query window.
   - To open the query window from a HP Service Manager application:
     i. Open the application Search form. For this example, click **Incident Management** > **Search Incidents**.
     ii. Open the More Actions menu.
     iii. Select **Expert Search**.
   - To open the query window from a Database Manager search form:
     i. Open the form in Database Manager. For this example, type `=probsummary` in the **Form** field, and then press Enter.
     ii. Open the More Actions menu.
     iii. Select **Expert Search**.

   The Query window opens.
2. Click **Select** to display the list of stored queries for the associated file.
3. Double-click a query to select it. For this example, select **status.resolved**. The Stored Query record opens.
4. Click **Select**.

   Service Manager returns you to the Query window.
5. Click **Execute Search** to run the stored query.

   Records matching the query display in a record list.
   - If only one record matches your search criteria, that record opens.
   - If no records match, the server displays the following message: *No records found*.

Cross-table join query improvements

The performance when running a join across database tables has been vastly improved as compared to version of Service Manager prior to 9.32. Previously, a query running across two tables was performed in memory by Service Manager. For data sets that contain millions of records, this process could be extremely time consuming.

The mechanism by which Service Manager executes queries across tables was changed in Service Manager 9.32. With the new mechanism, the initial Service Manager query is parsed, translated into SQL, and then issued directly to the RDBMS. The RDBMS then handles the initial join operation and returns a subset of results to Service Manager. However, because the database now performs this work, a number of limitations are imposed. For more information on these limitations, see "Limitations on cross-table join queries" on page 323

Additionally, the new functionality is limited to the Service Manager functions on the "Cross-table join functions" below.

Cross-table join functions

Under the new mechanism, you can only use the following functions in your query:

> It is highly recommended that you restrict queries to run only on indexed fields. Running these queries on non-indexed fields can result in poor performance.

lng()

The lng() function determines the length of a value specified by the parameter.

Example of the lng Function Translation

Service Manager function	Translated SQL equivalent:
`select ps.number, ps.title from probsummary ps where lng (ps.openen.by)=6`	`select t01.”NUMBER”, t01.”TITLE” from PROBSUMMARYM1 t01 where LENGTH(t01.”OPENED_ BY”) = 6`

The parameter can only be a field name or an expression that can be evaluated beforehand; It cannot be an expression that cannot be evaluated before translation into SQL:
The following cases are valid:

```
lng ( ps.category )    //In this example, “category” is a field from file “probsummary”

lng ( $L.var )    //In this example, $L.var is a local variable defined beforehand.
```

The following cases are invalid:

```
lng ( ps.title + ps.brief )

lng ( ps.title + $L.var )
```

null()

The Null() function determines whether a value specified by the parameter is null or not.

Example of the NULL Function Translation

Service Manager function	Translated SQL equivalent:
`select ps.number, ps.title from probsummary ps where null (ps.assignment)`	`select t01.”NUMBER”, t01.”TITLE” from PROBSUMMARYM1 t01 where t01.”ASSIGNMENT” is NULL`

The parameter can only be a field name or an expression that can be evaluated beforehand; It cannot be an expression that cannot be evaluated before translation into SQL:
The following cases are valid:

```
null( ps.category )    //In this example, “category” is a field from “probsummary” file.
```

```
null( $L.var )    //In this example, $L.var is a local variable defined beforehand.
```

The following cases are invalid:

```
null ( ps.title + ps.brief )
```

```
null ( ps.title + $L.var )
```

nullsub()

The nullsub() function substitutes the variable with a default value if the variable specified by the first parameter is null:

Example of the nullsub Function Translation

Service Manager function	Translated SQL equivalent:
`select ps.number, ps.title from probsummary ps where nullsub (ps.openen.by, “falcon”)=”falcon”`	`select t01.”NUMBER”, t01.”TITLE” from PROBSUMMARYM1 t01 where COALESCE (t01.”OPENED_BY”, ‘falcon’) = ‘falcon’`

The first parameter can only be a field name or an expression that can be evaluated beforehand, it cannot be an expression that cannot be evaluated before translation into SQL. The other parameters can only be constant numbers or expressions that can be evaluated to numbers before translation into SQL:
The following cases are valid:

```
nullsub ( ps.category, “abc” )   //In this example, “category” is a field from file “probsummary”
```

```
nullsub ( $L.var, “abc” )  //In this example, $L.var is a local variable defined beforehand.
```

The following cases are invalid:

```
nullsub ( ps.title + ps.brief, “abc” )
```

```
nullsub ( ps.title + $L.var, “abc” )
```

operator()

When you use the operator() function in part of a query, the Service Manager server will replace the call to operator() with the appropriate value. Therefore, the following query is valid:

```
select ps.number, ps.title from probsummary ps where ps.opened.by = operator()
```

substr()

The substr() function extracts a substring from a value specified by the parameter.

Example of the lng Function Translation

Service Manager function	Translated SQL equivalent:
`select ps.number, ps.title from probsummary ps where substr (ps.openen.by, 0, 3)=”fal”`	`select t01.”NUMBER”, t01.”TITLE” from PROBSUMMARYM1 t01 where SUBSTR (t01.”OPENED_BY”) = ‘fal’`

The first parameter can only be a field name or an expression that can be evaluated beforehand, it cannot be an expression that cannot be evaluated before translation into SQL. The other parameters can only be constant numbers or expressions that can be evaluated to numbers before translation into SQL:

The following cases are valid:

```
substr ( ps.category, 0, 4 )   //In this example, “category” is a field from file “probsummary”
```

```
substr ( $L.var, 0, 4 )   //In this example,$L.var is a local variable defined beforehand.
```

The following cases are invalid:

```
substr ( ps.title + ps.brief, 0, 4 )
```

```
substr ( ps.title + $L.var, 0, 4 )
```

tod()

When you use the tod() function in part of a query, the Service Manager server will replace the call to tod() with the appropriate value. Therefore, the following query is valid:

```
select ps.number, ps.title from probsummary ps where ps.open.date<tod()
```

toupper() and tolower()

The totoupper() and tolower() functions convert the given parameter to upper or lower case as appropriate.

Example of the toupper() Function Translation

Service Manager function	Translated SQL equivalent:
`select ps.number, ps.title from probsummary ps where toupper (ps.openen.by)="FALCON"`	`select t01."NUMBER", t01."TITLE" from PROBSUMMARYM1 t01 where UPPER(t01."OPENED_ BY") = 'FALCON'`

The parameter can only be a field name or an expression that can be evaluated beforehand; It cannot be an expression that cannot be evaluated before translation into SQL:

The following cases are valid:

```
toupper ( ps.category )   //where "category" is a field from file "probsummary"
tolower ( $L.var )    //where $L.var is a local variable defined beforehand.
```

The following cases are invalid:

```
tolower ( ps.title + ps.brief )
toupper ( ps.title + $L.var )
```

Cross-table join query examples and use cases

The following examples detail various scenarios in which you will see the benefits of the cross-table join queries:

Note: The table alias are mandatory for cross-table query. For example, `select pro.* from probsummary pro` is correct, whereas `select probsummary.* from probsummary` is incorrect.

Reserved words

Reserved words have a special meaning in Service Manager that is defined in RAD. These special words can only be used for that purpose. Do not use them for any other purpose such as field names or table alias. The following is an alphabetical list of reserved words:

AND, BEGIN, DO, ELSE, END, FALSE, FOR, IF, IN, ISIN, NOT, NULL, OR, STEP, WHILE, THEN, TRUE, UNKNOWN, INNER, OUTER, JOIN, ON, SELECT, AS, FROM, WHERE, ASC, DESC, BETWEEN, GROUP, BY, HAVING, LIKE, TABLE, VIEW, WHERE, MIN, MAX, AVG, COUNT, SUM, ID.

In addition, all the RAD and JavaScript function names, and standard SQL reserved word in Oracle, SQL Server, or DB2 are similarly reserved.

Use case: Incident Analyst

As an Incident Analyst, I would like to show interactions that have an incident attached that is in open status.

Task 1. Configure the relationship between interaction and incident.

1. Log on as a system administrator.
2. From the command line, enter `db` and then press the **Execute Command** button.
3. In **Form** enter `relatedObjectMap` and then click the **Search** button.
4. Type the following values in the corresponding fields and then click the **Search** button .

Field	Value
Source Table:	`incidents`
Source Field:	`incident.id`

5. Click **Delete** to delete the relation between `incident` and `incdepends`, which is not used in the Inbox.
6. Type the following values in the corresponding fields and then click the **Add** button.

Field	Value
Source Table:	incidents
Source Field:	incident.id
To Table:	screlation
To Field:	source

7. Type the following values in the corresponding fields and then click the **Add** button.

Field	Value
Source Table:	screlation
Source Field:	depend
To Table:	probsummary
To Field:	number

8. Log out of Service Manager.

Task 2: Create the view

1. Log in as a system administrator.
2. Switch to the Interaction queue.
3. Click **More** or the More Actions menu, and then click **Define Views**.
4. Click **New**.
5. In the name field, type the following, choose **System View**, and the click **Next**:

   All interactions that have an open incident attached

6. Click **Next**.
7. Click **Filter**, click **Advanced Filter**, and then click **Add New Filter Criteria**.
8. In **Field in Interaction**, select Interaction ID, and then click **Use Multi Level Field Chooser**.
9. Select field **Depend** for screlation, and select **Status** for Incident.
10. Click **Next**.

11. Set **Comparison** to `Equals`, set **Value** to `Open`, and then click **Next**.
12. Click **OK**.
13. Click **Fields**.
14. Specify the following values in the corresponding fields and then click **Add to List**.

Field	Value
Interaction	Interaction ID
screlation	Depend
Incident	Status

15. Specify the following values in the corresponding fields and then click **Add to List**.

Field	Value
Interaction	Interaction ID
screlation	Depend

16. Click **Finish** and then click **Apply View**.

After you complete these tasks, you can log in as an Incident Analyst, and then switch to the "All interactions that have an open incident attached" view.

Use case: Department manager

As a department manager, I would like to see all incidents that were closed by contacts from my department in last 30 days.

Task 1: Create the view

1. Log in as a system administrator.
2. Switch to the Incident queue.
3. Click **More** or the **More actions** menu, and then click **Define Views**.
4. Click **New**.

5. In the name field, type the following, choose **System View**, and the click **Next**:

   `All incidents closed by contacts from my department in last 30 days`

6. Click **Next**.
7. Click **Filter**, click **Advanced Filter**, and then click **Add New Filter Criteria**.
8. In **Field in Incident**, select `Closed By`, and then click **Use Multi Level Field Chooser**.
9. Select field **Contact Name** for `Operator`, and then select **Dept Name** for `Contact`.
10. Click **Next**.
11. Set **Comparison** to `Equals`, set **Value** to `North America - IT`, and then click **Next**.
12. Click **OK**.
13. Click **Fields**
14. Specify the following values in the corresponding fields and then click **Add to List**.

Field	Value
Incident	Close Time
Incident	Closed By

15. Click **Finish** and then click **Apply View**.

Task 2: Define the restriction in the inbox definition

1. Log in as a system administrator.
2. From the command line, enter `db` and then press the **Execute Command** button.
3. In **Form** enter `inbox` and then click the **Search** button.
4. Click **inbox.view**, click **Search**, and then click **All incidents closed by contacts from my department in last 30 days**.
5. Click **Query definition** and modify the query form as follows:

   From:

```
probsummary;closed.by;operator;name;operator;contact.name;contacts;contact.name
;contacts;dept.name="North America - IT"
```

To:

```
probsummary;closed.by;operator;name;operator;contact.name;contacts;contact.name
;contacts;dept.name=$lo.dept and close.time>tod() - '30 00:00:00'
```

6. Click **Save**.

After you complete these tasks, you can log in as an Incident Manager, and then switch to the “All incidents closed by contacts from my department in last 30 days” view.

Use case: JavaScript Developer

As a developer, I’d like to count the attachments related to interactions related to an incident.

