

Peregrine

ServiceCenter

System Administrator's Guide, Volume 2

Release 5.1

Copyright © 2002-2003 Peregrine Systems, Inc. or its subsidiaries. All rights reserved.

Information contained in this document is proprietary to Peregrine Systems, Incorporated, and may be used or disclosed only with written permission from Peregrine Systems, Inc. This book, or any part thereof, may not be reproduced without the prior written permission of Peregrine Systems, Inc. This document refers to numerous products by their trade names. In most, if not all, cases these designations are claimed as Trademarks or Registered Trademarks by their respective companies.

Peregrine Systems® and ServiceCenter® are registered trademarks of Peregrine Systems, Inc. or its subsidiaries.

This document and the related software described in this manual are supplied under license or nondisclosure agreement and may be used or copied only in accordance with the terms of the agreement. The information in this document is subject to change without notice and does not represent a commitment on the part of Peregrine Systems, Inc. Contact Peregrine Systems, Inc., Customer Support to verify the date of the latest version of this document.

The names of companies and individuals used in the sample database and in examples in the manuals are fictitious and are intended to illustrate the use of the software. Any resemblance to actual companies or individuals, whether past or present, is purely coincidental.

If you need technical support for this product, or would like to request documentation for a product for which you are licensed, contact Peregrine Systems, Inc. Customer Support by email at support@peregrine.com.

If you have comments or suggestions about this documentation, contact Peregrine Systems, Inc. Technical Publications by email at doc_comments@peregrine.com.

This edition applies to version 5.1 of the licensed program.

Peregrine Systems, Inc.
Worldwide Corporate Headquarters
3611 Valley Centre Drive San Diego, CA 92130
Tel 800.638.5231 or 858.481.5000
Fax 858.481.1751
www.peregrine.com



Contents

	About This Guide	7
	Knowledge Requirements	8
	Organization of This Guide.	9
	ServiceCenter’s Best Practices in Processes and Workflow	10
	Sample Data in the System	10
	Sample Screens and Examples.	11
	Documentation Web Site.	11
	Contacting Peregrine Systems.	11
	Peregrine’s CenterPoint Web Site	11
	Contacting Education Services	12
Chapter 1	Unload Script Utility	13
	Overview of the Unload Script Utility	13
	Creating an Unload Script	14
	Commonly Scripted Files	20
	Protecting Formats	21
	Editing an Unload Script	23
	Validating an Unload Script	26
	If the Validity Check Fails	28
	Unloading Records from an Unload Script	28
	Purging Records from an Unload Script	32

Chapter 2	Purge/Archive	35
	Understanding the Purge/Archive Function	35
	How to Access the Purge/Archive Function	36
	How to Archive Records in the Foreground	38
	How to Purge Records in the Foreground	44
	How to Purge/Archive in the Foreground	46
	How to Schedule Purge/Archive in the Background.	47
	Using Purge/Archive with Network Management Applications	52
	Copy Database Records to Provide On-Line Archival Copies.	54
Chapter 3	Data Mapping	55
	Accessing the <i>Data Map</i> File	57
	Field Descriptions.	60
	Data Map Types	63
	Type 0	63
	Type 1	64
	Type2.	65
	Rules Summarizing How Mapping Conditions Affect Data Map	66
	Invoking the Data Map Utility from RAD.	67
	report File	70
	format File.	73
	formatctrl File	76
	enclapplication File	78
	application File.	82
	code File.	84
	Unloading with a Data Map.	86
	Special Considerations.	91
Chapter 4	Unload/Load	93
	Overview	93
	Unloading ServiceCenter Data	93
	Unloading a Single Record from a Database.	95
	Unloading Multiple Records from One Database	103
	Unloading Multiple Records from a Single File Using Value-list	105
	Unloading Records from Multiple Databases	111
	Unloading a QBE Only (No Data Records) to Excel	115

	Unloading a Database Dictionary Only (No Data Records)	117
	Unloading a ServiceCenter Report Writer Report	119
	Loading ServiceCenter Databases	127
	Executing the Load as a Foreground Process	128
	Executing the Load as a Background Process	134
Chapter 5	Export Utility	137
	Overview of the Export Function	138
	Rules Used in the Export Utility	139
	Accessing the Export Utility.	140
	Exporting Records Example.	141
	Creating an Export Descriptor Record	148
	Modifying an Export Descriptor Record	152
	Exporting Records in the Foreground	153
	Exporting Records in the Background	159
	Printing an Export Descriptor Record	161
	Using Format Control with Export.	162
	Building the Format Control Record.	164
	Exporting to Excel in a Different Language	169
Chapter 6	Import Utility	171
	Overview of the Import Function	172
	Import Text Using the Import Wizard	172
	Accessing the Wizard	173
	Modifying an Import Descriptor record	183
	Deleting an Import Descriptor record	188
	Import records in Foreground	188
	Import records in Background	189
	Print an Import Descriptor record	191
	Using Format Control with Import	192

Chapter 7	Telephony	195
	Overview	195
	TAPI	196
	Prerequisites	196
	Configurations	196
	Call flow.	198
	Architecture	199
	Connectivity	201
	Enabling TAPI for ServiceCenter	201
	Receiving Calls	203
	Making Calls	206
	Debugging.	206
	TeleCenter Manages Your Telecommunications System.	209
Chapter 8	LDAP Server Interface	211
	Overview	211
	How LDAP Works	211
	Template	212
	System default records.	213
	Updating mapped fields	213
	Mapping other files to LDAP	213
	Performing Data Queries.	214
	Setup	215
	Defining the Default LDAP Server.	215
	Mapping individual Files and Fields to the LDAP Directory	219
	Using LDAP with a Secure Socket Layer (SSL) Connection	223
	Control an LDAP Query with sc.ini File Parameters	225
	Create ServiceCenter Records from LDAP Data	225
	Parameters for Binding to the LDAP Server	226
	Login Template.	227
	Defining a Default System Record	232

	LDAP Administrator Capabilities	234
	LDAP Tools	234
	Operator Record	234
	Manage LDAP Connection with sc.ini File Parameters	236
	Defining Alternate Servers	236
	Disabling LDAP	237
Chapter 9	Load Manager	239
	Overview	239
	Requirements	240
	Prerequisites	240
	Related Documentation	240
	What is Load Manager?	241
	Why Use Load Manager?	242
	Installing Load Manager	242
	Configuring Your System.	243
	Configuring the Primary System	243
	Configuring Secondary Servers	244
	Configuring Clients	245
	Using Load Manager.	246
	ServiceCenter Error Messages	247
Chapter 10	Report Writer	249
	Overview	249
	Understanding Report Writer	250
	How to Access Report Writer	250
	How to Create a Simple Report Listing	252
	Creating and Editing Report Formats	264
	How to Query for Reports	266
	How to Use Query Formats with Reports	268
	Creating Primary File Queries	268
	How to Use Initial Formats with Reports	275
	Accessing Forms Designer	276

How to Use Page Breaks and Report Totals	278
Accessing the Totals Sections of a Report	278
Using Simple Subtotals	283
Using Subtotals, Summary Totals, and Grand Totals	285
Using Calculations in a Report	288
Accessing Format Control	290
How to Use Format Control with Reports.	292
Format Control and Report Formats.	292
Using Format Control for Subheaders	293
Detail Formats	298
Using Format Control in a Matrix Report.	301
Subtotal Format	304
Total Format.	305
How to Run Reports in a Foreground	311
How to Schedule Reports for Background Processing	312
Understanding Background Report Processing	312
Scheduling a Report for Background Processing	313
Online Viewing of Reports	320
Accessing the Spoolheader and Spool Files	320
How to View Report Schedule Records	325
How to Copy/Rename Reports	326
How to Delete Reports.	327
How to Use Stacked Queries for Sorting	328
Primary Query Option.	330
Stacked Queries Option	330
Rules to Follow When Defining Stacked Queries.	332
Report Writer Options Menu Selections	334

Chapter 11	Benchmark Utility	337
	Overview	337
	System Requirements	338
	Preparing to Run the Benchmark Utility	338
	Updating the sc.ini File	338
	Creating ServiceCenter Operators	339
	Accessing the Benchmark Utility Menu.	340
	Creating Benchmark Configuration Records	342
	Starting a Benchmark Test	344
	Monitoring the Progress of the Test	346
	Viewing the Benchmark Test Results	348
	Clearing Test Results	351
	Stopping a Test	351
Appendix A	Keyboard Shortcuts	353
	Index	357

About This Guide

Welcome to the *ServiceCenter System Administrator's Guide, Volume 2*. The *ServiceCenter System Administrator's Guide* is a two-volume set that provides an introduction to ServiceCenter from a system administrator's perspective.

The guide provides instructions for starting, setting up, and managing ServiceCenter. It includes instructions for using database tools, supporting files, and some advanced administrative utilities. It also includes information on the basic ServiceCenter window, use of the applications, logging on and off of ServiceCenter, and using the menus and buttons. All menus and forms presented are those that a system administrator with all available modules would see.

Volume 1 is divided into the following sections:

- Getting Started
- Support Files
- User Access and Security Control
- Print Routines and Printer Setup
- Clocks and the Inactivity Timer
- Time Zone Function and Date Formats
- Monitor System Events
- Mandanten Database Protection
- Command Line Prompt Calls, Appendix A

Volume 2 is divided into the following sections:

- Unload Script Utility
- Purging and archiving records
- Data Mapping
- Loading and unloading records
- Export Utility
- Import Utility
- Telephony
- LDAP Server Interface
- Load Manager
- Report Writer
- Benchmark Utility
- Keyboard Shortcuts, Appendix A

You'll find specific installation and reference information in other ServiceCenter documentation. For instructions on getting started in the applications, see the *ServiceCenter User's Guide*. For instructions on administering the Java client, see the *ServiceCenter Java Client Installation and Setup Guide*.

Knowledge Requirements

A system administrator needs a thorough knowledge of the operating environment in which ServiceCenter is installed and accessed and a basic understanding of the ServiceCenter applications. Before you begin using this guide, thoroughly read and become familiar with the material in the *ServiceCenter User's Guide*.

ServiceCenter installation requirements are specific to the machine where ServiceCenter is being installed. These requirements are listed in the respective installation guides.

Organization of This Guide

This guide is organized around the main system administration tasks associated with setting up and maintaining the ServiceCenter, applications, modules, and user base. The following chart shows you which parts of the manual you need to reference to find the information you need.

To Find This	Look Here
Create standard ServiceCenter unload files through an enhanced user interface.	<i>Unload Script Utility</i> on page 13.
Manage your database size by deleting and/or storing multiple records in an external file.	<i>Purge/Archive</i> on page 35.
Define data maps for any group of associated ServiceCenter records to automatically discover and establish a relationship from a primary record to a secondary record.	<i>Data Mapping</i> on page 55.
Perform load and unload functions on multiple records within a database.	<i>Unload/Load</i> on page 93.
Export any field in any ServiceCenter file, including multiple lines in arrays.	<i>Export Utility</i> on page 137.
Import specified fields from an external file into a ServiceCenter file.	<i>Import Utility</i> on page 171.
Options for computer-telephone integration	<i>Telephony</i> on page 195.
Lightweight Directory Access Protocol (LDAP): setup, template, system defaults, mapping other files to LDAP, initialization file parameters	<i>LDAP Server Interface</i> on page 211.
Set up multiple servers and redirect servers that have reached capacity on memory usage.	<i>Load Manager</i> on page 239.
How to create simple Report Writer reports and query for reports.	<i>Report Writer</i> on page 249.
Determine whether or not the system can handle your expected user volumes and response times	<i>Benchmark Utility</i> on page 337.

ServiceCenter's Best Practices in Processes and Workflow

ServiceCenter uses the Information Technology Infrastructure Library (ITIL) processes and workflow. This is a set of standards and procedures developed by the Central Computer & Telecommunications Agency (CCTA), a United Kingdom government agency, for its best practices in processes and workflow. You will find that Peregrine Systems has given you the tools you will need to perform certain tasks in ServiceCenter. For more information on best practices for using ServiceCenter, see *ServiceCenter Introduction and Best Practices*.

Sample Data in the System

The standard ServiceCenter system contains test data (users, devices, contacts, and so on) with which to work and learn the system. You can use these records as a model for your actual data. This guide uses the sample data to illustrate the utilities and processes described. You can modify or delete these records as you learn the system. You can also create new records.

Sample data includes:

- Users—a set of fictitious users with associated profiles, incident tickets and other records.
- Menus—various startup menus with different system rights. You may copy these for your use or modify them to fit the users on your system.
- Inventory—a simulated network inventory (modems, PCs, workstations, mainframe hosts, and so on).
- Contacts—a number of contact records. Each individual in the contacts file is associated with a certain device or primary asset.