1. Log in as falcon,
2. Navigate to **Tailoring**> **Script Library**
3. Type `getAttachments` in the **Name** field.
4. Click **Add**.
5. Copy and paste the following code into the code window:

```
var start=new Date();
var file = new SCFile("probsummary");
var sql = "SELECT pro.*, scr.*, att.* FROM probsummary pro, screlation scr
inner join SYSATTACHMENTS att on (scr.source=att.topic ) where
scr.depend=pro.number and scr.depend=\"IM10002\" and att.segment=0";
var success=file.doSelect( sql ) ;


var findNextRecord=RC_SUCCESS;
var count=0;
//iterate to print the related attachment names
while ( success== RC_SUCCESS && findNextRecord == RC_SUCCESS && count<120)
{
    print("find attachment No.: " + count + " " + file.filename + " in " +
file.topic);
    //print(system.functions.descriptor(file));
    findNextRecord=file.getNext();
    count++;
```

```
 }

var end=new Date();
var timespent = end-start;
print ("select time is: " + timespent + "ms");
start=new Date();

//count how many attachments are related to this incidents
count=file.doCount(sql);
print ("There are " + count + " attachments");
end=new Date();
timespent = end-start;
print ("count time is: " + timespent + "ms");
```

6. Save and compile the JavaScript code.
7. Execute the compiled code.

Example: Query two tables

This query finds all open incidents for which your department is the service owner of the affected service:

```
var file = new SCFile("probsummary");
var sql = "select p.* from probsummary p LEFT OUTER JOIN device d ON
(p.affected.item=d.logical.name) where d.owner=$lo.dept and p.flag=true";
var success = file.doSelect(sql);

while (success == RC_SUCCESS)
{
      print(file.affected_item)
      success = file.getNext();
}
```

Example: Query more than two files

This query finds all open incidents that are assigned to my department:

```
var file = new SCFile("probsummary");
var sql = "SELECT ta01.*, ta03.dept.name AS dept.name FROM probsummary ta01 LEFT
OUTER JOIN operator ta02 LEFT OUTER JOIN contacts ta03 ON  ( ta02.contact.name =
ta03.contact.name )  ON  ( ta01.assignee.name = ta02.name )  WHERE ( ta01.flag =
true and ta03.dept.name=$lo.dept)";
var success = file.doSelect( sql ) ;
while (success == RC_SUCCESS)
```

```
{
      success = file.getNext();

}
```

Example: Sort

This query finds all open incidents for which your department is the service owner of the affected service, and sorts by service name in ascending order and incident number in descending order:

Note: If possible, use the `order by` clause in the query instead of `setsort()`, because the `order by` clause sort fields before the `select` operation. `setsort()` requires another SQL operation to sort. `setOrderBy` is not supported in cross table query.

```
var file = new SCFile("probsummary");
var sql = "select p.*, d.logical.name as lname from probsummary p LEFT OUTER JOIN
device d ON (p.affected.item=d.logical.name) where d.owner=$lo.dept and
p.flag=true";
sql += " order by d.logical.name asc, p.number desc"
var success = file.doSelect(sql);
while (success == RC_SUCCESS)
{
         findNextRecord = file.getNext();
}
```

Example: Count

This query counts all open incidents for which your department is the service owner of the affected service:

Note: For better performance, this example use `doCount` in the query instead of looping on the record list.

```
var file = new SCFile("probsummary");
var sql = "select p.* from probsummary p LEFT OUTER JOIN device d ON
(p.affected.item=d.logical.name) where d.owner=$lo.dept and p.flag=true";
var count = file.doCount(sql);
print(count);
```

Example: Handle two files that have the same field name

This query finds all open incidents for which your department is the service owner of the affected service, and then prints the category of the incidents and the category of the service.

Note: If two fields have same name, you must use an alias to distinguish the fields.

```
var file = new SCFile("probsummary");
var sql = "select p.*,d.category AS cate from probsummary p LEFT OUTER JOIN device
d ON (p.affected.item=d.logical.name) where d.owner=$lo.dept and p.flag=true";
var success = file.doSelect(sql);

print(file.category);
print(file.cate);
```

Example: Handle BLOB, CLOB, text and images

This query shows all open Incidents, where current operator is a member of an assignment group (Support Groups) that is defined in the affected service.

Note:

- Because an RDBMS cannot search on a BLOB field directly, if you want to use BLOB field as a query condition, you must map the BLOB field as multi-row array.
- You must map `support.groups` from `DEVICE2M1` to `DEVICE2A1` for this example to work. Otherwise, you will receive the following error:

  ```
  Queries on Blob/Clob/Text/Image field is not supported in cross table query.
  (display,process.option) Field "ta02.support.groups" in the query is not
  mapped or mapped to Blob/Clob/Text/Image. Queries on Blob/Clob/Text/Image
  field is not supported in cross table query. (display,process.option)
  ```

```
var file = new SCFile("probsummary");
var sql = "SELECT ta01.*, ta02.support.groups FROM probsummary ta01 LEFT OUTER JOIN
device ta02 ON  ( ta01.affected.item = ta02.logical.name )  WHERE (
ta02.support.groups isin $lo.pm.assignments and ta01.flag=true )";
var success = file.doSelect(sql);
while (success == RC_SUCCESS)
{
    success = findNextRecord = file.getNext();
}
```

Example: Handle a one-to-many relationship

This query shows all Incidents that have an associated open interaction relation.

Note: All previous examples show a many-to-one relationship; that is, multiple rows in the first table are associated to the same row in another table. For example, multiple Incidents have the same owner. In this case, the query returns only one row for each row in the Incident table. In this example, the query returns more than one row for each row in the first table. Therefore, the results of this query may resemble the following:

Incident	Interaction
IM10003	SD10060
IM10003	SD10033
IM10003	SD10003
IM10002	SD10006
IM10002	SD10002

```
var file = new SCFile("probsummary");
var sql = "SELECT ta01.*, ta03.incident.id AS iid FROM probsummary ta01 LEFT OUTER
JOIN screlation ta02 LEFT OUTER JOIN incidents ta03 ON  ( ta02.depend =
ta03.incident.id )  ON  ( ta01.number = ta02.source )  WHERE ( ta03.open#\"Open\" )
";
var success = file.doSelect(sql);
while (success == RC_SUCCESS)
{
    print(file.number + " " + file.iid);
    success = file.getNext();
}
```

Limitations on cross-table join queries

The new implementation of the Cross Table Join mechanism has the following limitations and behavior changes:

- You can only run a query on the eight functions that are described in "Cross-table join functions" on page 311
- The `DISTINCT` clause is not supported.

- You cannot run a join operation on the following:
  - BLOB or CLOB objects (This is a limitation common to most RDBMSs)
  - Image data-types
  - Text data-types
  - Encrypted fields
  - LDAP files
  - Joined files
  - Merged files
  - Tables in different RDBMSs
- Queries on IR files now behave differently. Specifically, the following behaviors now occur:
  - Full text search no longer stems. The revised mechanism uses an SQL `LIKE` clause to retrieve results. Therefore, a search based on `LIKE "%printer%"` will no longer return stemmed variants such as "printing" or "printed." In the previous implementation, stemming was included because the query was run entirely in Service Manager and leveraged the stemming dictionary.
  - You can no longer perform a full text search in the Cross Table Query in SM. Full-Text Search can only be performed in the initial relational search.
  - Results are returned only from the exact fields\ that you specify. For example, if you searched for the term "printer" on the "Title" field by using the previous mechanism, you would receive a hit even if the term was not mentioned in the "Title" field, but only in the associated "Desc" field. With the new mechanism, you will only receive a hit if the term appears in the "Title" field.
- You can no longer `GROUP` or `SORT` by BLOB objects.

Error Messages

To support the new cross table join mechanism, the following error messages have been implemented:

```
Unable to locate field %s in any file

The %S function is not supported in cross table queries.

Queries on Blob/Clob/Text/Image field %S is not supported in cross table query.

Cannot locate field %s in join condition \"%.*s\"
```

```
The join condition \"%.*s\" cannot be converted into an SQL expression. One or more fields might be a Blob/Clob/Text/Image field.

The file alias \"%.*s\" is specified in the "order by" clause, but the definition cannot be found.

The field \"%.*s\" in the "order by" clause cannot be found.

Cannot locate field %s.

Cannot sort or group on array field %s.

Joining tables of derived file type like JOIN_FILE or MERGED_FILE is not supported.

Joining tables from different RDBMS is not supported.

Cannot locate field %s in SELECT list.
```

Database performance tuning

The topics in this section list suggestions for tuning your database to improve performance, such as query speed and memory usage. Click any of the topics in this section to learn more. For additional information, you can search the knowledgebase in Knowledge Management or visit the HP Software Customer Support Web site for a variety of best practice documents and published white papers.

Tuning: Improving query speed

Most performance degradation occurs during queries. An inefficient query can adversely affect the performance of your system by allocating resources that are not always needed. Queries can be tuned to perform more efficiently so that the engine is not required to search entire files to determine which records match the query. Other factors that can affect the performance of file system searches include key design, key selection algorithms, and number of records.

Tuning: Number of fields in files

If you keep the number of fields in a database table low and limit the number of records, performance is improved in the following areas:

- Searching for fields in a dbdict.
- Compressing and writing records to disk.
- Decompressing and reading records from disk.

- Operating client and server communications.
- Exercising the Load and Unload options.

Troubleshooting and tuning queries

This section offers information about troubleshooting and tuning queries. The suggestions, though not required, are derived from best practices learned from previous applications of HP Service Manager in large working environments. For additional information and suggestions, you can search the knowledgebase in Knowledge Management or visit the HP Software support web site for a variety of documents and published white papers.

Tuning: Fully keyed queries

In fully keyed queries, all referenced fields are defined in the correct order using a single key to process the query. This is an efficient query type because the query engine determines precisely which records match the query by reading only part of the index tree and none of the data record.

True queries

A true query is a query that only uses the *true* expression. The query engine uses the first Unique key defined in the key list to process the query and then returns all records in the database. If no Unique key is defined, HP Service Manager uses the first key. Although a true query quickly and efficiently returns results, the results include all records in the table and do not specify only records you want to view.

Partially keyed queries

A partially keyed query is a query where some of the referenced fields are defined in the key that processes the query. Using the index, the query engine can only partially determine whether a record matches the query. For an efficient partially keyed query, the first field in the key must be included in the query expression. If it is not included, all index entries in the index must first be read to determine the results of the query.

Nonkeyed queries

Non keyed queries are typically used for small files only and are not efficient when used against a large file. The query engine must first read all index entries in an index and then read all records within those

entries to determine which records match the query.

Running the database maintenance utility

There are several ways to run the database maintenance utility. Pick the one that best suits your needs.

- "Run the database maintenance utility in foreground (interactively) on Unix" below
- "Run the database maintenance utility in background (batch) on Unix" on the next page
- "Run the database maintenance utility in foreground (interactively) on Windows" on the next page
- "Run the database maintenance utility in the background (batch) on Windows" on page 329

Run the database maintenance utility in foreground (interactively) on Unix

Applies to User Roles:

System Administrator

By default, the HP Service Manager database utility executable is located in the \ Service Manager\Server\RUN directory.

To run the database maintenance utility interactively on Unix:

1. Open a Unix Command prompt.
2. Navigate to the Service Manager\Server\RUN directory.
3. To have all input and output captured in a file, issue the following Unix command:

   ```
   script scriptname.out
   ```

4. Issue the following command to start the Service Manager database utility:

   ```
   sm -util
   ```

5. The script ends when the forked shell exits. (Use control-D to exit the Bourne shell (sh(1)), and exit, logout or control-d (if ignoreeof is not set) for the C-shell, csh(1)).
6. Check the output. If you receive error messages, contact HP Software Customer Support for assistance.

Run the database maintenance utility in background (batch) on Unix

Applies to User Roles:

System Administrator

To start the HP Service Manager database utility in the background, you must first create an input file that contains the commands that are normally entered using the keyboard.

To run the database maintenance utility in the background (batch) on Unix:

1. Shut down the Service Manager server. (This will terminate the clients as well.)
2. Open a Unix Command prompt.
3. Navigate to the \ Service Manager\Server\RUN directory and type the following command: `sm -util <filename.in >filename.out`
   Where the *filename.in* variable is the name of the prepared input file and the *filename.out* variable is the name of the file containing the output.
4. Check the output. If you receive error messages, contact Customer Support for assistance.

Run the database maintenance utility in foreground (interactively) on Windows

Applies to User Roles:

System Administrator

By default, the HP Service Manager database utility executable is located in the \ Service Manager\Server\RUN directory.

To run the database maintenance utility interactively on Windows:

1. Open a Windows Command prompt or the Window RUN dialog box.
2. Go to the Service Manager\Server\RUN directory and issue the following command: `sm -util`
   For example:

```
C:\Program Files\HP\ Service Manager\Server\RUN> sm -util
```
Your screen displays the Service Manager database utility main menu.

3. Select an option to proceed.
4. Check the output. If you receive error messages, contact Customer Support for assistance.

Run the database maintenance utility in the background (batch) on Windows

Applies to User Roles:

System Administrator

To start the database maintenance utility in the background, you must first create an input file that contains the commands that are normally entered using the keyboard.

By default, the database maintenance utility executable is located in the \ Service Manager\Server\RUN directory.

To run the database maintenance utility in background (batch) on Windows:

1. Prepare the input file for the utilities you want to run.
2. Shut down the Service Manager server. This will also terminate any active client sessions.
3. Open a DOS Command prompt or the Windows Run dialog box.
4. Navigate to the \ Service Manager\Server\RUN directory and type the following command:`sm -util <filename.in >filename.out`
Where the *filename.in* variable is the name of the input file prepared and the *filename.out* variable is the name of the file containing the output.
5. Check the output. If you receive error messages, contact Customer Support for assistance.

Entity relationship management

HP Service Manager defines relationships between files and fields, which are constraints that should be enforced for relevant data in your database. However, your database sometimes may contain invalid data that no longer obeys the constraints. For example, you have renamed or deleted an operator record in your database, while some existing records are still referencing the old operator name. You may want to find out these invalid records and clean them up as needed. Additionally, you may want to get a clear picture of all entity relationships in your database and identify the impact of potential data changes.

Service Manager provides three utilities that can help you manage your entity relationships , run data reference check reports for selected files, as well as generate entity relationship diagrams:

- "Entity Relationship Diagram utility" below
- "Missing Reference Report utility" on page 340
- "Relationship Manager" on page 359

Entity Relationship Diagram utility

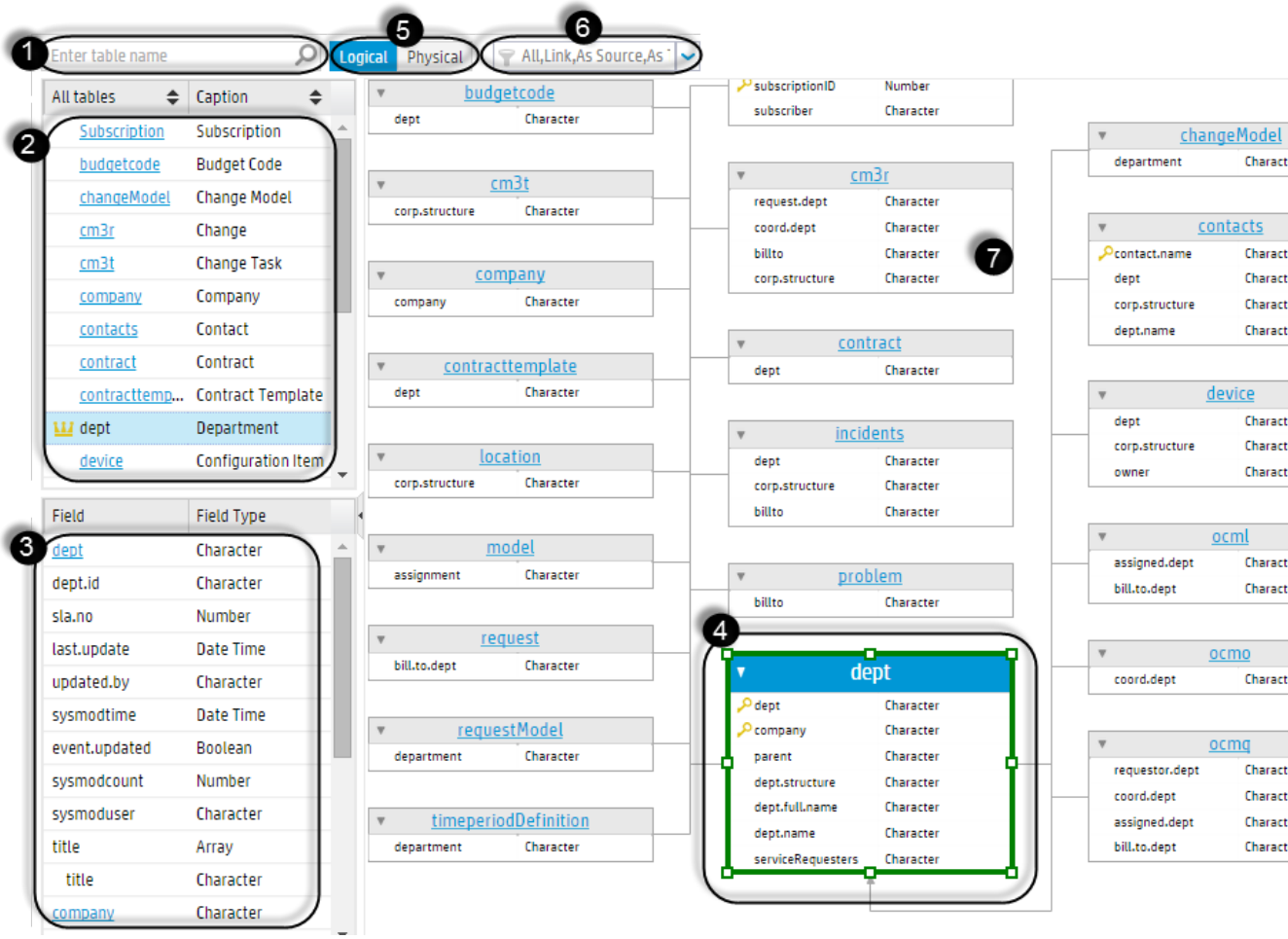
The Entity Relationship Diagram utility presents entity relationships for selected files and fields in your database in interactive diagrams (called Entity Relationship Diagrams or ERDs) so that you can have a clear picture of your entity relationships and identify the impact of potential data changes. Additionally, the utility can export the diagrams to static PDF format for offline use.

The utility processes only entity relationships that have a Definition Type of **link**, **erddef**, or **manual**.

- This functionality is available only from the web client.
- To use this utility, you must have the SysAdmin capability word.
- Screen readers are not supported for entity relationship diagrams generated by this utility.
- The accuracy of relationships in the diagrams depends on the correctness of the relevant **link** and **erddef** records.

Entity Relationship Diagram interface

The following figure shows the Entity Relationship Diagram interface.



The following table explains different parts of the interface.

Number	Name	Description
1	Table name input box	Enter the name of a table for which you want to generate an ERD, and then click the magnifying glass icon. For details, see "Generate an entity relationship diagram for a file" on the next page.
2	Table list	This is a list of tables, including the current table and all its related tables. The current table is identified by a crown icon before its name (for example, **dept**).
3	Field or relationship list	Initially, this area lists all fields of the current table. However, if you select a field of the current table in area 4, this area switches to a list of relationship records of the selected field.
4	Current table and its fields	In the ERD, the current table is highlighted with a green border. Felds that are the unique key of a table have a key icon before their names.
5	View tabs	These two tables allow you to switch the diagram between two views:

Number	Name	Description
		• **Logical**: This view displays the tables and fields using their names defined in the DBDICT records. • **Physical**: This view displays the tables and fields using their SQL Tables (for tables) and SQL Names (for fields) defined in the DBDICT records. By default, the Logical view is displayed.
6	Filter box	This box enables you to select filters to further filter information in the diagram. • **All**: Shows all of the relationships between the current table and other tables. By default, all diagrams are displayed using this filter. • **Link** > **As Source**: Shows only those relationships between the current table and other tables where the Definition Type is **link** and the Source Table is the current table. • **Link** > **As Target**: Shows only those relationships between the current table and other tables where the Definition Type is **link** and the Target Table is the current table. • **ERDDef**: Shows only those relationships between the current table and other tables where the Definition Type is **erddef**. • **Manual**: Shows only those relationships between the current table and other tables where the Definition Type is **manual**.
7	Related tables and fields	This area displays all related tables and fields of the current table.

Dynamic display

When you select another table or field in the diagram, the diagram is reloaded automatically. For details, see "View relationships of a specific field in an entity relationship diagram" on page 336.

Toolbar

The Entity Relationship Diagram interface also provides a toolbar, which allows you to interact with the diagram using the tools on it. For more information, see "Interact with an entity relationship diagram" on page 335.

Generate an entity relationship diagram for a file

Applies to User Roles:

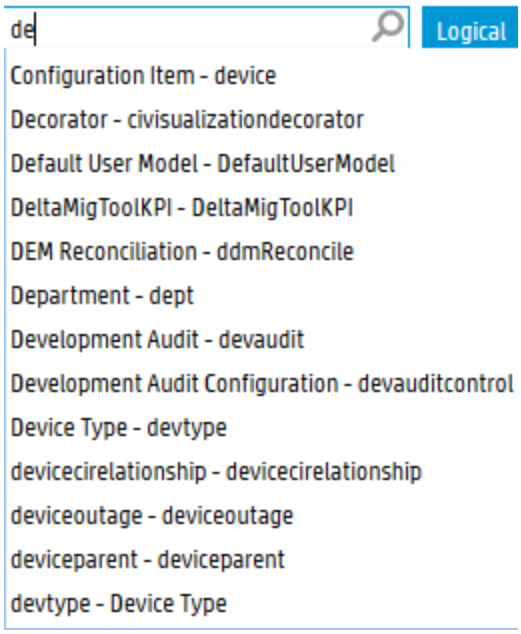
System Administrator

To generate an entity relationship diagram (ERD) for a file, follow these steps:

1. Navigate to **Tailoring** > **SQL Utilities** > **Entity Relationship Diagram** to open the Entity Relationship Diagram utility.

2. In the Table Name box at the top left corner, type the name of the file. For example, **dept**.

   Auto Complete is available for table names. When you are typing, suggested items are automatically listed. The list items are in the following format and are sorted in alphabetical order by caption: *<caption>* - *<table name>*. Instead of typing the full table name, you can select the name from the list.



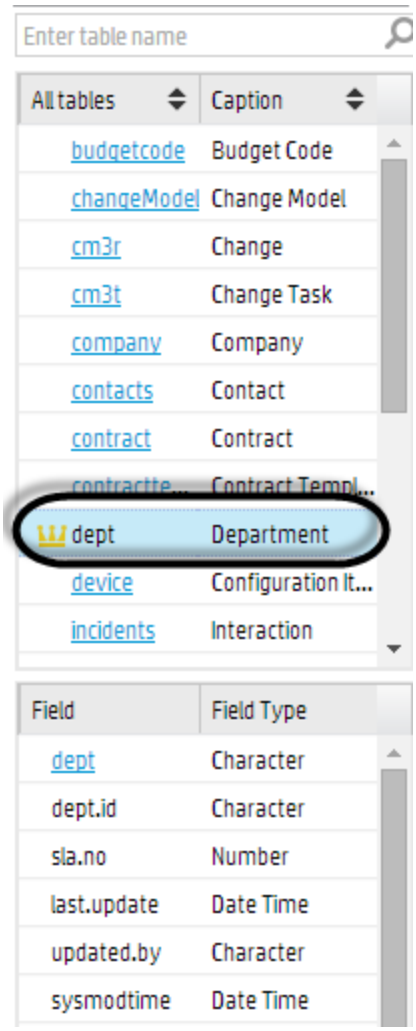
3. Press Enter, or click **Search** (the magnifying glass icon).

   The Entity Relationship Diagram utility creates a list of related tables and fields on the left pane, and generates a dynamic ERD that shows all relationships between the current file (**dept** in this example) and other files.

4. View the list of tables and fields on the left pane.

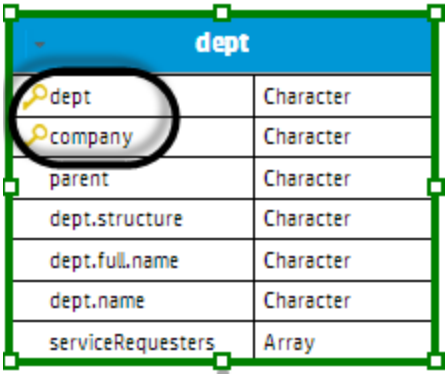
   In the upper area, a list of tables that have a relationship with the current file (**dept** in this example) is displayed along with their captions; in the lower area, a list of fields of the current file that have a relationship with other tables is displayed along with their field types.