Other Supporting Files—additional sample data in such files as location, company, model, and vendor. Information in these files is used throughout ServiceCenter.

Sample Screens and Examples

The sample screens and examples included in this guide are for illustration only and may differ from those at your site.

Documentation Web Site

For a complete listing of the current ServiceCenter documentation, see the Documentation pages on the Peregrine CenterPoint Web site at <http://support.peregrine.com/>.

You will need your current login and password to access this Web page.

For copies of the manuals, you can download .PDF files of the documentation using the Adobe Acrobat Reader (also available on the Web site). Additionally, you can order printed copies of the documentation through your Peregrine Systems sales representative.

Contacting Peregrine Systems

For further information and assistance with ServiceCenter in general, contact Peregrine's Customer Support.

Peregrine's CenterPoint Web Site

Current details of local support offices are available through Peregrine's CenterPoint Web site at <http://support.peregrine.com/>.

To find Peregrine Worldwide Contact Information:

- 1 Log on with your login User Name and Password.
- 2 Click **Go for CenterPoint**.
- 3 Select **Whom Do I Call?** in the navigation bar on the left side of the page. Peregrine worldwide information is displayed for all products.

Contacting Education Services

Training services are available for the full spectrum of Peregrine Products including ServiceCenter.

Current details of our training services are available through the following main contacts or at:

<http://www.peregrine.com/education>

Address:	Peregrine Systems, Inc. Attn: Education Services 3611 Valley Centre Drive San Diego, CA 92130
Telephone:	+1 (858) 794-5009
Fax:	+1 (858) 480-3928

1 Unload Script Utility

CHAPTER

The unload script utility allows you to create standard ServiceCenter unload files through an enhanced user interface. Unload files created with this utility have the following features:

- All related records are included in a single unload file.
- All components of an unload file are identified in a single unload script record.
- An administrator can modify and repeat the unload process as needed.
- Unload files are backwardly compatible with earlier releases of ServiceCenter.

Overview of the Unload Script Utility

The unload script utility is particularly useful for system upgrades. Customized features and files, such as forms, link records, or Format Control records, can be unloaded intact into the new system. Entire classes of records can be unloaded from a system and transferred to any other ServiceCenter system as a single file. You may unload the unload script record itself, thereby keeping the query list intact for further upgrades or data swapping.

The script you create to tie records together for an unload can also be used to purge those records from your system. You may specify in the unload script record those formats or subformats in your system you want to protect from purging or unloading.

Creating an Unload Script

To create an unload script:

- 1 Select the Toolkit tab in the system administrator's home menu.

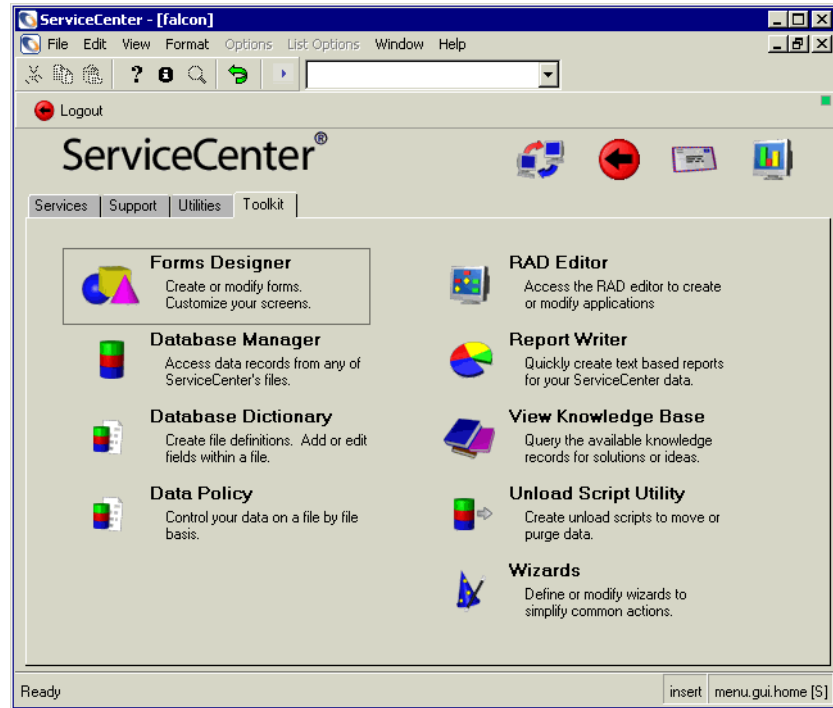


Figure 1-1: Toolkit tab in the system administrator's home menu

- 2 Click Unload Script Utility.

The Search Unload Records form appears.

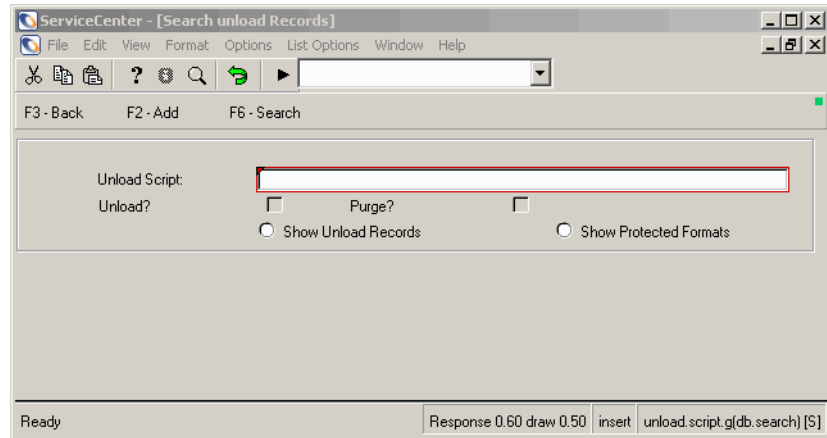


Figure 1-2: Search Unload Records form

3 Click Search.

A QBE list of existing unload script records is displayed.

Unload Name	Unload Records?	Purge Records?
Benchmarking Tool	true	false
CIB DOB	true	false
CIB Wizard	true	false
CM3 Category Unload	true	false
Enterprise Connections for PeopleSoft A990	true	false
External Database Mapping	true	false
Get.IT elements	true	false
ICM Event Services	true	false
Knowlix A9902	true	false
PM Category active flag	true	false
Popup Calendar	true	false
PurgeAll	false	true
Receipt	true	false
SAP HR	true	false
SAP MM/RM	true	false
SQL Upgrade Utility	true	false
Service Mgmt Event Services Support	true	false
Set Reminder	true	false
Topaz Purge Script	false	true
Topaz Purge Script (2)	false	true
Topaz Purge Script (3)	false	true
Topaz Purge Script (4)	false	true
Topaz Purge Script (5)	false	true
Topaz Purge Script - RAD	false	true
Unload Script Utility	true	false
WishDesk	true	false
gi Portal	true	false
insight	true	false
preupg.bin	true	false
upgdisp2.dta	true	false

Selected line is row 1 of 30 records Response 0.160 draw 0.80 insert unload.qbe.g(db.list) [US]

Figure 1-3: QBE List of Available Unload Script Records

4 Select Options>Add/Update.

A blank unload script record form is displayed.

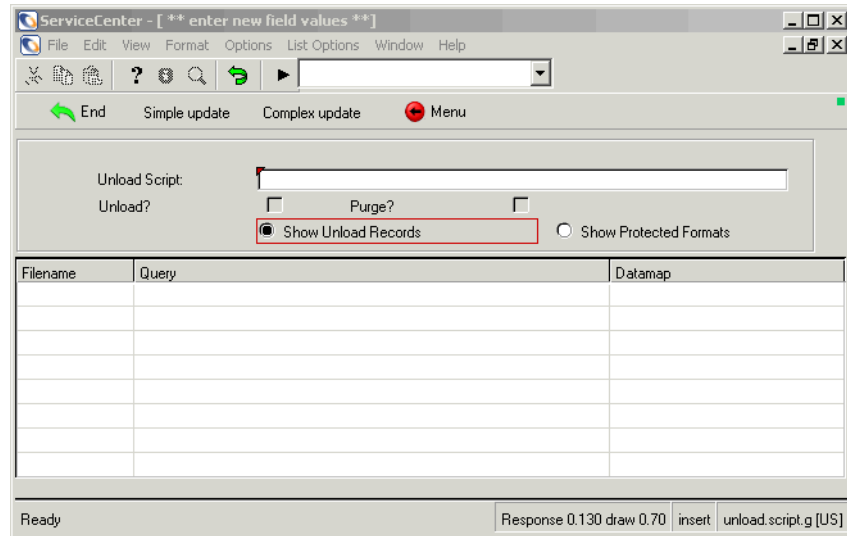


Figure 1-4: Blank Unload Script record

5 Complete the following fields:

Field	Description
Unload Script	Descriptive name of the script.
Unload?	Creates an unload file if checked
Purge?	Purges the records returned by your queries (or the data map) from your system if checked
Show Unload Records	Displays the list of files and the queries you have created to unload records
Show Protected Formats	Displays protected forms excluded from any unload. This is a user-defined list designed to protect important forms or subformats from being purged from your system.
Filename	Name of the ServiceCenter file from which the records are to be selected

Field	Description
Query	Query used to select the records from the file listed in the Filename field. This query selects records for the unload/purge based on values in a keyed field.
Datamap	Determines whether or not the system uses the datamap to unload related records. If this field evaluates to <i>false</i> , the utility unloads only those records selected by the query. If this field evaluates to <i>true</i> , the utility unloads related records as defined in the datamap file.

- 6 Enter a descriptive name for your script in the **Unload Script** field.
For this example, we have entered **Tech Manager**.
- 7 Select the **Unload** check box to unload the records to an external file.
- 8 Select the **Show Unload Records** option.
- 9 Select the names of all the ServiceCenter files from which you want to unload records from the drop-down list in the **Filename** column.
 - a Click in an empty field to display the drop-down list.
 - b Type the first letter of the file to advance the list to the proper section of the alphabet.
 - c Use the scroll bar to find and select the name.

For this example, we have selected the following files:

- format
 - menu
 - operator
 - pmenv
 - smenv
 - unload
- 10 Write the queries in the **Query** column needed to select the particular records you want to unload.
Use the following guidelines when writing a query:
 - Know the name of the field to use in your search.
 - You may use any operator (**# = <> ~**, etc.)
 - You may query against any field in a file; however, it is preferable to use a keyed field.

For this example, we have entered the following queries:

- `name#"menu.tech"`—unloads all forms beginning with *menu.tech*
 - `name="TECHMANAGER"`—unloads a menu record called TECHMANAGER.
 - `name#"tech"`—unloads all operator records of users from the operator file whose login name begins with *tech*.
 - `name#"tech"`—unloads all Incident Management profile records from the `pmenv` file for users whose login name begins with *tech*.
 - `name#"tech"`—unloads all Service Management profile records from the `smenv` file for users whose login name begins with *tech*.
 - `name="Tech Manager"`—unloads the script record we just created. This action (recommended) preserves your script record and the queries you have created for use in another system.
- 11 Enter either true or false in the Datamap column to direct the system to use or ignore the ServiceCenter datamap.
- a Place the cursor in the field.
 - b Type t for true or f for false.

The system fills in the remainder of the word.

Note: The data map establishes relationships between files that can be used to retrieve or purge related records, such as links and Format Control. Check the predefined relationships for the files you have selected in the `datamap` file before using this feature. See *Data Mapping* on page 55 for more information.

Your unload script record may look like the example in Figure 1-5 on page 20.