Tip: The current file is identified by a crown icon before its name. When you select a file in the list, a crown icon appears before the name of that file. You can use the up/down arrows to sort the table list in ascending or descending order by table name or by table caption.



5. View the ERD in the right pane.

   In the ERD, the current table is highlighted with a green border, and fields that are the unique key of a table have a key icon before their names.



Next, you can work with the ERD as described in the following tasks:

"Interact with an entity relationship diagram" below

"View relationships of a specific field in an entity relationship diagram" on the next page

Interact with an entity relationship diagram

Applies to User Roles:

System Administrator

The Entity Relationship Diagram (ERD) utility generates dynamic diagrams, which allow numerous user interactions.

1. Generate an ERD. For details, see "Generate an entity relationship diagram for a file" on page 332.
2. You can interact with the table in the following ways.

Interaction	Description
Filter the diagram	Use the filters from the box.
View field specific relationships	For details, see "View relationships of a specific field in an entity relationship diagram" on the next page.
Drag and drop the diagram	You can drag and then drop the diagram to a desired place on the right pane.
Collapse all fields	By default, an ERD is displayed with all field names expanded under the table names. You can collapse the fields to get a clearer picture of the relationships

Interaction	Description
	between the tables. To collapse all fields in an ERD, click the **Collapse All** icon at the top right of the diagram pane. **Tip:** Once the fields are collapsed, you can click the **Expand All** icon to expand them.
Show a thumbnail	You can show a thumbnail of an ERD. To do so, click the **Show Thumbnail** icon at the top right of the diagram pane. A thumbnail appears at the upper right of the diagram pane. **Tip:** Once a thumbnail is displayed, you can click the **Hide Thumbnail** icon to hide it.
Zoom in/out	You can zoom an ERD in or out by dragging the zoom slider to the right or left or by clicking the plus (+) or minus (-) icon at the top right of the diagram pane.
Fit the diagram to the page	You can click the **Fit** icon at the upper right of the diagram pane to fit the diagram to the page.
Export the diagram to a PDF	You can export the diagram to a PDF file by clicking the **Export** icon at the top right of the diagram pane.

View relationships of a specific field in an entity relationship diagram

Applies to User Roles:

System Administrator

Before you make updates to a specific field in a table or to its values in the database, you may want to see the relationships of this specific field to understand the potential impact on other records.

To view the relationships of a specific field, follow these steps:

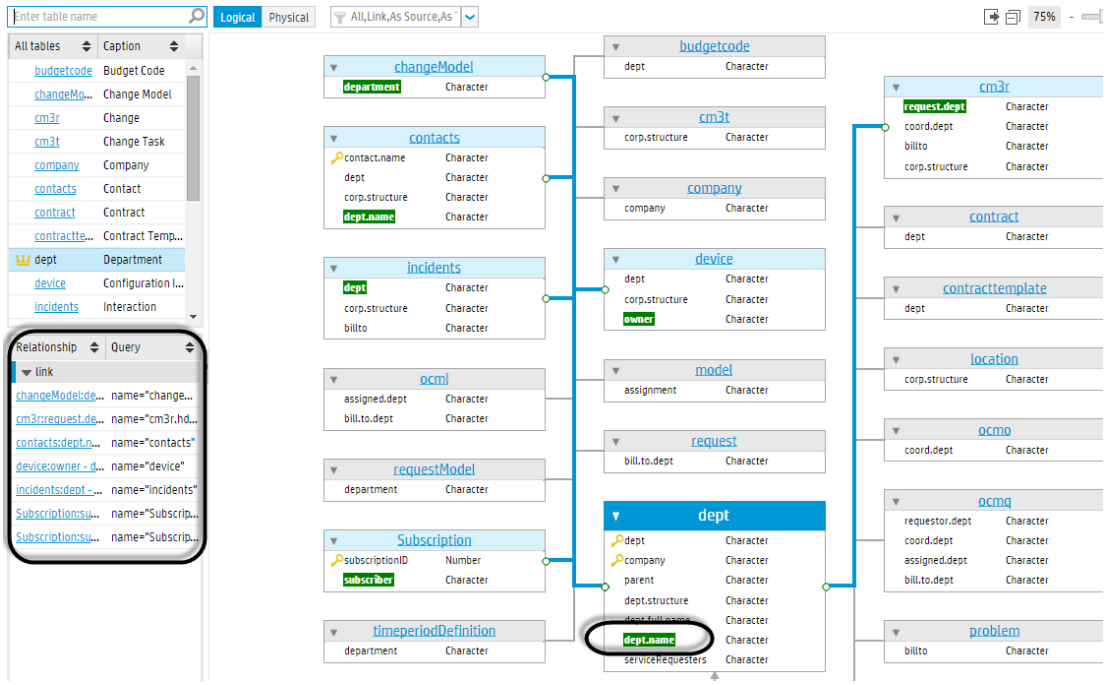
1. Navigate to **Tailoring** > **SQL Utities** > **Entity Relationship Diagram**.
2. In the Table Name box, type the name of the table that contains the field.

3. Click **Search**.

   Service Manager generates an ERD for the specified table.

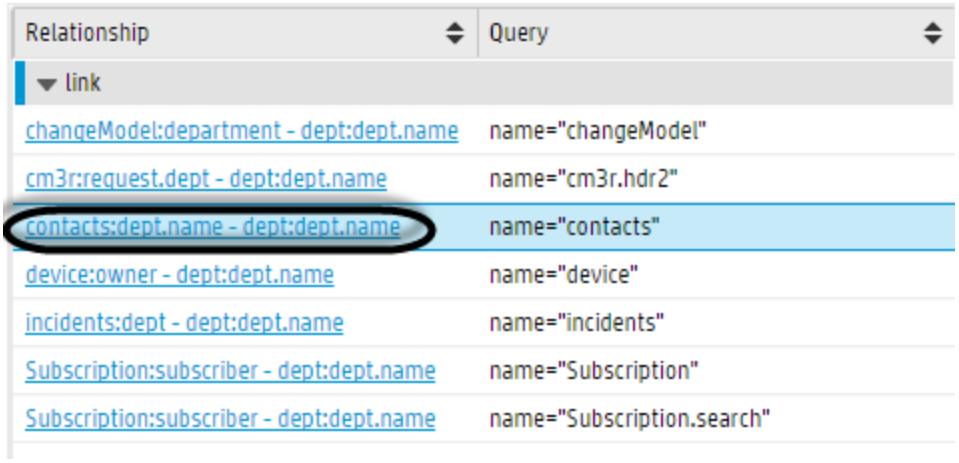
4. In the diagram, select a field in the current table.

   The selected field and its related fields in other tables are highlighted in green, and the relationships between them are highlighted with blue connector lines. The following figure shows an example in which the relationships between the **dept.name** field in the **dept** table and related fields in other tables are highlighted. The related fields are highlighted with a green background, and the relationships are highlighted with blue connector lines.



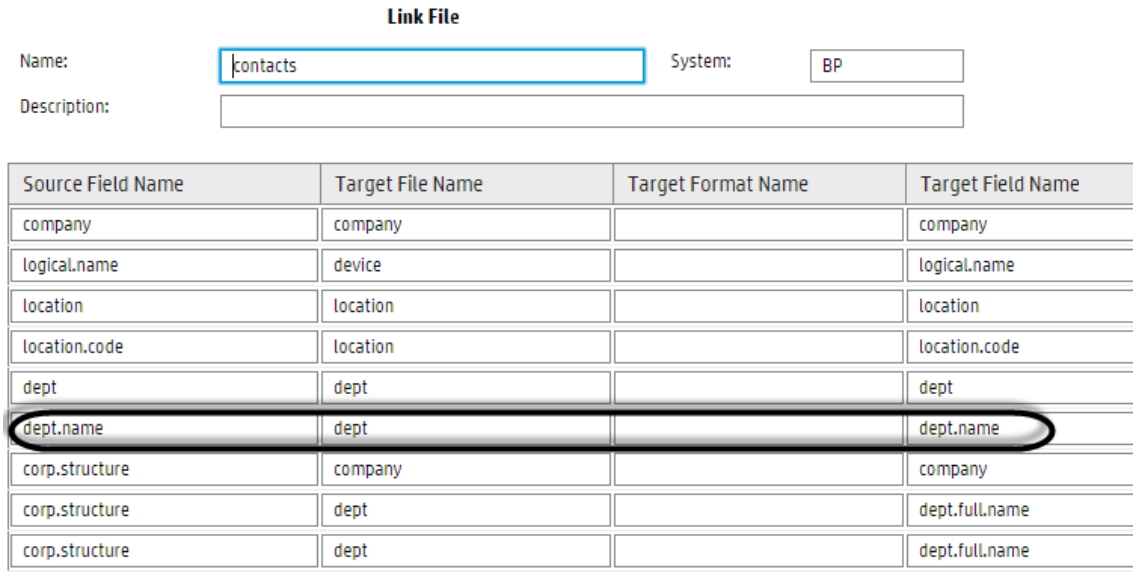
In the meantime, the lower part of the left pane is refreshed to display the relationship definitions for the current field, as well as their queries. The relationship definitions are displayed as links, which you can directly open. See the following figure for an example in which the relationship definitions are all **link** files.

The display text of each relationship definition link uses the following format: *<source table name>:<source field alias>-<target table name>:<target field alias>*, where the aliases are the Source Field Name and Target Field Name that are defined in the relationship definition. The following is an example: **contacts:dept.name - dept:depart.name**.



Relationship	Query
link	
changeModel:department - dept:dept.name	name="changeModel"
cm3r:request.dept - dept:dept.name	name="cm3r.hdr2"
contacts:dept.name - dept:dept.name	name="contacts"
device:owner - dept:dept.name	name="device"
incidents:dept - dept:dept.name	name="incidents"
Subscription:subscriber - dept:dept.name	name="Subscription"
Subscription:subscriber - dept:dept.name	name="Subscription.search"

If you click the example link, the **link** record is displayed, in which you can see the source and target table names and field names that are displayed in the link text above.



Link File

Name: contacts System: BP

Description:

Source Field Name	Target File Name	Target Format Name	Target Field Name
company	company		company
logical.name	device		logical.name
location	location		location
location.code	location		location.code
dept	dept		dept
dept.name	dept		dept.name
corp.structure	company		company
corp.structure	dept		dept.full.name
corp.structure	dept		dept.full.name

Identify misconfigured reference fields

Applies to User Roles:

System Administrator

A reference field retrieves data from another table instead of the table to which it belongs. An example reference field is the **affected.item** field in the **probsummary** table, which retrieves data from the **device** table. A reference field must have the correct referenced table configured in the datadict

record. The Entity Relationship Diagram utility can help you identify reference fields whose referenced table is not configured correctly.

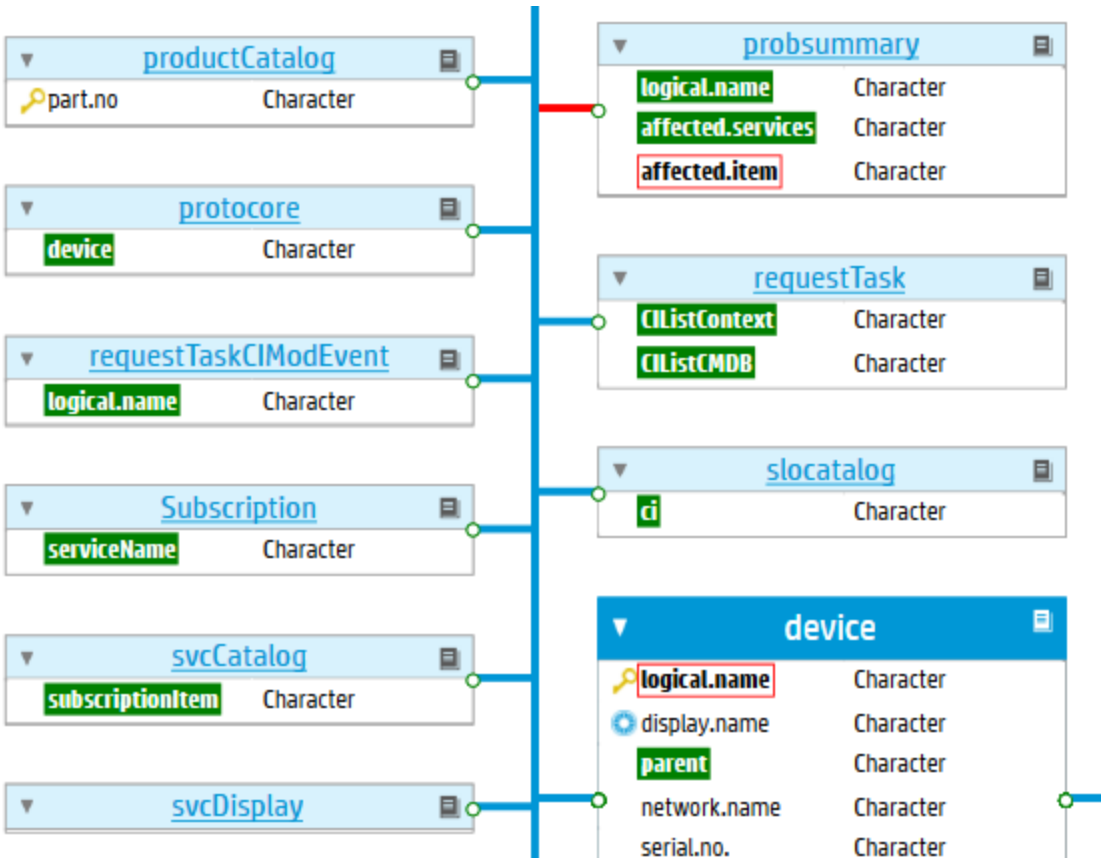
To identify misconfigured reference fields, follow these steps:

1. Navigate to **Tailoring** > **SQL Utities** > **Entity Relationship Diagram**.
2. In the Table Name box, enter the name of a referenced table. For example, enter `device`.
3. Click **Search**.

   Service Manager generates an ERD for the specified table.
4. Select the **Logical** view.
5. Select **Link** and then **As Target** as the filter.
6. Select the key of the referenced table. For example, select the **logical.name** field of the **device** table.
7. Check for red connector lines between the referenced table and other tables. If a red line connects the referenced table and another one, check that table for fields that are enclosed in a red box. Such fields are misconfigured reference fields.

   The following figure shows an example, which indicates the **affected.item** field in the

probsummary table does not have a referenced table configured.



Missing Reference Report utility

Service Manager uses the database to store all information. Relationships between tables are defined by certain tables (**dbdict**, **erddef**, **link**, and **triggers**) and applications other than by primary and foreign keys. There is a challenge to maintain database integrity when there is a change of a referenced value in the database.

Prior to version 9.40, although Service Manager can define relationships between Service Manager files through the **link**, **erddef**, and **relatedObjectMap** tables, there is no way to know if data stored in the files complies with the predefined relationships. As a result, orphan records and data redundancy may occur, which could lead to data corruption or data inconsistency. For example, renaming a Configuration Item or a department may result in orphan records, which still contain a reference to the old name (that is, a missing reference).

As of version 9.40, Service Manager provides a tool, Missing Reference Report, which can generate a report for selected files to help you find out and solve potential data integrity problems. This utility can

help your organization mitigate system risk and save the total cost of ownership (TCO) of adopting HP Service Manager.

Missing Reference Report can report the following items:

- Records that do not follow their primary key constraint
- Records that reference values that do not exist in the target file.
- Records that reference nonexistent field values and contain referential fields that are mapped to a large object (LOB)

Note: This utility skips some relationships in the out-of-box system by default. For details, see "List of skipped out-of-box relationships " on page 344.

Limitations

This utility has the following limitations:

- This utility can detect only missing references to fields of either the **number** or the **character** data type (including number and character fields in arrays). Other data types, such as **logical** and **date/time**, are not supported.
- This utility ignores **new** or **fixed** relationships in which the same source field has multiple parents (target fields) defined. The **location.full.name** field in the **contacts** table is an example, as shown in the following table.

Source Table	Source Field	Target Table	Target Field	Definition	Status
contacts	location.full.name	company	company	link	new
contacts	location.full.name	location	location.full.name	link	new
contacts	location.full.name	location	location.full.name	link	new

Using the utility

Note: To use this utility, you must have the SysAdmin capability word.

The following tasks describe how you can use this utility:

"Start the refcheck.startup scheduler" on the next page

"Enable data reference checking for a file" on the next page

"Skip data reference checking for a relationship" on the next page

"Run a data reference check against selected files" on page 347

"View the status of existing data reference checks" on page 348

"Cancel a scheduled data reference checking job" on page 349

"View missing reference report results" on page 350

"Identify the root cause of a missing reference" on page 352

Start the refcheck.startup scheduler

Applies to User Roles:

System Administrator

The Missing Reference Report utility relies on the refcheck.startup scheduler to work. Before running this utility, start this scheduler.

To start this scheduler, follow these steps:

1. Navigate to **System Status**, and click **Start Scheduler**.
2. Double-click **refcheck.startup**.

   A message is displayed, indicating the scheduler is started.

Enable data reference checking for a file

Applies to User Roles:

System Administrator

The Missing Reference Report utility can detect records in your database that contain missing references. However, to run this utility on a file, you must first update the data policy record of this file to enable data reference checking for the file. There are several ways of accessing a data policy record. The following steps use the Database Dictionary utility.

Note: Data reference checking is enabled by default for the following files in the out-of-box system: **assignment**, **dept**, **device**, **company**, **operator**, **location**, and **contacts**.

To enable data reference checking for a file, follow these steps:

1. In the Service Manager command line, type `dbdict` and press Enter.
2. In the **File Name** field, enter the name of the file. For example, **probsummary**.
3. Click **Search**. A list of records is displayed.
4. Click **probsummary** in the list. The **probsummary** DBDICT record opens.
5. From the DBDICT record, click **More** or the More Actions icon, and select **Data Policy**.

   The data policy record opens.
6. Select the **Enable Data Reference Check** option.
7. Click **Save**.

   By default, enabling data reference checking for a file will apply to all relationships of the file, no matter whether the file is a source file or target file in the relationships. However, you can choose to skip data reference checking for a specific relationship of this file. For details, see "Skip data reference checking for a relationship" below.

   Once you have enabled data reference checking for all desired files and specified the relationships that must be skipped for data reference checking, you are ready to run the Missing Reference Report utility against the files. See "Run a data reference check against selected files" on page 347.

Skip data reference checking for a relationship

Applies to User Roles:

System Administrator

The **Enable Data Reference Check** flag in a relationship record controls whether or not this relationship is skipped when you run a data reference check. The Missing Reference Report utility only scans relationships that have this flag turned on. This mechanism is useful when you want to skip data reference checking for a specific relationship. For example, you already know that a relationship is defined incorrectly, you do not care about the relationship, or you simply want to skip certain fields of a file for performance considerations.

Note: Some out-of-box relationships have data reference checking disabled by default. For more information, see "List of skipped out-of-box relationships " on the next page.

To skip data reference checking for a relationship, follow these steps:

1. Navigate to **Tailoring** > **Tailoring Tools** > **Relationship Manager**.
2. Perform a search to locate the relationship you want to configure.
3. If the relationship has the **Enable Data Reference Check** flag turned on, click **Skip Data Reference Check** on the task bar.

   The **Enable Data Reference Check** check box is cleared. Also, an **Enable Data Reference Check** button appears on the task bar; if you click this button, data reference checking is enabled again.

   Now, if you run a data reference check against the source file or target file of this relationship, this relationship is skipped.

   Note: The **Enable Data Reference Check** setting is retained when running the Build Relationships functionality. For more information, see "Build relationships " on page 360.

List of skipped out-of-box relationships

HP Service Manager has a number of out-of-box relationships that do not really require data reference checking, such as relationships for operator names. For example, your Service Manager system may contain records that reference the names of some former employees who have already resigned from your organization. However, such missing references do not really require any remedial actions.

For this reason, Service Manager allows you to disable data reference checking for a specific relationship. For details, see "Skip data reference checking for a relationship" on the previous page. Additionally, the Missing Reference Report utility does not scan some out-of-box relationships, to achieve optimized performance without compromising the functionality.

The following table lists the out-of-box relationships that are skipped when running the Missing Reference Report utility. These relationship records have their **Data Reference Check** flag turned off by default.

Source File	Source Field	Target File	Target Field	Relationship Type	Definition Type
assignment	company	company	company	Many to Many	link
budgetcode	manager	contacts	contact.name	Many to One	link

Source File	Source Field	Target File	Target Field	Relationship Type	Definition Type
cm3t	header,requested.by	contacts	contact.name	Many to One	link
contacts	manager	contacts	contact.name	Many to One	link
incidents	contact.name	contacts	contact.name	One to One	erddef
incidents	alternate.contact	contacts	contact.name	Many to One	link
incidents	callback.contact	contacts	contact.name	Many to One	link
incidents	owner.name	operator	name	Many to One	link
incidents	opened.by	operator	name	Many to One	link
incidents	updated.by	operator	name	Many to One	link
incidents	closed.by	operator	name	Many to One	link
knownerror	closed.by	operator	name	Many to One	link
knownerror	updated.by	operator	name	Many to One	link
ocml	contact.name	contacts	contact.name	Many to One	link
ocml	employee.id	contacts	user.id	Many to Many	link
ocml	ship.to.code	location	location.code	Many to Many	link
ocml	ship.to.code	location	location	Many to One	link
ocml	coordinator	operator	name	Many to One	link
ocml	coordinator	operator	full.name	Many to Many	link

Source File	Source Field	Target File	Target Field	Relationship Type	Definition Type
ocmo	bill.to.code	location	location.code	Many to Many	link
ocmo	ship.to.code	location	location.code	Many to Many	link
ocmo	orig.operator	operator	name	Many to One	link
ocmq	requested.for	contacts	contact.name	Many to One	link
ocmq	orig.operator	operator	name	Many to One	link
ocmq	work.manager	operator	name	Many to One	link
oncallsched	exception.schedule,exception.schedule,ex.contact	contacts	contact.name	Many to One	link
oncallsched	exception.schedule,exception.schedule,ex.contact	operator	name	Many to One	link
patcotask	rm.requestor	contacts	contact.name	Many to One	link
patcotask	cm.coordinator	operator	name	Many to One	link
patcotask	cm.work.manager	operator	name	Many to One	link
patcotask	rm.assigned.to	operator	name	Many to One	link
patcotask	cm.assign	operator	name	Many to One	link
patcotask	problem.owner	operator	name	Many to One	link
patcotask	rm.coordinator	operator	name	Many to One	link
problem	middle,assignee.name	operator	name	Many to One	link
probsummary	actor	operator	name	Many to One	link

Source File	Source Field	Target File	Target Field	Relationship Type	Definition Type
probsummary	opened.by	operator	name	Many to One	link
probsummary	updated.by	operator	name	Many to One	link
rootcause	updated.by	contacts	full.name	Many to Many	link
rootcause	updated.by	contacts	contact.name	Many to One	link
timeperiodDefinition	created.by	operator	name	Many to One	link
timeperiodDefinition	updated.by	operator	name	Many to One	link

For more information about relationships, see "Add a relationship record" on page 366.

Run a data reference check against selected files

Applies to User Roles:

System Administrator

Once you have enabled data reference check for desired files, you can run a data reference check against them to find out invalid records associated with them.

Note: The Missing Reference Report utility can detect only missing references to fields of either the **number** or **character** data type (including number or character fields in arrays). Other data types, such as **logical** and **date/time**, are not supported.

To run a data reference check, follow these steps:

1. Navigate to **Tailoring** > **SQL Utilities** > **Missing Reference Report**.
2. Click **Fill** to select the files you want to check.
3. Optionally, select the **Check LOB fields** option to also check fields that are mapped to a large

object (LOB).

> **Caution:** Enabling this option may impact your system performance if the selected files have a large number of LOB field values.

4. Click **Generate**.

   A message is displayed, indicating that the reference check job is scheduled. When the processing of a file is finished, a message is displayed, indicating the progress.

   You can continue to view the job status, as well as the status of other data reference check jobs. See "View the status of existing data reference checks" below.

View the status of existing data reference checks

Applies to User Roles:

System Administrator

Once you have triggered a data reference check, you can view its status, along with the status of other existing data reference checks.