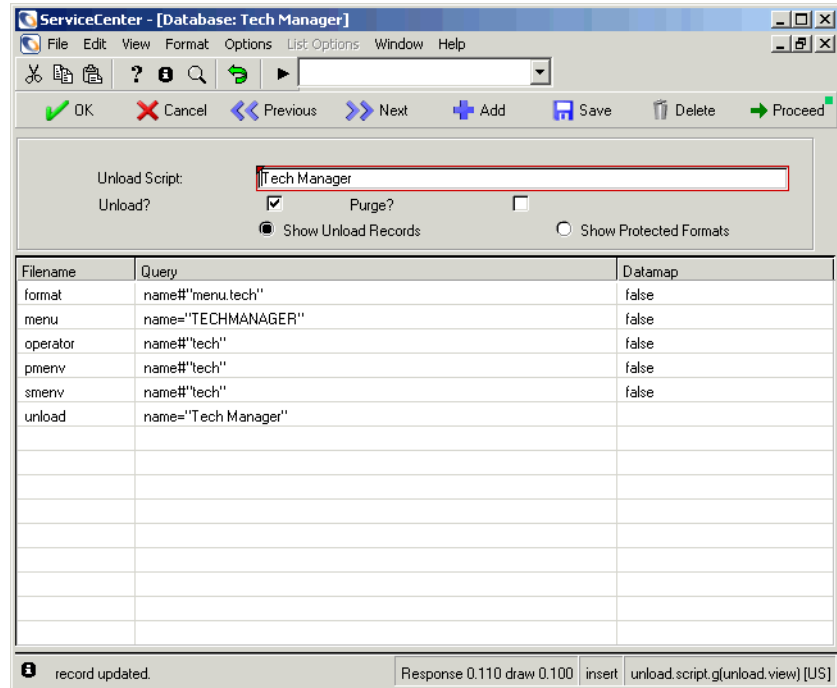


Figure 1-5: Unload Script Record—New

- 12 Click Add when you are finished to add the record to the database. The following message is displayed in the status bar: *unload record added.*

Commonly Scripted Files

The following files and corresponding key field names are commonly used when creating unload scripts:

File	Field
format	name
link	name
formatctrl	name
menu	name

File	Field
operator	name
probsummary	number
problem	number
incidents	incident.id
ocmq	number
ocml	number
ocmo	number
cm3r	number
cm3t	number

Protecting Formats

Warning: If you choose to purge any data, you must be sure to protect formats that might be required elsewhere in the system. This is particularly true when enlisting the **datamap** file to purge forms. Subformats used by forms are deleted automatically when related forms are purged.

Inspect the relationships established in the **datamap** file and enter the names of any such format in the **Protected Formats** list in your unload script record. See *Database Management and Administration* for more information.

Examples

New Categories

For this example, assume your system uses three categories for tickets in Incident Management:

- network.hardware
- network.software
- network.performance

As a result, your system would contain the following nine forms that would allow you to open, update, and close problem tickets:

- `problem.network.hardware.open`
- `problem.network.hardware.update`
- `problem.network.hardware.close`
- `problem.network.software.open`
- `problem.network.software.update`
- `problem.network.software.close`
- `problem.network.performance.open`
- `problem.network.performance.update`
- `problem.network.performance.close`

Perhaps you want to consolidate these three categories into a single category called **network**. For this new category, you need only the following three forms:

- `problem.network.open`
- `problem.network.update`
- `problem.network.close`

To purge only the old forms using the unload script utility, query for `name#"problem.network"` and specify the three new forms for open, update, and close in the **Protected Formats** list.

Moving Development Systems into Production

Your production system contains five forms whose names begin with *vendor*. You have modified all these forms in your development system and are ready to replace the old ones in your production system with the new vendor forms. In the process of developing the vendor forms, you created an experimental form called `vendor.new.g`, but you are not ready to move this form into the production system. To prevent this form from being unloaded with the others, query for `name#"vendor"` in the format file and list `vendor.new.g` in the **Protected Formats** list.

Protecting Subformats

You are consolidating device types and want to purge a particular device form, such as **device.Group.g**, from your system. You want to use the data map for the format file to purge link and Format Control records associated with this form. Subformats are mapped to their parent forms in the **datamap** file and are subject to the same conditions defined in your unload script record. (See *Data Mapping* on page 55 for more information.)

The **device.Group.g** form and many others in the **device** file use three important subformats called **location.asset**, **contacts.asset** and **vendor.asset** to link with the location, contacts, and vendor files respectively. A query for name#“device.Group” in the format file purges both the **.w** and **.g** forms of **device.Group** as well as any subformats used. To keep **location.asset**, **contacts.asset**, and **vendor.asset** from being purged, you must add them to the Protected Formats list in the unload script record.

Editing an Unload Script

To edit an unload script:

- 1 Select the Toolkit tab in the System Administrator's home menu (Figure 1-1 on page 14).
- 2 Click **Unload Script Utility**.

The **Unload Script** record is displayed.

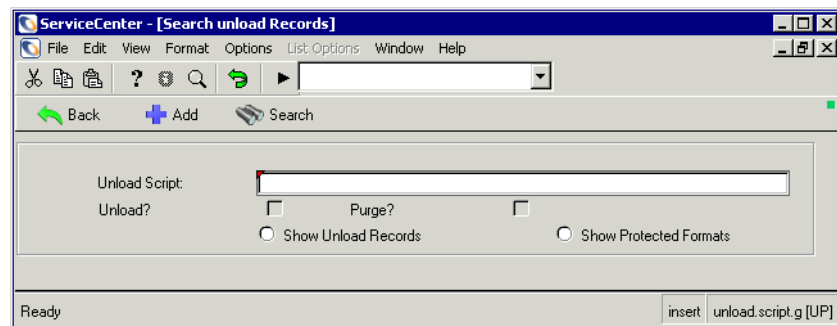


Figure 1-6: Unload Script record

- 3 Click **Search**.
- 4 Double-click the script record you want to edit.

The unload script record is displayed.

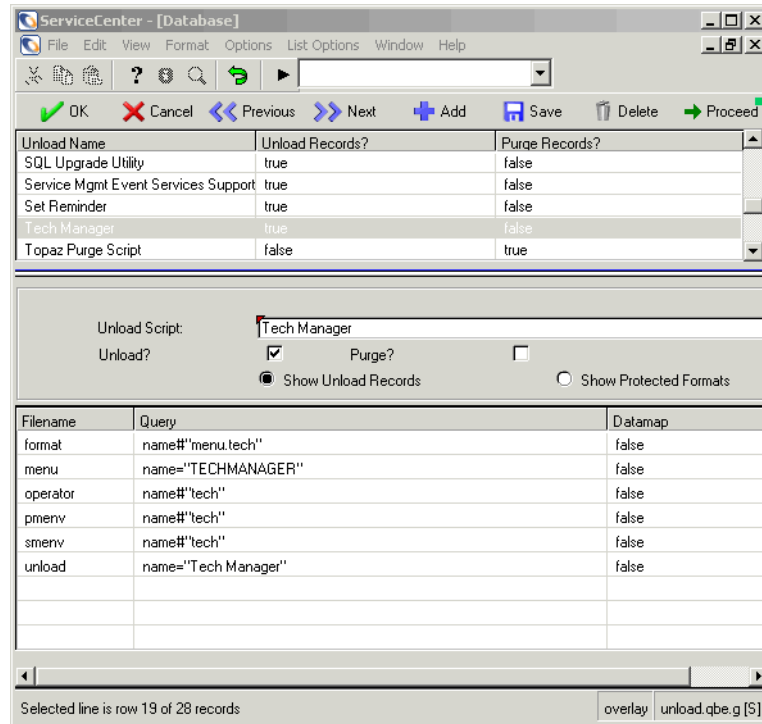


Figure 1-7: Unload Script record

- 5 Click **Edit**.
- The record is displayed in edit mode.
- 6 Edit the record.
 - 7 Click **Save** or **OK**.

The following message is displayed in the status bar: *Record updated in the unload file.*

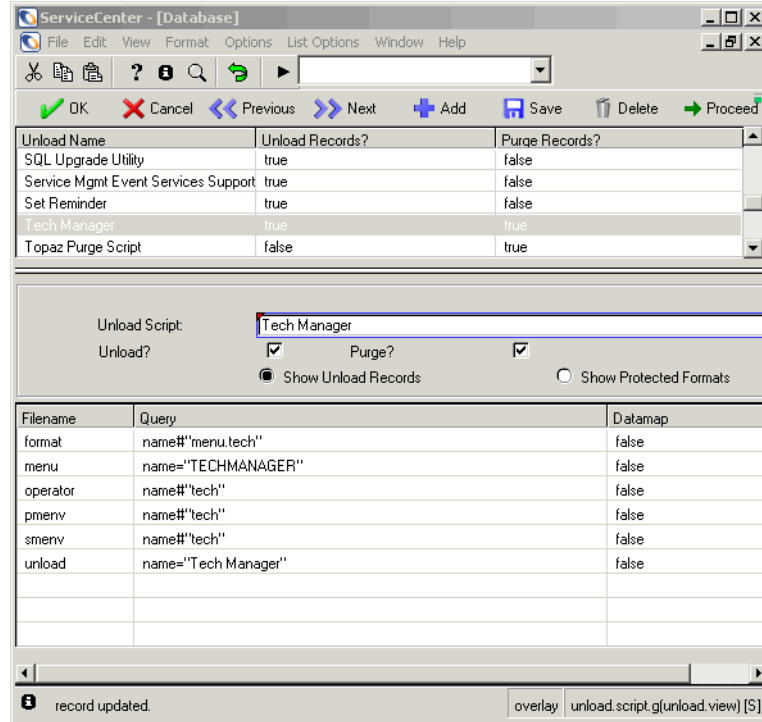


Figure 1-8: Unload Script record—edit mode

Validating an Unload Script

To validate an unload script:

- 1 Select the Toolkit tab in the system administrator's home menu (Figure 1-1 on page 14).
- 2 Click Unload Script Utility.

The Unload Script record is displayed.

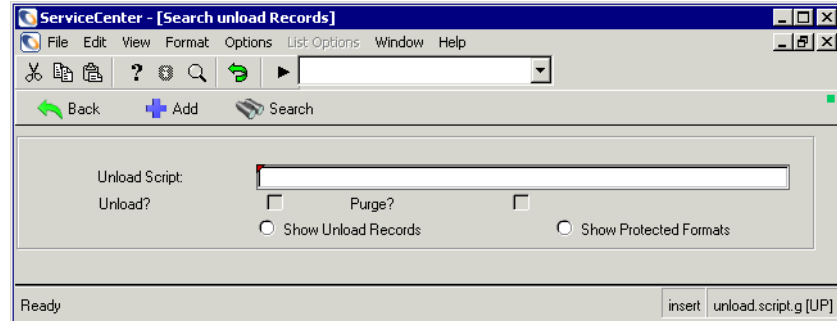


Figure 1-9: Unload Script record

- 3 Click Search.
- 4 Double-click on the script record you want to validate.
The unload script record is displayed (Figure 1-10 on page 27).

5 From the Options menu, choose Validate.

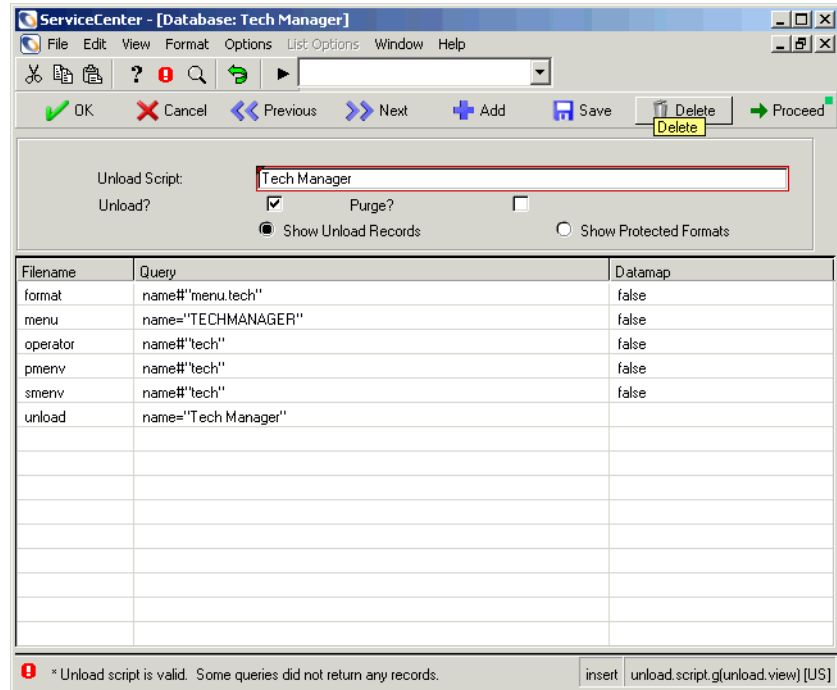


Figure 1-10: Unload Script record—validating the scripts

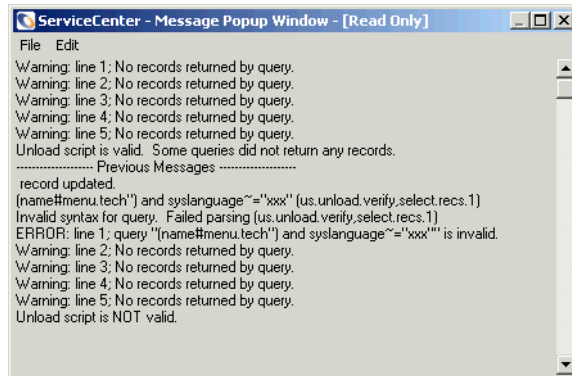
If the script is valid, the following message is displayed in the status bar:
Unload script is valid.

If the script fails, the following message is displayed in the status bar: *Unload script is invalid.* See *If the Validity Check Fails* on page 28.

If the Validity Check Fails



If the validity check fails, an error message providing details is displayed in the status bar. Click the message icon that is displayed to see the entire text in a pop-up window.



Warning: A query can be valid and not unload the desired set of records. Make sure your query defines the exact records you want to unload, particularly when using the *purge* feature.

Unloading Records from an Unload Script

There are two methods for unloading files from an unload script:

- Peregrine Unload/Export Facility—accessed from the Options menu of an unload script record in the edit mode (refer to *Unloading ServiceCenter Data* on page 93).
- ServiceCenter Unload utility—described in the following section.

To unload files from an unload script:

- 1 Select the Toolkit tab in the system administrator's home menu (Figure 1-1 on page 14).
- 2 Click **Unload Script Utility**.

The Unload Script record is displayed.

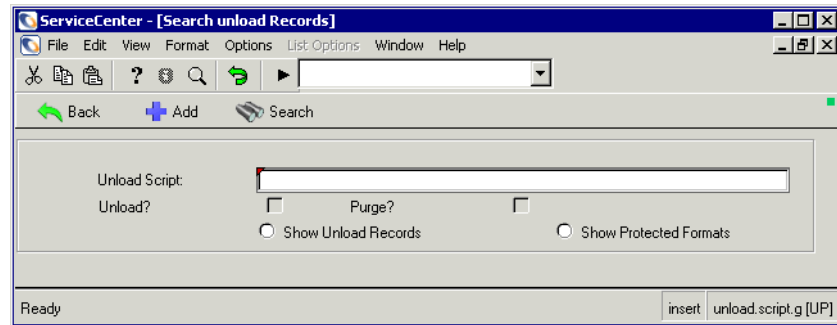


Figure 1-11: Unload Script record

- 3 Click **Search**.
- 4 Double-click on the script record you want to unload.
The unload script record is displayed (Figure 1-10 on page 27).
- 5 Click **Proceed**.
A blank unload utility form is displayed.
- 6 Enter the name and path for your unload file (Figure 1-12 on page 30).

You may specify the fully qualified path to a particular directory, or none at all. If no path is defined, the system places your unload file in the Run directory of the Server or in the Run directory of the client. It is not necessary to add a file extension, but may be useful when identifying the file later.

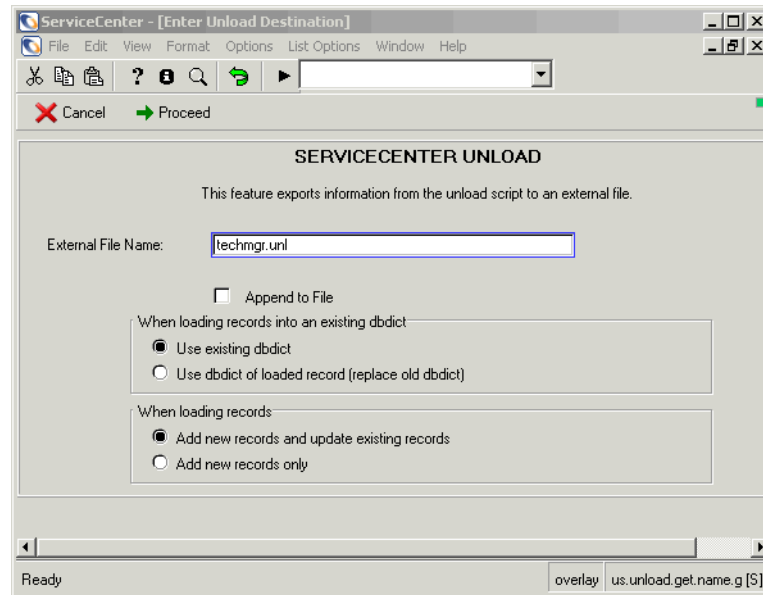


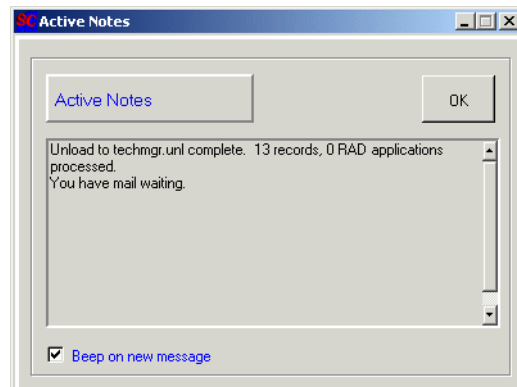
Figure 1-12: Unload utility form

7 Select one of the following options:

Field	Description
Append to File	Allows you to append new data to existing unload files. If this field is checked (<i>true</i>), you may add records to a previously created unload file. If this field is blank (<i>false</i>), a new unload file with the same name will overwrite an existing unload file.
Use existing dbdict	Unloaded records are added using the existing Database Dictionary. Any changes to fields or keys are ignored.
Use dbdict of loaded record (replace old dbdict)	Database Dictionary of the files being unloaded replaces the Database Dictionary of the corresponding files in the new system. This allows you to unload updated files as well.
Add new records and update existing records	Adds new records to the files of the new system and updates existing records with new information from records in the unload file.
Add new records only	Adds new records to the files of the new system only. Does not replace existing records.

8 Click **Proceed** to unload the file.

You are returned to the Toolkit tab. If Active Notes is enabled, an unload confirmation message is displayed.



Purging Records from an Unload Script

You can use the unload script utility to purge (delete) selected records from your system. You have the option to unload the specified records at the same time, or simply to remove the records from your system permanently.

Warning: Purging records can result in the inadvertent loss of records because of an incorrectly worded query. To prevent this, perform an unload at the same time and check the results of the purge. If you have purged the wrong records, you can restore the data from the unload file and edit the query accordingly.

To purge records with the unload script utility:

- 1 Open the desired unload script record in the edit mode as described in *Editing an Unload Script* on page 23.
- 2 Select the **Purge?** check box.

In this mode you can either display the list of unload records you have defined or view the list of protected forms.

- 3 Click **Show Protected Formats** to display the list of protected forms for this unload script.

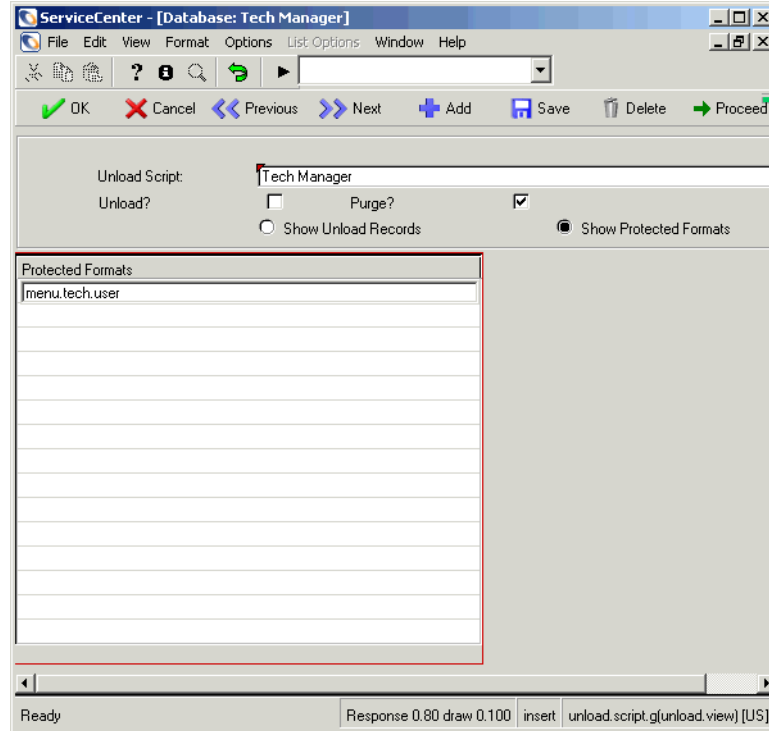


Figure 1-13: Unload Script record with protected format view

- 4 Enter the names of any forms you want to protect in the **Protected Formats** field.
- 5 Click **Add** to save your changes.
- 6 Click **Search**.
- 7 Click **Proceed**.
A prompt is displayed asking you to confirm the purge.
- 8 Click **Yes** to purge/unload the records.
Click **No** to abort the process and return to the unload script record.

2 Purge/Archive

CHAPTER

This chapter introduces you to Purge/Archive. The following topics are discussed in this chapter.

- Access Purge/Archive.
- Archive records in foreground.
- Purge records in foreground.
- Purge/archive records in foreground.
- Schedule purge, archive, or purge/archive processing in background.
- Use Purge/Archive with Network Management Applications.
- Make on-line copies of files.

Understanding the Purge/Archive Function

The Purge and Archive functions allow you to manage your database size by deleting and/or storing multiple records in an external file. You can run purge and archive in the foreground or schedule them to run at prescribed intervals. However, we suggest that you archive records before purging them. This provides you with a recovery point, if needed. Also, due to the input/output time involved with this process, we suggest scheduling purges and archives during non-peak hours. The *purge* and *archive* options are located on the System Administration Maintenance menu.

Note: An archive record will retain the structure of the DBdict table at the time of the unload. If fields are added within the structure of the table and not at the end, loading an archived record will cause these restored records to have their field indexing off.

Using the purge/archive function with other database files is similar to performing a mass unload and a mass delete in the Database Manager with the exception that a single key can be used to purge the records from a ServiceCenter file, immediately following the archive, and you can schedule the purge, archive, or purge/archive processing.

An important feature of Purge/Archive is its ability to process **data maps**. This is especially useful when purging or archiving records from one of the Network Management Applications (such as, Incident Management, Change Management, or Request Management). These applications employ complex data relationships among potentially multiple records and multiple files. Purging a Request Management order, for example, entails purging the order and all associated alert logs, approval logs, line items, line item pages, order pages, and phase logs. See *Data Mapping* on page 55 for more information.

How to Access the Purge/Archive Function

There are two ways to access the **Purge/Archive functions within ServiceCenter**:

- Via the **menu option**.
- Via **Command line** shorthand notations from command prompt.

Command Line Access:

- ▶ Enter *purgarch* on any menu command line.
- or-

Menu Access:

- 1 On the Main Menu, select the Utilities tab.
- 2 Click **Maintenance**.
- 3 Click **Purge/Archive**.

The pa.prompt.1 form is displayed.

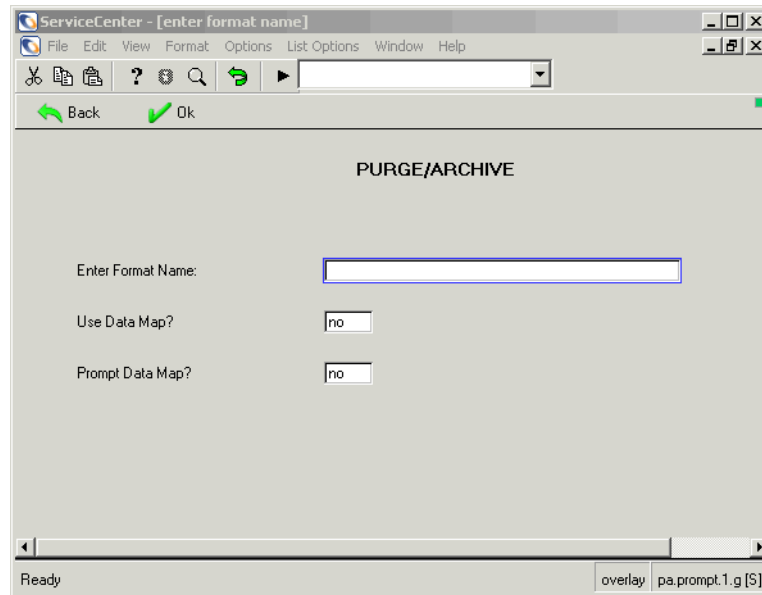


Figure 2-1: The Purge/Archive function

- 4 Enter field values to begin the Purge/Archive process.

Field	Description
Enter Format Name	Enter the format name of the file whose files you want to archive or purge. Press Enter. Note: The format you select must be associated with a file.
Use Data Map?	Default is no. If yes, Data Map will execute for every record that is unloaded from the primary file. Refer to <i>Using Purge/Archive with Network Management Applications</i> on page 52, or refer to <i>Data Mapping</i> on page 55.
Prompt Data Map?	Default is no. If yes, and Use Data Map is yes, the Data Map application will display the standard Data Map confirmation screen for every record that is unloaded from the primary file. The use of data mapping is optional, but highly recommended when purging or archiving records from any of the Network Management Applications. See the <i>Database Management and Administration Guide</i> for more information.

- 5 When you have completed entering values in these fields, press Enter.