To view the status of a data reference check, follow these steps:

1. Navigate to **Tailoring** > **SQL Utilities** > **Missing Reference Report**.
2. Click **View Status** to view the status of existing data reference checks.
   - **Id**: This is the unique identification number of the report. If you click this field, the report opens.
   - **File Name**: This field are the names of the files against which the data reference check is run. If you selected multiple files, their names are separated by a forward slash, for example, **contacts/operator**.
   - **Status**: This column indicates the status of the check. For each check, the initial status is **Scheduled**. When the job starts to run, the status changes to **In Progress**. When the job is finished, the status becomes **Finished**.
   - **Invalid**: This column indicates the total number of invalid records that are detected for the files that are included in this report. Invalid records reference a nonexistent value.

- **Lob Check**: This column indicates if the **Check LOB fields** option was selected when running the job (**true**: selected).

3. Click **Refresh** to refresh the status display.

> **Tip:** If the status still remains **Scheduled** after a long period of time, go to **System Status** and make sure the **refcheck.startup** background scheduler is started.

If you no longer want a scheduled job to run, you can cancel the job; you can also delete the report of a finished job. See "Cancel a scheduled data reference checking job" below.

You can also view the results of a finished job, and perform necessary data cleanup. See "View missing reference report results" on the next page.

Cancel a scheduled data reference checking job

Applies to User Roles:

System Administrator

Once you have triggered a data reference checking job by clicking the **Generate** button in the Missing Reference Report interface, the job is placed in the job queue with a status of **Scheduled**. You can cancel a scheduled job if you like.

> **Note:** Once the job has started running (the status has changed to **In Progress**), you cannot cancel it. If a job has finished, you can delete the job report.

To cancel a scheduled data reference check job, follow these steps:

1. Navigate to **Tailoring** > **SQL Utilities** > **Missing Reference Report**.
2. Click **View Status**.
3. Select a job whose status is **Scheduled**.
4. Click **Delete**.

> **Note:** If a job has a **Finished** status, clicking **Delete** will delete the report instead.

5. Select **Yes** in the confirmation dialog.

   The job is canceled and removed from the list.

View missing reference report results

Applies to User Roles:

System Administrator

Data changes may result in invalid records in your database. For example, a department name has changed in your database, which may cause previous incidents to disappear from an Incident Manager's “Incidents assigned to my group” view. In this case, you can run a data reference check on the **dept** table to find out the invalid records, and further view the detailed results so that you can clean up the invalid data accordingly.

Tip: Instead of using the steps described here, you can directly view missing reference report results by clicking the **Id** field in the Missing Reference Report Status form. See "View the status of existing data reference checks" on page 348.

To view the results of a data reference check and fix the invalid data, follow these steps:

1. Navigate to **Tailoring** > **SQL Utilities** > **Missing Reference Report**.
2. In the **Report Set** box, click Fill to select the data reference check report.

   Note: The **Batch ID** field corresponds to the **Id** field that appears when you click **View Status**.

3. Optionally, complete one or more of the following fields to narrow down your results:
   - Table Name: the name of a file on which the data reference check was performed
   - Field: the name of a field in the file
   - Relationship Type: a type of relationships for the file
   - Definition Type: a type of relationship definitions for the file
4. Click **View Result**.

A list of relationships for the files is displayed. In these relationships, the files can be either a source file or a target file.

5. Select a relationship from the list. The reference check results of the relationship are displayed. The results consist of two sections: **Details** and **Record Analysis**.

6. View the information in the Details section. This read-only section displays the relationship details and statistic information about the source and target records.

Field	Description	Notes
Id	The unique identifier number of the relationship record	
Source File	The name of the source file of the relationship record	For more information, see "Add a relationship record" on page 366
Source Field	The name of the source field of the relationship record	
Target File	The name of the target file of the relationship record	
Target Field	The name of the target field of the relationship record	
Relationship Type	The type of the relationship record	
Definition Type	The type of the relationship definition **Note:** To view the relationship definition record, click the **Go to definition** arrow button next to this field.	
Definition Query	The query of the relationship definition	
From Lob	Indicates if the source field or target field is mapped to a large object (LOB)	
Report ID	The unique identifier number of the date reference check report	
Report Time	The time when the report was generated	
Null Records	The number of null records based on the	

Field	Description	Notes
	relationship	
Source Records	The number of source records based on the relationship	
Target Records	The number of target fields based on the relationship	
Missing Reference Records	The number of records that contain a missing reference	
Source field in First unique key	Indicates if the source field is, or is part of, the first unique key of the source file.	The first unique key of a file may consist of multiple fields.
Target field in First unique key	Indicates if the target field is, or is part of, the first unique key of the target file.	

7. The Record Analysis section displays the numbers and percentages of valid and invalid records. To view or edit the invalid records, click **Show All Invalid Records**.

   A list of invalid records that use the default form of either the source file or the target file is displayed.

   Note: If there is only one invalid record, the invalid record opens directly.

Next, you need to identify the root cause of each invalid record in the report and take necessary actions. For more information, see "Identify the root cause of a missing reference" below.

Identify the root cause of a missing reference

If a missing reference report has detected any invalid records, you need to identify their root causes. Remember that the data in the source file should also exist in the target file, otherwise the data is an “orphan” or there is a non-relational **link** or **erddef** definition behind that “orphan” data.

Example 1 : The missing reference is caused by a pure orphan value.

This example demonstrates the steps to handle a missing reference, using the **operator** file and out-of-box data in the system.

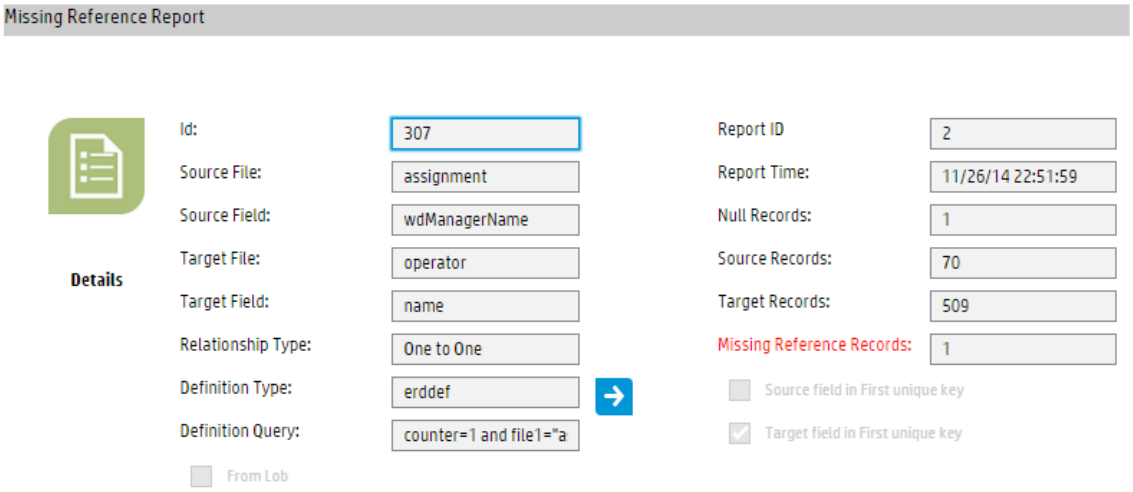
Task 1. Generate a missing reference report.

1. Make sure the background scheduler for data reference check is started. For details, see "Start the refcheck.startup scheduler" on page 342.

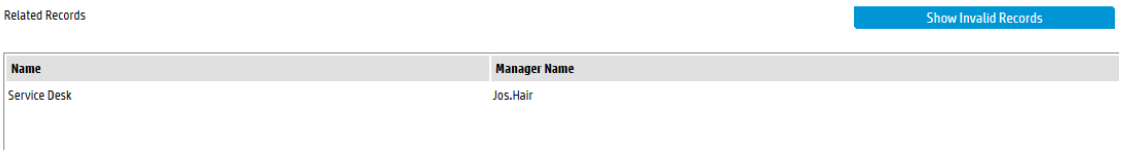
2. Generate a missing reference report for the **operator** file. For details, see "Run a data reference check against selected files" on page 347.

3. View the report results. For details, see "View missing reference report results" on page 350.

   In this example, only one invalid record is reported, as shown in the following figure.



   The Related Records section lists the invalid record (which is the Service Desk assignment group), as shown in the following figure.



   By following the steps above, you can generate a missing reference report for any files. This is only the first step of handling a missing reference report, and your final goal is to resolve each “orphan” value or non-relational link or erddef definition.

Task 2. Understand the report results.

1. Verify that **Jos.Hair** does not exist in the **operator** file. To do this, follow these steps:

a. Type `operator` in the command line, and press Enter.

b. Type `Jos.Hair` in the **Login Name** field, and click **Search**.

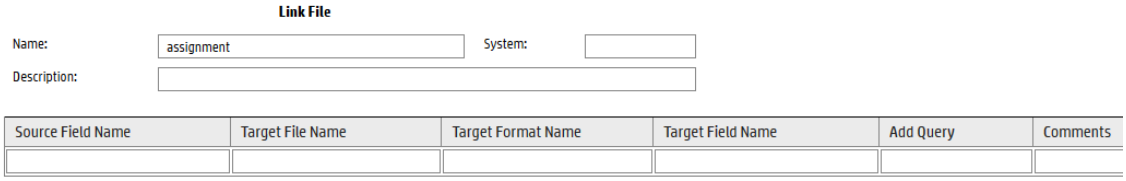
The search returns no results, which means that **Jos.Hair** does not exist in the **operator** file.

2. Open several existing operator records to understand what their Login Name values look like.

> **Note:** This step can help you find out if all values in a field should follow certain rules.

3. In the command line, type `link` and press Enter. The Search Link Records form opens.

4. Type `assignment` in the **Name** field, and click **Search**.



5. Locate the row in which the Source Field Name is **wdManagerName**.

This is the relevant relationship definition. You should check to see if this link relationship complies with the business logic or is designed as desired. If not, the inappropriate link definition is probably the root cause of the “orphan” data; in this situation, you need to take actions on the **link** definition. If yes (which is true in most scenarios), the “orphan” data is then a pure “orphan” (that is, a real missing reference).

Link File

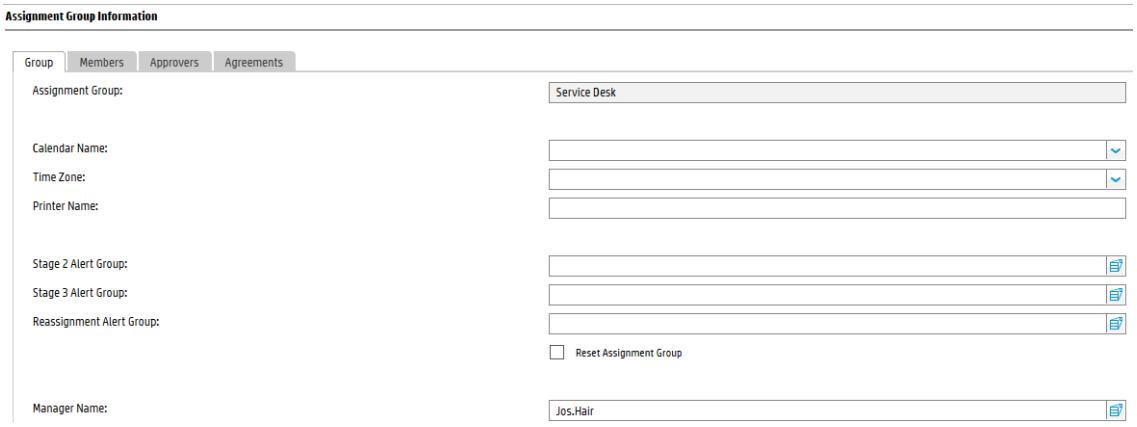
Name: assignment System: miscellaneous

Description:

Source Field Name	Target File Name	Target Format Name	Target Field Name	Add Query
operators	operator	operator.view	name	$query
assignment2	assignment		name	
assignment3	assignment		name	
reassignment	assignment		name	
company	company		company	
coordinator.incident	operator	operator.view	name	
coordinator.change	operator	operator.view	name	
coordinator.problem	operator	operator.view	name	
wdManagerName	operator	operator.view	name	
coordinator.request	operator	operator.view	name	
vendor	vendor		vendor	
$relatedId	sla		name	$query

6. Find the record in the **assignment** table whose Manager Name is **Jos.Hair**.

   The Manager Name field is the **wdManagerName** field in the **assignment** table, which is mapped to the **name** field in the **operator** file.