See *How to Archive Records in the Foreground* on page 38 to continue with the process.

How to Archive Records in the Foreground

The *archive* process unloads selected records to a file external to ServiceCenter. Here are a few points to remember:

- The external file may be a member of a partitioned data set (OS/390).
- It is not necessary to pre-allocate the external file unless required by your specific installation. If the external file is pre-allocated, it *must* have specific data control block parameters (OS/390):

```
dcb=(blksize=4100, lrecl=4096, recfm=vb)
```

- Once the QBE list of selected records is displayed, you cannot select specific records for purge or archive.
- Records are not automatically purged after archive. You must use the **purge** or **purge/archive** options to purge records.

The following screens illustrate how to archive all *enduser* records (form name: *endusr.g*) with a *logical.name* that begin with a *logical.name* that begins with *a108*.

The screenshot shows the ServiceCenter application window titled "ServiceCenter - [enter search argument]". The menu bar includes File, Edit, View, Format, Options, List Options, Window, and Help. The toolbar contains icons for Cut, Copy, Paste, Help, Find, and Refresh. Below the toolbar are buttons for End, Query, and Menu. The main window displays the "ENDUSER DEVICES" form with the following fields:

- Logical Name:
- Parent:
- Location:
- Manufacturer:
- Customer No:
- Polling Address:
- Local Software:
- Primary Contact:
- Alternate Contact:
- Feature No:
- Comments:
- Availability Information:
 - Number of Incidents:
 - Explicit Unavailability:
 - Implicit Unavailability:
 - Perceived Unavailability:
 - Operating Hours: Start:
 - End:
- Graph of Unavailability: Date Opn m

Other fields include Serial No., Model, Fixed Asset No., Type, VTAM Name, VTAM Parent, Vendor, Modem Name, Remote Software, Controlling Software, Primary Phone, and Description. The status bar at the bottom shows "Ready" and "insert enduser.g [P]".

Figure 2-2: Archiving records in the foreground

- 1 Enter the field values that will select the records you want to archive and press **Enter**. For this example, enter *a108* in the **Logical Name** field.

The *enduser.qbe* list form will display. In this example, the screen displays a list of all the records with a *logical.name* that begins with *a108* in the *enduser* file.

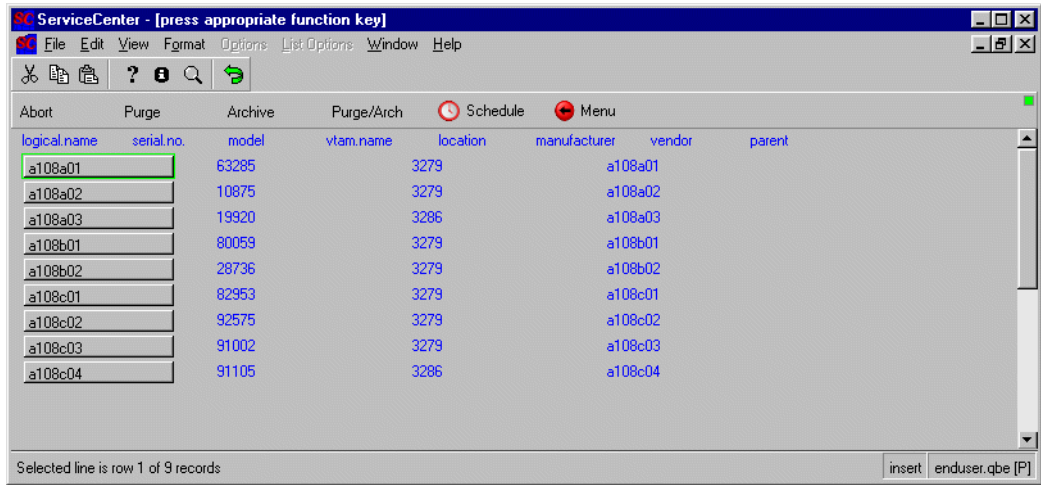


Figure 2-3: Displaying records

- 2 Click **Archive** to archive the displayed records to an external file.

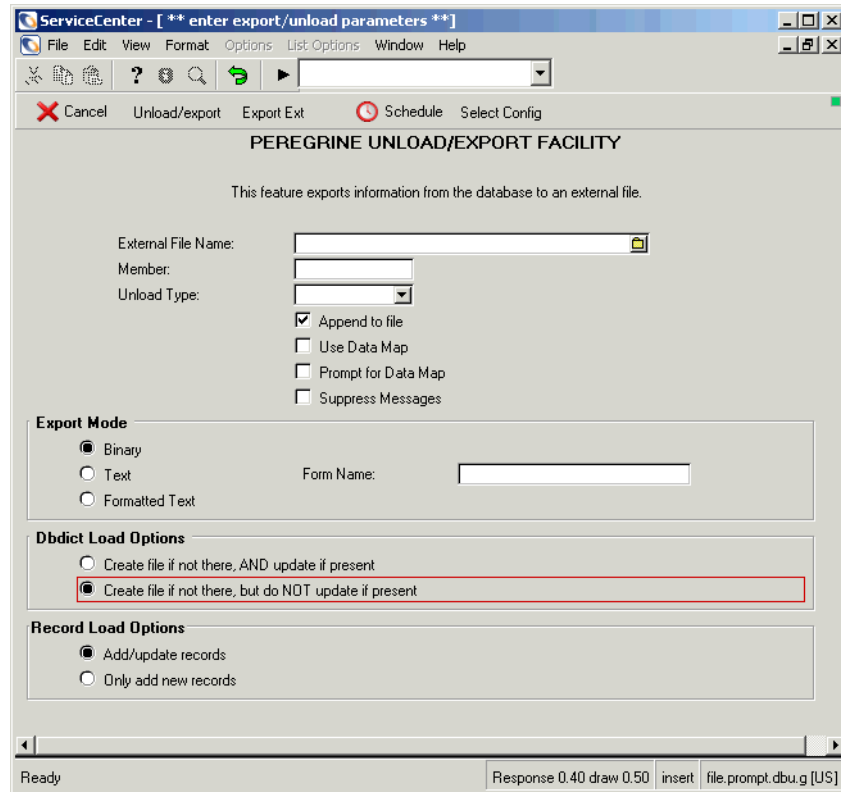


Figure 2-4: Archiving records to an external file

- 3 Enter the field values, as described below. Using the tab key, position the cursor at each field and enter the required data.

-or-

Click **Query** to enter a query to select the records to be archived.

Field	Description
External File Name	<p>Enter the fully qualified name of the external file to which you want the information written.</p> <p>Note: In the OS/390 environment, you must enter an existing high level qualifier or the process will wait on execution for action at the system console.</p> <p>Note: In the Unix environment, enter the full pathname if the file is being written to a directory other than the one in which the ServiceCenter files are located.</p> <p>Note: If pre-allocated (OS/390), the data control block must be:</p> <p><code>dcb=(blksize=4100,lrecl=4096,recfm=vb)</code></p>
Member	(OS/390 only) If the external file is a partitioned data set, enter the name of the member. Otherwise, leave the field blank.
Unload Type	If the external file is a QSAM data set, select OS/390. Otherwise, select unix or winnt for a file.
Append to File	<p>The default is true (checked), which indicates that the archived information will be appended to any existing file of the same name. If a file of the same name does not exist, one will be created.</p> <p>Note: If the field is set to <i>false (unchecked)</i>, any existing file of the same name will be replaced with the archived information or, if none exists, a file of that name is created.</p> <p>Note: This option is not applicable to members of a partitioned data set (PDS). In the OS/390 environment, files cannot be appended to PDS members. When a member is defined, the application <i>always</i> creates a new data set without respect to the parameter entered.</p>
Use Data Map	Default is <i>false</i> . If this box is checked (<i>true</i>), Datamap executes for every record that is unloaded. Refer to the Data Mapping on page 55 for information on data mapping.
Prompt Data Map	Default is <i>false</i> . If this box is checked (<i>true</i>), and the Use Data Map field is <i>true</i> , the Datamap application displays the standard Datamap confirmation screen for every record that is unloaded.

Field	Description
Suppress Messages	Allows you to ignore all messages produced during the unload process, including the final confirmation message.
Export Mode	<ul style="list-style-type: none"> ■ Binary—writes the file in ServiceCenter internal binary formatted mode, which can only be read by the Database Manager's Load facility. ■ Text—Writes the file in ServiceCenter internal text mode. Exported text can be read by humans, unlike Binary mode. However, it is not formatted for data transfer to external applications. It is only readable by another ServiceCenter system, using the Database Manager's Load facility. ■ Formatted Text—Exports information in a format that can be read by external programs, and structured to match an existing ServiceCenter text (not Windows) format. ■ Form Name—name of the format that describes the information to be exported via the Formatted Text Unload/export system tray option. If the format does not exist, you will be prompted to create it.
Dbdict Load Options	<p>When you export a dbdict, you also set the actions for <i>import</i>. The following fields determine those actions. Select one of the following, based on your particular needs:</p> <ul style="list-style-type: none"> ■ Create file if not there, but do NOT update if present ■ Create file if not there, AND update if present
Record Load Options	<p>When you export a record, you also set the actions for <i>import</i>. The following fields determine those actions. Select one of the following, based on your particular needs.</p> <ul style="list-style-type: none"> ■ Add/update records ■ Only add new records

- 4 When you have completed entering data in the fields, click **Unload/export** to begin the archive process. This process runs as a foreground task, taking control of your terminal session until the process ends.

Note: A search window displays if no records are immediately found or does not initially satisfy the QBE search.

Click **Search** to perform a *true* query. Make your selection by clicking on a record.

Note: Press **Cancel** to end the archive process.

When the archive is complete, a message is issued indicating that the file has been unloaded to an external file:

*<#> records from enduser unloaded to: a108, 00:00:00

Note: If the message contains an asterisk (*) in the far left column, be sure to scroll the remaining messages.

If you have not preallocated the external file, the message: *Unable to allocate (file.unload.connect)* is displayed. This *does not* mean that the records have not been archived.

If you use *Data Map* to archive records, disregard the message, as the count will not reflect the correct total number of records archived. See *Data Mapping* on page 55 for information.

How to Purge Records in the Foreground

The **purge** option allows you to delete (purge) records from a ServiceCenter file. This process runs as a foreground task, taking control of your terminal session until the process ends.

To purge records in the foreground:

- 1 Repeat the process to access the Purge/Archive function by following the detailed instructions in *How to Access the Purge/Archive Function* on page 36.
- 2 Enter the same field values as discussed earlier in *How to Archive Records in the Foreground* on page 38. The fields explanations on page 42 will give you details on the values needed for each field.

-or-

Click **Query** to enter a query to select the records to be purged.

- 3 When you have completed entering data in the fields, press **Enter** to begin the purge process. This process runs as a foreground task, taking control of your terminal session until the process ends.
- 4 Click **Abort** to abort the purge process.

-or-

Click **Purge** to purge the selected records.

Warning: If you choose to purge the selected records, you will be prompted to select *Abort Purge* or *Purge All* records. Make sure the records you have selected are the records that need to be purged.

If the selected records should not be purged, click **Abort Purge**.

If the selected records should be purged, click **Purge All** to confirm purging the displayed records from the database.

Once the records have been purged, a message indicating how many records were purged is displayed on your screen.

*<#> records purged from the enduser file.

Note: If the message contains an asterisk (*) in the far left column, be sure to scroll the remaining messages.

If you have not preallocated the external file, the message: *Unable to allocate (file.unload.connect) is displayed*. This *does not* mean that the records have not been archived.

If you use *Data Map* to archive records, disregard the message, as the count will not reflect the correct total number of records archived. See [Data Mapping](#) on page 55 for information.

- 5 Click **End** to return to the main menu.

How to Purge/Archive in the Foreground

The purge/archive option allows you to archive records and purge the same records from your database.

- 1 See the step-by-step instructions and rules presented in *How to Archive Records in the Foreground* on page 38 and *How to Purge Records in the Foreground* on page 44 for purge/archive process.
- 2 Click **Purge/Arch** to unload and export (archive) the displayed records to an external file.

Once the unload/export is complete, a message indicating that the file has been unloaded to an external file is displayed on your screen.

*<#> records from enduser unloaded to: a108, 00:00:00

Note: If the message contains an asterisk (*) in the far left column, be sure to scroll the remaining messages.

If you have not preallocated the external file, the message: *Unable to allocate (file.unload.connect) is displayed*. This *does not* mean that the records have not been archived.

- 3 Click **Purge All** to confirm the purge process in the foreground.

Once the purge/archive is complete, a message indicating how many records were purged is displayed on your screen.

*<#> records purged from the enduser file.

Note: If the message contains an asterisk (*) in the far left column, be sure to scroll the remaining messages.