7. According to the business logic, manager “Jos.Hair” should not have appeared in the assignment group record, because an assignment group manager should also be an operator in Service Manager. Furthermore, if you remove “Jos.Hair” from the Manager Name field and then click the Fill button, a list of operators is displayed. However, your previous search has already shown that “Jos.Hair” does not exist in the **operator** file. Therefore, it is true that “Jos.Hair” is an orphan value.

   In summary, the root cause of this orphan value is that the assignment group record defines a manager name that cannot be found in the system.

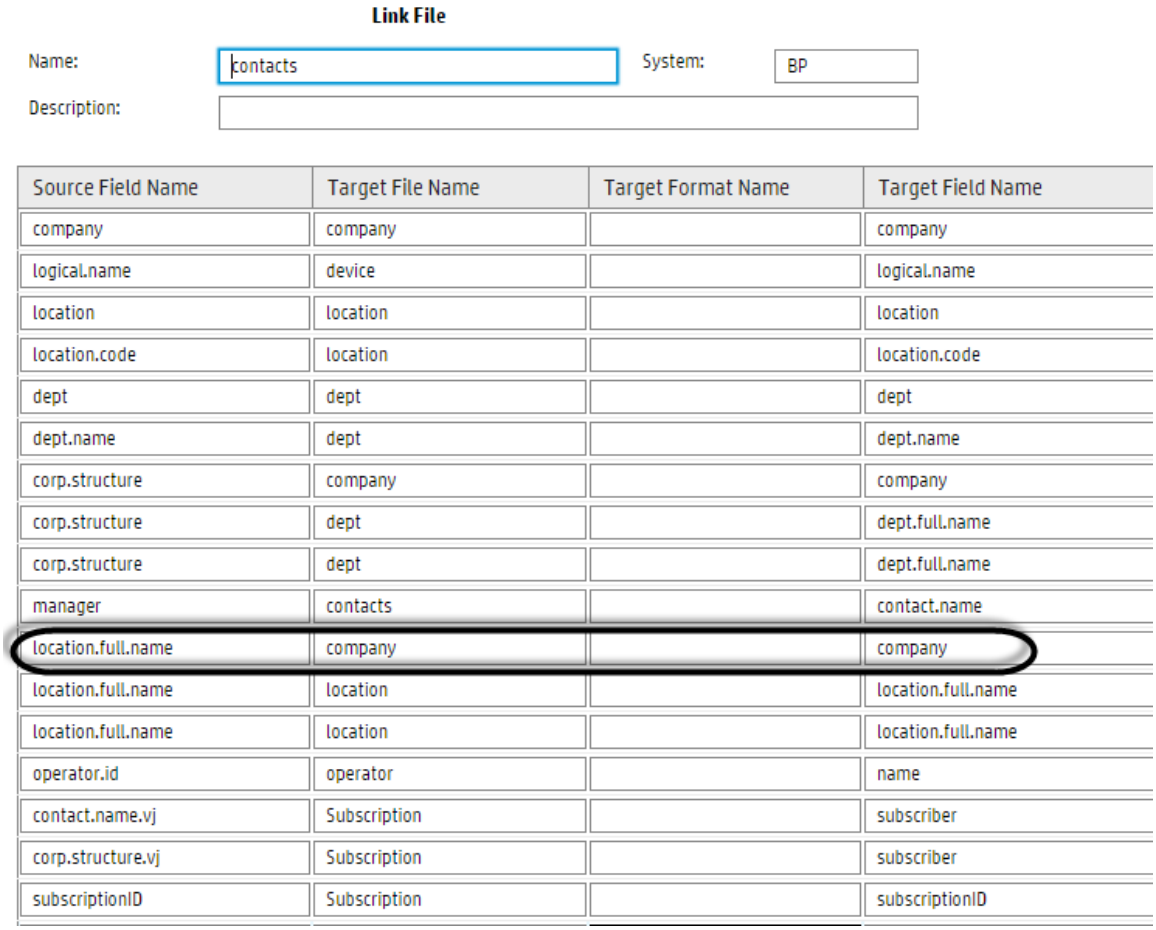
Example 2. The missing reference is caused by an inappropriate link definition

The following describes an example of an inappropriate link definition.

1. Navigate to **System Administration** > **Tailoring** > **Tailoring Tools** > **Relationship Manager**, locate the following relationship record through a search.

SOURCE_ FILE	SOURCE_FIELD	TARGET_ FILE	TARGET_ FIELD	RELATION_ TYPE	DEFINITION_ TYPE
contacts	location.full.name	company	company	Many to Many	link

2. In the relationship record, click **Goto Definition** to open the link definition, and locate the row that has a **Source Field Name** of **location.full.name** and a **Target Field Name** of **company**.

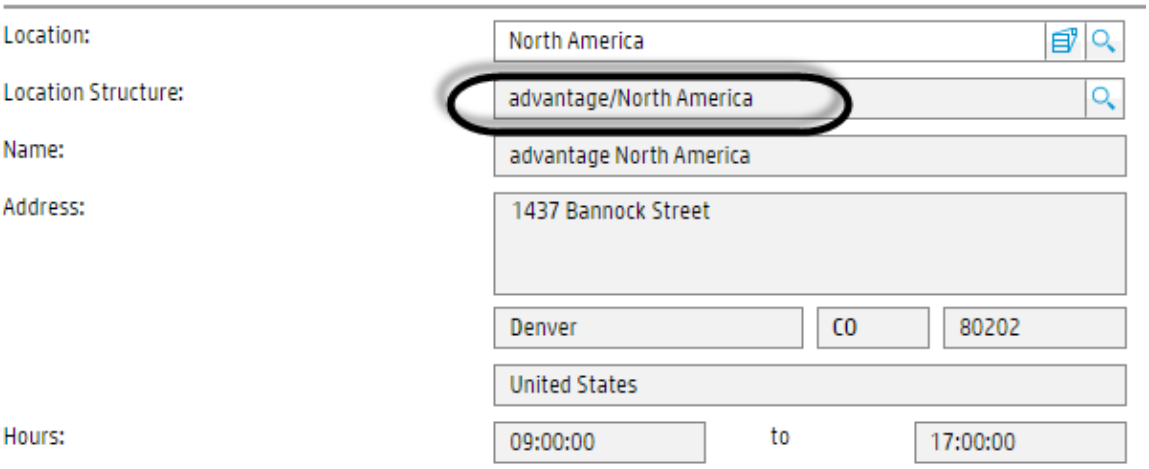


Source Field Name	Target File Name	Target Format Name	Target Field Name
company	company		company
logical.name	device		logical.name
location	location		location
location.code	location		location.code
dept	dept		dept
dept.name	dept		dept.name
corp.structure	company		company
corp.structure	dept		dept.full.name
corp.structure	dept		dept.full.name
manager	contacts		contact.name
location.full.name	company		company
location.full.name	location		location.full.name
location.full.name	location		location.full.name
operator.id	operator		name
contact.name.vj	Subscription		subscriber
corp.structure.vj	Subscription		subscriber
subscriptionID	Subscription		subscriptionID

3. Type `contacts` in the Service Manager command line, press Enter, and then open a contact

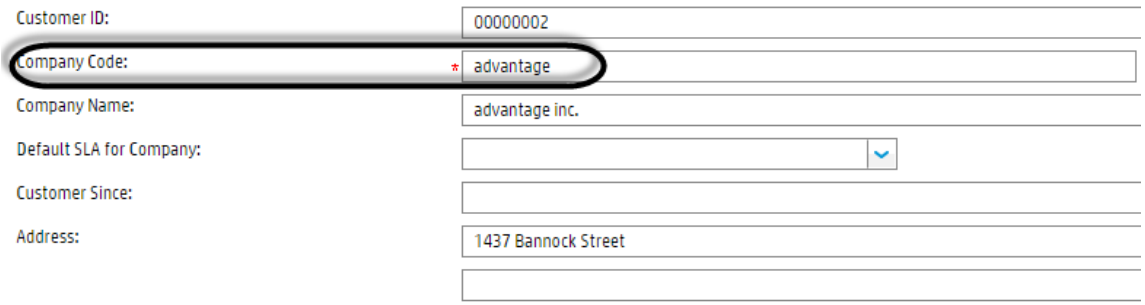
record through a search.

The **location.full.name** field in the **contacts** table has a label of **Location Structure** in contact records (the **Address** tab). As shown in the following figure, this field should be a combination of company and location according to the out-of-box business logic (for example, **advantage/North America**).



4. Open the **company** table in Database Manager, and locate a company record through a search.

   The **company** field in the **company** table has a label of **Company Code** in company records. The following figure shows an example value.



   It is obvious that the **link** file defines a pair of source and target fields that have different value structures and therefore should not be mapped to each other. So this is an inappropriate **link** definition.

> **Note:** Because of a limitation (see the Limitations section in "Missing Reference Report utility" on page 340), the Missing Reference Report utility does not report invalid records caused by

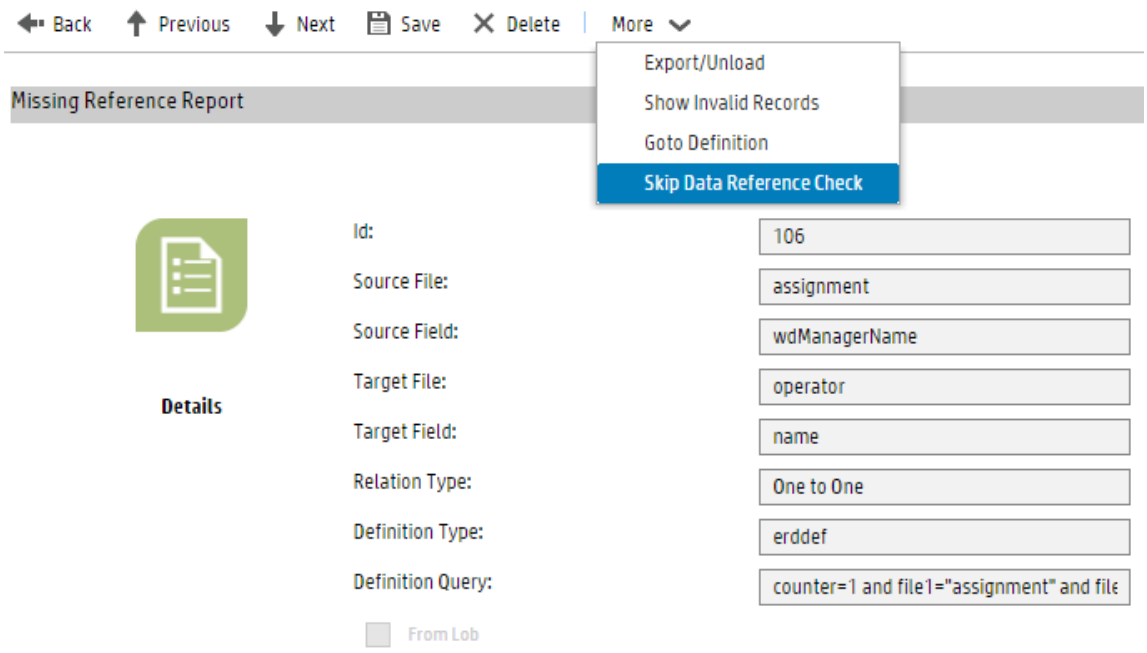
this **link** definition when running a date reference check against the **contacts** table.

Steps to analyze a missing reference record

Normally, you can use the following process (which is similar to the example process described above) to analyze a missing reference record:

1. Search in the target file to make sure the orphan value cannot be found in the target file, and examine the data in the target file to see what a correct value should look like.
2. Go to the **link** or **erddef** record to locate the relevant relationship definition.
3. Identify the problem:
   - Find the invalid "orphan" value in the source file.
   - Find the non-relational definition in the **link** or **erddef** file.
4. Analyze the problem based on the business logic.
5. Find out the root cause of the problem.
6. Perform the following tasks based on the root cause identified:
   - If the invalid record does contain a missing reference, correct the wrong information and then click **Save**. Alternatively, you can leave it as is if it is only historic data or you can click **Delete** to delete the record if you no longer need to keep it.
   - If the invalid record was caused by an inappropriate **link** or **erddef** definition, update the **link** or **erddef** definition.
   - Mark the corresponding relationship record so that it is skipped for future data reference checking. To do so, on the report details page of this relationship, click **More** or the More Actions menu and then select **Skip Data Reference Check** (see the following figure). If you open this relationship record from Relationship Manager, you will see the **Enable Data Reference**

Check flag in the relationship is now turned off.



Relationship Manager

Relationship Manager is a tool that can automatically discover entity relationships between tables and fields in your HP Service Manager system by querying the **erddef**, **link**, and **relatedObjectMap** tables, and then save the results in the **relation** table. It also enables you to manually create relationships and manage them.

Also, Relationship Manager provides output that is consumed by the Missing Reference Report utility and Entity Relationship Diagram utility.

Note: To use Relationship Manager, you must have the SysAdmin capability word.

Using Relationship Manager, you can perform the following tasks:

- Automatically trigger a reload of all relationships in your system. See "Build relationships " on the next page.
- Find entity relationships for tables and fields with other tables. See "Search for relationships between specific tables and fields" on page 361.
- Manually add relationships. See "Add a relationship record" on page 366.

- Update invalid or fixed relationships. See "Update an invalid or fixed relationship" on page 367.
- Update the relationship definition of a relationship record. See "Access the relationship definition of a relationship record" on page 365.

Build relationships

Applies to User Roles:

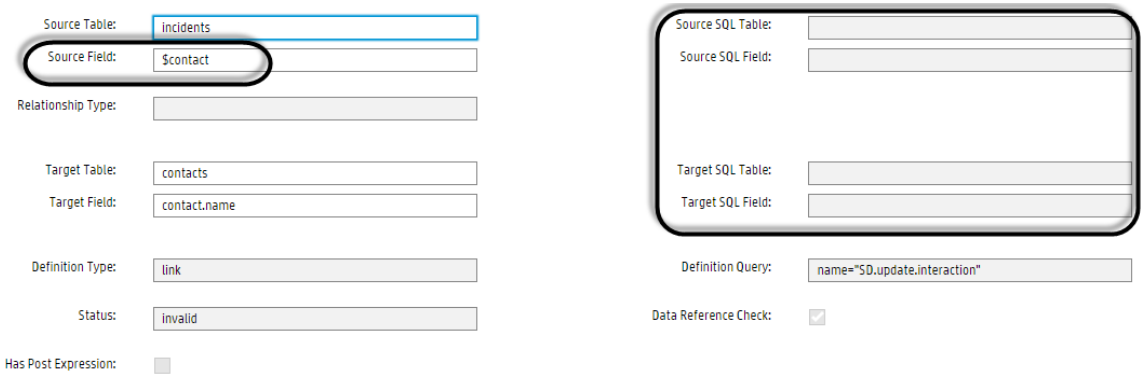
System Administrator

Relationship Manager can discover all relationships that are defined by the **erddef**, **link**, and **relatedObjectMap** tables. When upgrading to or installing the HP Service Manager 9.40 applications (or a later version of the applications), the **Build Relationships** functionality is automatically executed to load all relationships and save them in the **relation** table. When the **Build Relationships** functionality is executed, the following rules apply:

- Relationships that were manually created are retained, and all other previously built relationships are cleared and then rebuilt.
- Duplicate relationships from multiple sources are saved, and identified by the unique key, **id**.
- Relationship Manager generates a pair of relationship records for both relationship directions from one **erddef** definition. See the following table for an example.

Source Table	Source Field	Target Table	Target Field	Definition Type	Relationship
category	name	probsummary	category	erddef	One to Many
probsummary	category	category	name	erddef	Many to One

- Relationships without a SQL name for any of the **Source Table**, **Source Field**, **Target Table**, or **Target Field** fields are saved as an **invalid** relationship. The following figure shows an example, in which the source field uses a variable.



- Links that have a post expression are saved with the **Has Post Expression** field set to **true**.
- The **Enable Data Reference Check** setting (true or false) is retained when rebuilding each relationship. See "Skip data reference checking for a relationship" on page 343.
- While relationships are being built, messages are displayed, showing the building progress. If an error occurs, you are prompted to fix it and retry.

There may still be times when your system contains outdated relationship records. The **Build Relationships** option allows you to manually trigger a reload of all relationship records in your system. To manually build relationships, follow these steps:

1. Navigate to **Tailoring** > **Tailoring Tools** > **Relationship Manager** to open Relationship Manager.
2. Click the **Build Relationships** button.

   Service Manager starts to purge all relationships that were manually created, and then reload all the relationships in your system. This process may take a while. When the building process is complete, a "done" message is displayed.

   If an error occurs, you are prompted to fix it and retry.

Once all relationships are discovered and saved in the **relation** table, you can search for all relationships or relationships between specific tables and fields. See "Search for relationships between specific tables and fields" below.

Search for relationships between specific tables and fields

Applies to User Roles:

System Administrator

In Relationship Manager, you can search for relationships by one or more of the following fields: Table Name, Field Name, Definition Type, and Status.

To search for relationships between specific tables and fields, follow these steps:

1. Navigate to **Tailoring** > **Tailoring Tools** > **Relationship Manager** to open Relationship Manager.
2. In the **Table Name** field, type the name of either the source or the target table. For example, **operator**.
3. In the **Field Name** field, type the name of a field in the table you specified. For example, **name**.
4. In the **Definition Type** field, select a definition type: **link**, **erddef**, **relatedObjectMap**, or **manual**.
5. In the **Status** field, select a status: **new**, **invalid**, or **fixed**.
6. Click **Search**.

   A list of records is displayed, showing the relationships between the specified field and fields in other tables.

   Note: You can also perform a search by leaving some or all of the search fields empty. We strongly recommend that you review all relationships to ensure they are correct, because some other features (such as the Entity Relationship Diagram utility and the Missing Reference Report utility) heavily rely on the correctness of your relationship records.

7. Double-click a record from the list to open the details page and view the details.

   A relationship record may contain the following fields.

Field	Description	Notes
Source Table	One of the following fields depending on the definition type of the relationship record (see Definition Type in this table): ◦ **First Filename** in the **erddef** definition ◦ **Name** in the **link** definition ◦ **Source Table** in the **relatedObjectMap** definition	
Source Field	One of the following fields depending on the definition type of the relationship record:	

Field	Description	Notes
	◦ **Field Names from First Filename** in the **erddef** definition ◦ **Source Field Name** in the **link** definition ◦ **Source Field** in the **relatedObjectMap** definition	
Relationship Type	The type of the relationship. It can have one of the following values: ◦ One to Many ◦ Many to One ◦ One to One ◦ Many to Many	System-generated
Target Table	One of the following fields depending on the definition type of the relationship record: ◦ **Second Filename** in the **erddef** definition ◦ **Target File Name** in the **link** definition ◦ **To Table** in the **relatedObjectMap** definition	
Target Field	One of the following fields depending on the definition type of the relationship record: ◦ **Field Names from Second Filename** in the **erddef** definition ◦ **Target Field Name** in the **link** definition ◦ **To Field** in the **relatedObjectMap** definition	
Definition Type	The type of the relationship definition, which indicates how the relationship is defined: ◦ **link**: the relationship is defined in a **link** record. ◦ **erddef**: the relationship is defined in an **erddef** record.	System-generated

Field	Description	Notes
	◦ **relatedObjectMap**: the relationship is defined in a **relatedObjectMap** record. ◦ **manual**: the relationship is manually defined.	
Status	The status of the relationship: ◦ **new**: the relationship was automatically created and contains valid data. ◦ **fixed**: the relationship was manually created or updated and contains valid data. ◦ **invalid**: the relationship was automatically created and contains invalid data (any of the SQL table names or SQL field names cannot be found).	System-generated
Has Post Expression	Indicates if the link record of the relationship has a post expression defined. Applicable only for relationships whose definition type is **link**: ◦ true ◦ false **Note:** A post expression is an expression that is executed after the link function is triggered by clicking **Fill**.	System-generated
Source SQL Table	The SQL Table name of the source table	System-generated
Source SQL Field	The SQL Name of the source field	System-generated
Target SQL Table	The SQL Table name of the target table	System-generated
Target SQL Field	The SQL Name of the target field	System-generated
Definition Query	The query of the relationship definition	System-generated

Field	Description	Notes
	Note: This field is empty if the relationship is manually created.	
Enable Data Reference Check	This option indicates if this relationship is skipped when you run a data reference check against its source file or target file: ◦ If this option is selected: not skipped ◦ If this option is not selected: skipped	System-generated

Note: You can use one of the following methods if you have any invalid relationships:

- Update its relationship definition using the **Goto Definition** button. See "Access the relationship definition of a relationship record" below.
- Add a new relationship for it by using the **Copy** button. See "Add a relationship record" on the next page.
- Update the relationship if the Definition Type is **link**. See "Update an invalid or fixed relationship" on page 367.

Access the relationship definition of a relationship record

Applies to User Roles:

System Administrator

For relationships whose Definition Type is not **manual**, Relationship Manager provides a **Goto Definition** button, which enables you to view the relationship definition and update it if needed.

Note: Manually created relationships, that is, relationships whose Definition Type is **manual**, do not have a **Goto Definition** button.

To access the relationship definition of a relationship, follow these steps:

1. Navigate to **Tailoring** > **Tailoring Tools** > **Relationship Manager**.
2. Complete the **Table Name** and **Field Name** fields.

   Note: This step is optional. You can leave one or both of the fields empty.

3. Click **Search**. A list of relationships is displayed.
4. Select the relationship record from the list, and click **Goto Definition** in the toolbar of the relationship detail pane.
5. View the relationship definition, and perform any of the following tasks if needed:
   - Update the relationship definition and then click **Add** to save it as a new relationship definition.
   - Update the relationship definition and click **Save** to save your changes to the existing relationship definition.
   - Click **Delete** to delete the relationship definition.

Add a relationship record

Applies to User Roles:

System Administrator

Relationship Manager allows you to manage not only relationships that are defined by the **erddef**, **link**, and **relatedObjectMap** tables, but also relationships that you manually create. You can manually add a relationship record, and then update it or delete it as needed.

Note: Manually created relationship have a Definition Type of **manual**. You can update or delete this type of relationships .

To add a relationship record, follow these steps:

1. Navigate to **Tailoring** > **Tailoring Tools** > **Relationship Manager** to open Relationship Manager.
2. In the **Table Name** field, type the name of the table for which you want to add a relationship. For example, **operator**.
3. Optionally, enter other search fields as needed.

4. Click **Search**. A list of relationships is displayed.
5. Select a record from the list, and click **Copy**.
6. Change the values of the following fields as needed: **Source Table**, **Source Field**, **Target Table**, and **Target Fields**.
7. Click **Add**. Service Manager adds the relationship record.

   Caution: If your input contains incorrect data, data validation fails and the record is not added.

   The **Relationship Type** and **Status** fields are automatically populated with **manual** and **fixed**, respectively.
8. The **Data Reference Check** option is automatically selected. If you want to skip this relationship for data reference checking, click **Skip Data Reference Check**, and then click **Save**. See "Skip data reference checking for a relationship" on page 343.
9. Verify the new record.
   a. Click **OK** to return to Relationship Manager.
   b. Perform a search to locate the record you added. The record should have the following values:
      - **Relationship Type**: manual
      - **Status**: fixed
10. If needed, you can further update the record, and then click **Save**.
11. If you no longer want to keep the record, click **Delete** to delete it.

Update an invalid or fixed relationship

Applies to User Roles:

System Administrator

In Relationship Manager, you can update only relationship records that have either of the following:

- a Status of **invalid** and Definition Type of **link**
- a Status of **fixed**

Note: You cannot update a relationship whose status is **new**.

To update a relationship record, follow these steps:

1. Navigate to **Tailoring** > **Tailoring Tools** > **Relationship Manager** to open Relationship Manager.
2. Optionally, enter or select a value in the **Table Name** and **Field Name** fields.
3. In the **Status** field, select **fixed** or **invalid**.
4. If you selected **invalid** in the **Status** field, select **link** in the **Definition Type** field.
5. Click **Search**. A list of relationships is displayed.
6. Select the record that you want to update, and update the relationship record as needed.
7. Click **Save** to save your changes.

   If the record had an **invalid** status, it now changes to a **fixed** status. If it had a **fixed** status, the status does not change.

   Caution: If your input contains incorrect data, data validation fails and the record is not updated.

   Note: If you no longer want to keep the record, click **Delete** to delete the record.

Send Documentation Feedback

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

Feedback on Database Management help topics for printing (Service Manager 9.41)

Just add your feedback to the email and click send.

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

We appreciate your feedback!


Java™
COMPATIBLE