If you have not preallocated the external file, the message: *Unable to allocate (file.unload.connect) is displayed*. This *does not* mean that the records have not been archived.

If you use *Data Map* to archive and purge records, disregard the message, as the count will not reflect the correct total number of records purged. See *Data Mapping* on page 55 for information.

- 4 Click **End** to return to the main menu.

How to Schedule Purge/Archive in the Background

The schedule option allows you to purge, archive, or purge/archive selected records and schedule them to run at selected intervals as a background operation.

The following screens illustrate how to archive and purge all records with a model that begins with 3279 from the *enduser* file on a monthly basis.

To schedule selected records to be purged, archived, or purged/archived as a background process:

- 1 Repeat the process to access the Purge/Archive function by following the detailed instructions in *How to Access the Purge/Archive Function* on page 36.
- 2 Enter the same field values as discussed earlier in *How to Archive Records in the Foreground* on page 38. The field explanations on page 42 will give you details on the values needed for each field.

-or-

Click **Query** to enter a query to select the records to be purged, archived, or purged/archived.

- 3 When you have completed entering the field values to schedule the records to be purged, archived, or purged/archived in the background, the records are displayed.

See the following example, which displays a list of all the records with a *model* that begins with 3279 in the *enduser* file.

Click **Schedule** to enter the scheduling information.

Note: If you plan to schedule an archive, be sure to click **Archive** to set up the external file name to where the files will be archived. Otherwise, the scheduling setup will fail because there is not an external file format named.

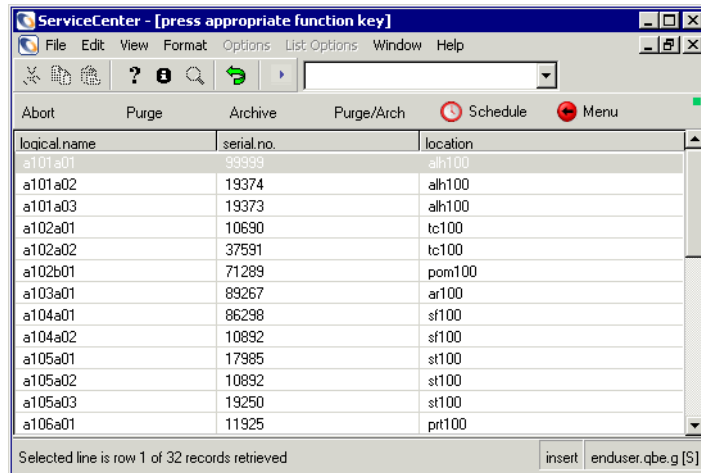


Figure 2-5: Selecting records to be archived, purged, or archived/purged

- 4 See the field descriptions below to enter the required data to set up scheduling for the archive, purge, or purge/archive background processing.

The screenshot shows the 'ServiceCenter - [Search typecheck Records]' window. The 'Date/Time to Run' field is set to '04/22/03 10:00:00'. The 'Repeat Interval' section includes a text input for 'Enter Number of Days and Time' with an example '(Example: 7 00:00:00 would be a 7 day interval)'. Below this are radio buttons for 'Monthly', 'Quarterly', 'Semi-Annually', and 'Annually'. The 'Archive Only' section has fields for 'Ext File', 'Member', and a dropdown for 'Append to existing file or create new file (append/create)' set to 'Append'. It also has checkboxes for 'Use Data Map?' and 'Suppress Messages?' both set to 'no'. The 'Dbdict load options include' section has a dropdown set to 'Create'. The 'Record load options include' section has a dropdown set to 'Add/Update'. The status bar at the bottom shows 'Ready', 'Response 0.30 draw 0.60 insert pa.schedule.prompt.gldb.search) [US]'

Figure 2-6: Scheduling records to be archived, purged, or archived/purged

Field	Description
Date/Time To Run	This defaults to the current date and time. You can change this value to the specific date/time you want the process to begin.
Repeat Interval	Use the repeat interval if you want the process to repeat on a regular basis.
Enter Number Of Days and Time	Enter the number of days and time (ddd hh:mm:ss) you want the process to repeat.
Monthly, Quarterly, Semi-annually, Annually	Place an x next to the interval you want this process repeated. If you entered a value for the repeat interval, leave these fields blank.

Field	Description
Archive Only	Enter the following fields only if you are using the Archive or Purge/Arch options. Note: If you are using the Purge option, leave the <i>Archive Only</i> fields blank.
Ext. File	Enter the name you want to call the flat, sequential QSAM file (OS/390) or Unix file. If pre-allocated (OS/390), the data control block must be: dcb=(blksize=4100,lrecl=4096,recfm=vb)
Member	(OS/390 only) If the external file is a partitioned data set, enter the member name of the external file. Otherwise, leave it blank.
Append To Existing File Or Create New File (Append/Create)	The default append allows you to append to records that already reside in the external file. Use create only when you want to replace records that currently reside in the external file. Note: This option is not applicable to members of a partitioned data set (PDS). In the OS/390 environment, files cannot be appended to PDS members. Unloading to a PDS member always creates a new member.
Use Data Map?	Default is no. If yes, Data Map will execute for every record that is unloaded from the primary file. Refer to <i>Using Purge/Archive with Network Management Applications</i> on page 52, or refer to <i>Data Mapping</i> on page 55 for information.
Suppress Messages?	Default is no. If yes, messages are not displayed in your status bar when the background processes are complete.
Dbdict Load Options	Use the default value create.
Record Load Options	Use the default value update.

- 5 When you are finished entering data in the required fields, proceed with one of the following steps:
 - Click **Purge** to begin the scheduled background process to purge records.
 - Click **Archive** to begin the scheduled background process to archive records.
 - Click **Purge/Arch** to begin the scheduled background process to begin the purge/archive records.
 - Click **Abort** to cancel the scheduled process.

The screenshot shows a window titled "ServiceCenter - [enter search argument]". The menu bar includes File, Edit, View, Format, Options, List Options, Window, and Help. The toolbar contains icons for End, Query, and Menu. The main area is titled "ENDUSER DEVICES" and contains the following fields:

- Logical Name:
- Serial No.:
- Model:
- Parent:
- Fixed Asset No.:
- Type:
- Location:
- VTAM Name:
- Manufacturer:
- VTAM Parent:
- Customer No.:
- Vendor:
- Polling Address:
- Modem Name:
- Local Software:
- Remote Software:
- Controlling Software:
- Primary Contact:
- Primary Phone:
- Alternate Contact:
- Phone:
- Feature No:
- Description:
- Vendor:
- Comments:
- Availability Information:
 - Number of Incidents:
 - Explicit Unavailability:
 - Implicit Unavailability:
 - Perceived Unavailability:
 - Operating Hours: Start:
 - End:
- Graph of Unavailability:
 - Date: 0am 4 8 0pm 4 8 m

The status bar at the bottom shows "Purge/Archive scheduled to run at 09/30/01 10:15:49" and "insert enduser.g [P]".

Figure 2-7: Scheduling a Purge/Archive for future background processing

A message indicating the scheduled time the background process should run is displayed in the status bar.

Note: If the message contains an asterisk (*) in the far left column, be sure to scroll the remaining messages.

- 6 Click **End** to return to the Purge/Archive screen (*pa.prompt.lpn* form).

Using Purge/Archive with Network Management Applications

The Network Management Applications (Incident Management, Change Management, and Request Management) employ complex data relationships among potentially multiple records and multiple files. For example, a Request Management order may consist of alert logs, approval logs, line items, line item pages, order pages, and phase logs.

In order to preserve the integrity of such data relationships during purging or archiving, all directly-related records must be purged or archived together. The feature that allows Purge/Archive to process related data components as a unit is called **data mapping**. Refer to *Data Mapping* on page 55 for details on how to access and create data maps.

Pre-built data maps are already in place for selected Network Management Applications. Take a moment to review and understand the data maps for the databases you plan to purge or archive. The pre-built data maps were created expressly for situations requiring preservation of data relationships, especially purging and archiving. Understanding how an application's records are data mapped is the key to performing successful purges and archives.

The following table summarizes the data relationships for selected databases.

Application	Purge/Archive This Database...	To Include Data Mapped Records from...
Incident Management	probsummary	problem, screlation
Change Management	cm3r	cm3t, screlation
Request Management	ocmo	ocmalertlog, ocmapprlog, ocml, ocmlpage, ocmopage, ocmphaseolog
Request Management	ocmq	ocmalertlog, ocmapprlog, ocml, ocmlpage, ocmphaseolog, ocmqpage

You may also want to purge/archive data mapped records from *work* and *SYSBLOB* for the preceding applications.

Note: Data mapping is not an automatic feature of purge/archive. You have the opportunity to enable data mapping when you submit or schedule a purge/archive; otherwise the default setting for data mapping is disabled. See [Data Mapping](#) on page 55 for information.

For example: If you submit an archive of the *probsummary* database without enabling data mapping, only *probsummary* records will be archived. Submitting the same archive with data mapping enabled will archive *probsummary* records with their related *problem* and *screlation* records.

Copy Database Records to Provide On-Line Archival Copies

Using the Database Dictionary **Copy/Rename** option allows you to selectively copy information from one ServiceCenter file to another. The primary uses for this functionality are to provide on-line archival copies of selected ServiceCenter data, and to copy entire files to new files with different database structures.

In Incident Management, once problems are resolved and have been inactive for some period of time, you can remove them from the primary file.

For example: You may want to keep individual files for each category or for each quarter of a year. These *on-line archive files* are available for queries (including IR queries) and reporting.

If you decide to restructure one of your existing files, removing unused fields or eliminating unnecessary ones, you can design the new database structure and use the **Copy/Rename** option to selectively move data from the old database file to the new one.

To access the Database Dictionary **Copy/Rename** option, see *ServiceCenter System Tailoring*.

3 Data Mapping

CHAPTER

The Data Map utility allows you to automatically discover and establish a relationship from a primary record to a secondary record. For instance, for each report, discover its associated RAD and formats, as well as the Format Control associated with those formats. Data Map establishes entity relationships at a generic field definition level. This allows you to apply the same Data Map to any record from a primary file. The actual entity relationship is determined at run time based on the contents of the input fields defined in the Data Map. You can define Data Maps for any group of associated ServiceCenter records. For example, you can map a CM3 Phase definition, a RAD program, or a Financial Inventory record.

The standard ServiceCenter Unload and Purge/Archive utilities use the Data Map. Data Map builds a series of arrays (referred to as Map Lists) that are returned to and used by these applications. This is explained in detail in the following sections.

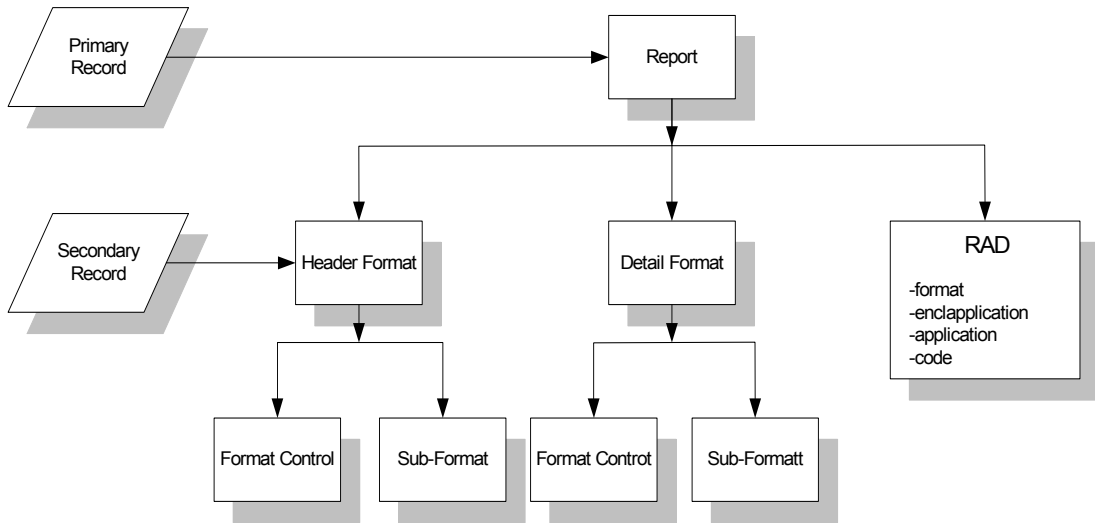


Figure 3-1: Hierarchical relationship of records associated with a standard ServiceCenter report

Data Map lets you define a generic map of this record hierarchy which is translated to a literal map at run time. You can apply this generic map to any ServiceCenter report.

Conceptually, each primary record points to secondary records. In the report example, the report itself is the primary record; the formats and RAD are the secondary records. When the Data Map builds the literal map, the secondary records then become primary records and the records related to them are secondary records. For instance, in the previous report example, the report definition itself is the original primary record, the formats are the secondary records. When the format record is processed, it becomes the primary record and the Format Control record is secondary to it. If we were to then find records related to the Format Control record, it would become the primary record and the records related to it become secondary records. This process can continue for any number of levels.

In order to avoid a looping condition, Data Map checks each entry added to the Map List based on *filename* and *query*. If the entry already exists, it is not added again and that record relationship path build is interrupted. Also, before adding an entry to the list, a query is made against the appropriate file with the indicated query. If no records are found, the record is not added to the list.

If you are working in foreground, Data Map optionally displays a confirmation screen that shows the Map List (that is, the name of each file and the query used to extract records from that file). You can delete, modify and add Map List entries as necessary. After selecting the proceed option, Data Map returns control to the calling application. If you are processing a list of records, Data Map displays the Map List for each record in the list. This process is explained on the following pages.

Note that the Map Lists are not saved in any ServiceCenter files. They are created on an as-needed basis for each record processed. Once the calling application is finished with the Map List, it is discarded.

Accessing the *Data Map* File

You may access the `datamap` file using either of two methods:

- Command line
- Menu options

To access the Data Map from the command line:

- 1 Make sure that the command line is visible by selecting **Command Line** from the **View** menu.
- 2 Type `DataMap` on the command line.
- 3 Click the **Start Command** arrow or press **Enter**.

To access the Data Map from a menu:

- 1 Select the Utilities tab in the System Administrator's home menu.

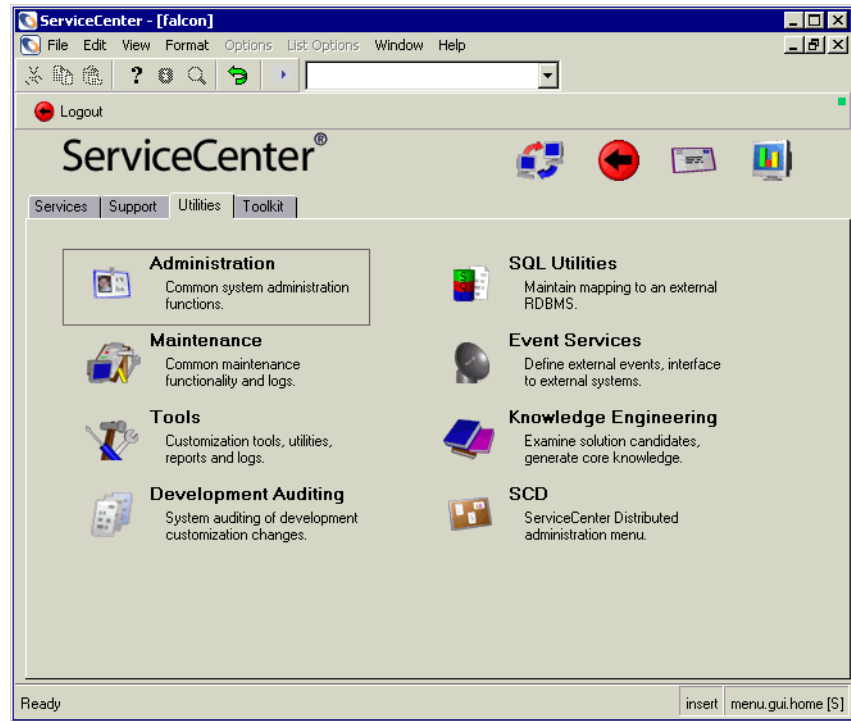


Figure 3-2: Utilities tab in the system administrator's home menu

- 2 Click Tools.

The Tools menu is displayed.

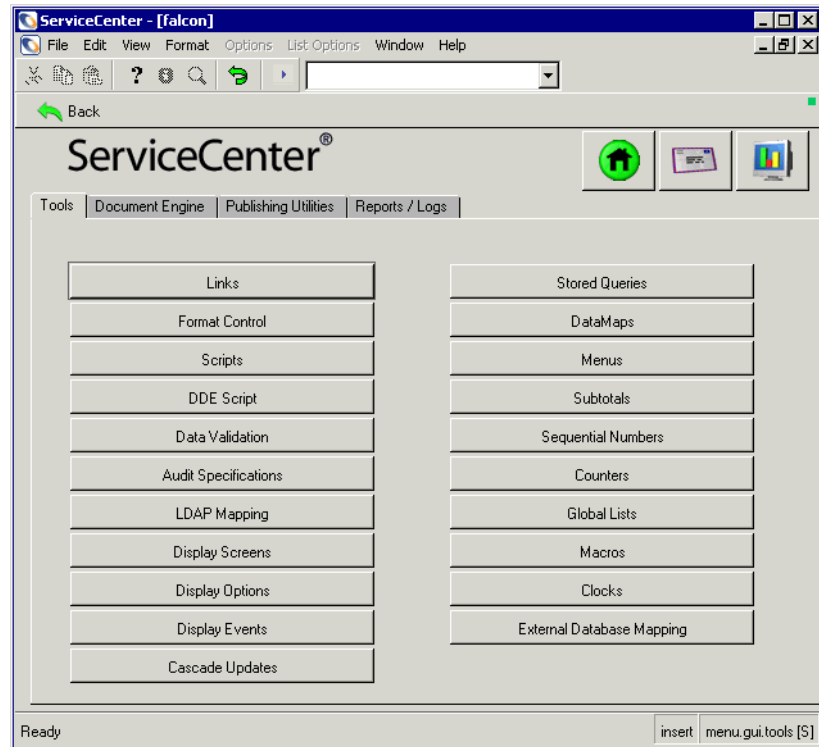


Figure 3-3: Tools menu in the system administrator's home menu

3 Click Data Maps.

A blank Data Map record is displayed.

The screenshot shows the ServiceCenter application window titled "ServiceCenter - [Search Data Map Records]". The window has a menu bar with "File", "Edit", "View", "Format", "Options", "List Options", "Window", and "Help". Below the menu bar is a toolbar with icons for Back, Add, and Search. The main area is titled "Data Map" and contains the following fields:

- Desc: A text input field with a red border.
- Filename: A text input field.
- Field Name: A text input field.
- Array?: A checkbox.
- Arrayed Structure?: A checkbox.
- AS Field No.: A text input field.
- Map Query: A text input field.
- Query File: A text input field.
- Override Qry: A text input field.
- Map Cond: A text input field.
- Add to Map List: A text input field.

Below the Map Query field, there are three lines of italicized text:

- (Use \$umapdata to reference the Field Name contents for Types 0 and 1)*
- (Use "fieldname in \$umapfile0" for Type 0)*
- (Use "fieldname in \$umapfile" for Type 2)*

At the bottom of the window, the status bar shows "Ready", "Response 0.111 draw 0.40", "insert", and "datamap.g(db.search) [US]".

Figure 3-4: Data Map record

Field Descriptions

Field	Description
Desc	Brief text description of the Data Map record.
Filename	Name of a valid ServiceCenter file. For a Type 0 record, it defines the name of the file from which records are unloaded. For a Type 1 record, it defines the name of the file that contains input fields which point to other secondary records. For a Type 2 record, it defines the name of the file to execute the Unload Query against. This is a required field.

Field	Description
Type	Controls what type of processing occurs during the execution of the Data Map build routine. The Data Map Format Control record automatically builds this field. You should not modify the contents of this field or update the Format Control calculations statements that determine the value for this field. If you do, the results of the Data Map utility are unpredictable.
Field Name	Name of a field in the Filename dbdict. For a Type 0 record, it is the input field that contains the data that uniquely defines records in the Filename dbdict. For Type 1 and 2 records, it is the name of the field in the record that contains data that defines a record in a secondary file.
Array?	Specifies if the field defined in Field Name is an array. It is only used for Type 1 maps. The default is false. If it is set to true, Data Map processes each element from the array against the secondary file.
Arrayed Structure?	<p>Specifies if the field defined in Field Name is an arrayed structure. It is only used for Type 1 maps. The default is false. If it is true, you must indicate in the AS Field No which field number within the indicated arrayed structure contains the data that uniquely defines a record in a secondary file. If this field is set to true, Data Map processes each indicated field within each arrayed structure element against the secondary file.</p> <p>A good example of when you would set the Arrayed Structure flag to true is for the <i>format</i> file <i>field</i> arrayed structure. The <i>field</i> arrayed structure contains all of the input and label field definitions for a given format. By mapping field number 10 (the dbdict name for this field is <i>format</i>) of the <i>field</i> arrayed structure, Data Map will build a Map List entry for the subformats of a given format.</p>
AS Field No	Defines the field number within the arrayed structure that contains the data that uniquely defines a record in a secondary file. This field is required if the Arrayed Structure? field evaluates to true
Query File	<p>Name of a valid ServiceCenter File. (Not used for Type 0 records.)</p> <ul style="list-style-type: none"> ■ For Type 1 records, defines the file related to the input field defined in the Field Name. There must be a Type 0 record for the Query File of each Type 1 record. ■ For Type 2 records, defines the file that the Unload Query selects records from.

Field	Description
Map Query	<ul style="list-style-type: none"> ■ For <i>Type 0</i> records, defines the query that unloads a unique record from a file. If a record is found by executing this query, the Filename and Unload Query are added to the Map List. ■ For <i>Type 2</i> records, defines the query that extracts records from the Filename file. <p>You can refer to certain variables in this processing statement when building the unload query. They are:</p> <ul style="list-style-type: none"> ■ \$mapdata—this variable contains the data from the FieldName field of a primary record. ■ \$mapfile—this is a file variable of the current primary record. ■ \$map.modtime—this is the <i>modtime()</i> of the record currently contained in the <i>\$mapfile</i> record. ■ \$last.mod.date—either the current date/time or the data/time passed to the Data Map utility via the Last Mode Date parameter. ■ \$acntr—refers to the element currently being processed in the array or array of structures. This is available only for those Data Maps which have the Array? or Arrayed Structure? flags set to true.
Override Qry	<p>Defines the query that extracts records from a secondary file. This query overrides the default built in the <i>Type 0</i> definition Map Query input field.</p> <p>Under most circumstances, the relationship between a primary file and a secondary file is based on the contents of one input field in the primary and one input field in the secondary file. For instance, the relationship between the format file (the primary) and the formatctrl file (the secondary) is based on the contents of the name input field in the format file and the name input field in the formatctrl file. This allows you to minimize the number of times that you define the map query for a file which results in easier record maintenance.</p> <p>However, there are circumstances where the relationship between a primary file and a secondary file is based on many fields in the primary and many fields in the secondary and possibly in data that is not contained in the data record. In order to define this type of relationship, you must define the query (i.e., establish the relationship) in this field. This query is processed against the secondary file. If any records are found, the query is then added to the Map List. Refer to the Map Query field definition for more information on how the standard query is built.</p>

Field	Description
Map Cond	<p>Boolean field that allows you to control when to process a <i>Data Map</i>. This field works in conjunction with the Add to Map List condition. The rules are summarized in the Add to Map List field definition.</p> <hr/> <p>Warning: If this condition is <i>false</i> on the first record processed during a database unload or purge/archive, then none of the records in the list are processed.</p> <hr/>
Add to Map List	<p>Boolean field that allows you to control when to add an entry to the Map List. The field works in conjunction with the Map Cond field. This field controls the processing of <i>Type 0</i> records only.</p>

Data Map Types

You can define three different types of records in the `datamap` file.

- Type 0
- Type 1
- Type 2

Each type has different definition requirements. The *Data Map* Format Control automatically determines the type of a record based on the following rules.

Type 0

A *Type 0* record is the primary definition for a given file. If only the **Filename**, **Field Name** and **Map Query** fields are defined, the record is a *Type 0*. There is only one *Type 0* record for each file. This is the only type of map that actually adds an entry to the Map List. You should define the query in such a way that when it is executed against the indicated file, the mapping routine will retrieve certain specific records.

For example, to map a form from the format file, you must define the map with the following values:

Field	Input Value
Filename	format
Field Name	name This is the input field in the format file that uniquely identifies each record.
Map Query	"name=\\"+str(\$umapdata)+"\""

Type 1

A *Type 1* record is a secondary definition for a given file. If only the **Filename**, **Field Name** and **Query File** fields are defined the record is a *Type 1*. There can be multiple *Type 1* definitions for each file. They define which field in a given file contains information that points to another record in a different file. For example, in order to unload the Format Control associated with a format, you would define the map as:

Field	Input Value
Filename	format
Field Name	name
Query File	formatctrl

Note: There are circumstances where the relationship between two records is based on more than one field in the primary and secondary files. Refer to the definition for the **Override Query** field for more information on how Data Map accommodates this situation.

You can interpret the meaning of this *Type 1* definition as the **name** input field in the *format* file defines which record to map from the *formatctrl* file. This record points to the *Type 0* record for the *formatctrl* file.

To complete this map you would also need a *Type 0* entry for the *formatctrl* file.

Type2

A *Type 2* record is a *secondary definition* for a given file. If the **Filename**, **Field Name**, **Query File** and **Map Query** fields are defined the record is a *Type 2*. There can be multiple *Type 2* records for each file. A *Type 2* record defines which records to extract from a given file and then which field within those records contain information which points to another record in a different file.

An example situation for a *Type 2* record is to automatically map the formats defined on the RIO panels of a given RAD routine. To do this define the map as:

Field	Input Value
Filename	application
Field Name	name
Query File	format
Map Query	"format=\"rio\" and application=\""+str(application in \$umapfile)+"\""

You can interpret the meaning of this definition as: Execute the given Map Query (i.e., select all RIO panels for a particular application) against the **application** file. The **name** input field in each record found by executing this query defines the record to unload from the **format** file.

To complete this Data Map you would also need a *Type 0* record for the format file.

Rules Summarizing How Mapping Conditions Affect Data Map

The rules summarizing how the **Map Cond** and **Add to Map List** condition affect Data Map are:

For Type 0 Records

If the **Map Cond** and the **Add to Map List** conditions are *false*, a Map List entry is not created and the Type 1 and 2 records associated with that file are not processed.

If the **Map Cond** is *false* and the **Add to Map List** is *true*, a Map List entry is created for the file; however, the Type 1 or 2 records associated with that file are not processed.

If the **Map Cond** and **Add to Map List** conditions are *true*, a Map List entry is create and the Type 1 and 2 records associated with the Type 0 file are processed.

For Type 1 and 2 Records

If the **Map Cond** is *false*, the build process is interrupted and processing proceeds to the next Type 1 or 2 record for that file.

If the **Map Cond** is *true* the build process proceeds as described earlier.

Invoking the Data Map Utility from RAD

You can invoke the Data Map utility with a standard RAD call. The `data.map` application is automatically invoked by the standard database unload process and the standard purge/archive process.

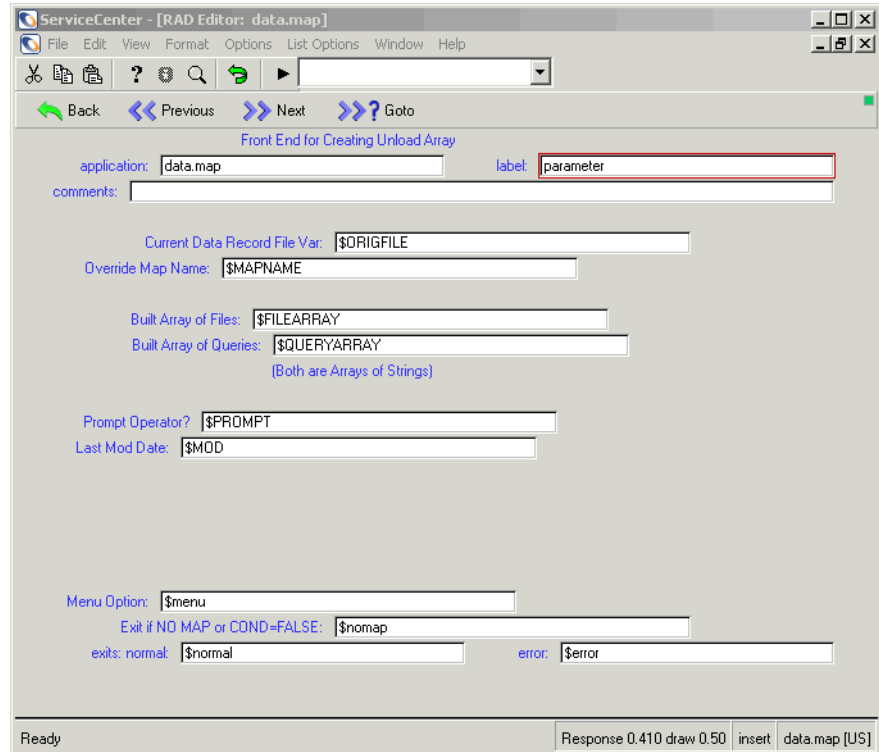


Figure 3-5: Parameter panel from the data.map application

Field	Description
Current Data Record File Var	Pass the current high level or original primary record file variable. This is the original primary file that has related secondary records.
Override Map Name	Default processing is to find the Data Map definition that has the filename() of the file variable passed to the previous parameter. You can override this by passing a character string to the Override Map Name parameter.
Built Array of Files	Array of files built by the Data Map utility. You should pass an empty array of characters to this parameter. There is a one to one relationship between the element in this array and the elements in the Built Array of Queries array.
Built Array of Queries	Array of queries built by the Data Map utility. You should pass an empty array of characters to this parameter. There is a one to one relationship between the element in this array and the elements in the Built Array of Files array.
Prompt Operator	Optional boolean field that controls when the operator is prompted with the Data Map Script and with the results of the Data Map build process. The default is false.
Last Mod Date	Optional date/time field that is copied to the global variable \$last.mod.date when the Data Map utility begins building the Map List. You can reference this variable in the Map Query field and control which data records are placed into the Map List by comparing this variable to the \$umap.modtime variable.

In this example, we will show the definitions required to unload all records associated with a standard ServiceCenter report (*cm3.change.committee*). This includes records from the following files:

report	format
formatctrl	enclapplication
application	code

To access the *cm3.change.committee* report:

- 1 In the System Administrator's home menu, select Toolkit > Report Writer. The Report Writer dialog box is displayed.
- 2 Enter *cm3.change.committee* in the Report field.
- 3 Click Search or press Enter.

The *cm3.change.committee* report definition is displayed in read-only mode.

The screenshot shows the 'ServiceCenter - [Report Writer]' application window. The main area is titled 'Report Maintenance' and is divided into several sections:

- Basic:**
 - Name:
 - Title: (dropdown menu)
 - Application: (dropdown menu)
 - Initial Form:
 - Primary File:
 - Query Form:
 - Author: (09/25/91 14:34:47)
 - Editor: (01/21/92 15:31:13)
 - Save Report
 - Summary
 - Detail
 - Auto
 - Manual
- Formats:**
 - Header: (edit icon)
 - Subheader: (edit icon)
 - Detail: (edit icon)
 - Print Master Header: (dropdown)
 - Print Subheader: (dropdown)
 - Subheader Page Break:
- Select:**
 - Primary Query:
 - Stacked Queries: (dropdown)
 - Sort Sequence: (dropdown)
 - (dropdown)
 - Lines Per Page:
 - Width (80/132):
- Subtotals:**

Format Name	Field/Variable Names to Total	Calculation	Variable Names on Format
<input type="text"/>	<input type="text"/>		
<input type="text"/>	<input type="text"/>		
<input type="text"/>	<input type="text"/>		

The status bar at the bottom shows 'Ready', 'Response 0.170 draw 0.100 insert report.g [US]'.

Figure 3-6: Change Management report

This is an automatic report, with Format Control records defined for both the *subhdr* and *detail* forms. There is no RAD associated with this report, and there are no *Totals* forms.

report File

- The Type 0 record is required for the report file to map the actual report definition itself. Any other Data Map definitions that have a Query File of *report* will use this record.

Note: The query in the Map Query field retrieves a unique record from that file.

Data Map

Desc: map the report definition itself

Filename: report Type: 0

Field Name: report.name

Array? f

Arrayed Structure? f AS Field No:

Map Query: "report.name='\''+str(\$umapdata)+'\''"

(Use \$umapdata to reference the Field Name contents for Types 0 and 1)
(Use "fieldname in \$umapfile0" for Type 0)
(Use "fieldname in \$umapfile" for Type 2)

Query File:

Override Qty:

Map Cond: true

Add to Map List: true

Ready Response 0.70 draw 0.60 insert datamap.g(db.search) [US]

Figure 3-7: Type 0 record for the report file—report definition

- This is one of the Type 1 records for the report file that causes the forms of the report to be unloaded. There are two other formats definitions needed for the report file. The input field names are *format.subfmt* and *format.detail*.

Note: The value in the Query File field is *format*. This will cause the Data Map utility to use the *format* Type 0 record that was defined earlier.

The screenshot shows a window titled "ServiceCenter - [Search Data Map Records]" with a menu bar (File, Edit, View, Format, Options, List Options, Window, Help) and a toolbar with icons for Back, Add, and Search. The main area is titled "Data Map" and contains the following fields:

- Desc: map the header format of a report
- Filename: report
- Type: 1
- Field Name: format.header
- Array?: f
- Arrayed Structure?: f AS Field No: []
- Map Query: []
- Query File: format (highlighted with a red box)
- Override Qty: []
- Map Cond: true
- Add to Map List: true

Below the Map Query field, there are three instructional lines:

- (Use \$umapdata to reference the Field Name contents for Types 0 and 1)*
- (Use "fieldname in \$umapfile0" for Type 0)*
- (Use "fieldname in \$umapfile" for Type 2)*

The status bar at the bottom shows "Ready", "Response 0.70 draw 0.60 insert", and "datamap.g(db.search) [US]".

Figure 3-8: Type 1 record for the report file—forms mapping

- This is another Type 1 record for the report file. In this case, note that we are defining an arrayed structure. This definition will map all of the Subtotal forms defined to the report. These forms are defined in AS Field No 2 (*format.subtotals*) of the *report.totals* arrayed structure. The value in the **Query File** field points to the standard Data Map for the format file.

Data Map

Desc: map the subtotals formats for a report

Filename: report Type: 1

Field Name: report.totals

Array? f

Arrayed Structure? t AS Field No: 2

Map Query:

(Use \$umapdata to reference the Field Name contents for Types 0 and 1)
(Use "fieldname in \$umapfile0" for Type 0)
(Use "fieldname in \$umapfile" for Type 2)

Query File: format

Override Qty:

Map Cond: true

Add to Map List: true

Ready Response 0.70 draw 0.60 insert datamap.g(db.search)[US]

Figure 3-9: Type 1 record for the report file—arrayed structure mapping

format File

- This is the Type 0 definition for the format file. Any other Data Map definitions that have a value of **report** in the **Query File** field will use this Data Map record.

Note: The query in the **Map Query** field retrieves a unique record from that file.

The screenshot shows a web-based configuration interface for a Data Map. The window title is "ServiceCenter - [Search Data Map Records]". The interface includes a menu bar (File, Edit, View, Format, Options, List Options, Window, Help) and a toolbar with icons for Back, Add, and Search. The main content area is titled "Data Map" and contains the following fields:

- Desc:** map the format itself
- Filename:** format
- Field Name:** name
- Array?:** f
- Arrayed Structure?:** f
- AS Field No.:** (empty)
- Map Query:** "name=\" + str(\$umapdata) + \"'
- Query File:** (empty)
- Override Qry:** (empty)
- Map Cond:** true
- Add to Map List:** true

Below the Map Query field, there are three lines of instructional text:

- (Use \$umapdata to reference the Field Name contents for Types 0 and 1)*
- (Use "fieldname in \$umapfile0" for Type 0)*
- (Use "fieldname in \$umapfile" for Type 2)*

The status bar at the bottom of the window displays: Ready, Response 0.70 draw 0.60 insert, datamap.g(db.search)[US]

Figure 3-10: Type 0 record for the format file

- This is a Type 1 record for the *format* file. This definition will map any Format Control records associated with the form.

ServiceCenter - [Search Data Map Records]

File Edit View Format Options List Options Window Help

Back Add Search

Data Map

Desc: map the formatctrl associated with a format

Filename: format Type: 1

Field Name: name

Array? f

Arrayed Structure? f AS Field No:

Map Query:

(Use \$umapdata to reference the Field Name contents for Types 0 and 1)
(Use "fieldname in \$umapfile0" for Type 0)
(Use "fieldname in \$umapfile" for Type 2)

Query File: formatctrl

Override Qry:

Map Cond: true

Add to Map List: true

Ready Response 0.70 draw 0.60 insert datamap.g(db.search) [US]

Figure 3-11: Type 1 record for the format file—Format Control mapping

- This is another Type 1 record for the *format* file. This definition will map any subformats associated with a format.

ServiceCenter - [Search Data Map Records]

File Edit View Format Options List Options Window Help

Back Add Search

Data Map

Desc: map all subformats defined in a given format record

Filename: format Type: 1

Field Name: field

Array? f

Arrayed Structure? t AS Field No: 10

Map Query:

(Use \$umapdata to reference the Field Name contents for Types 0 and 1)
(Use "fieldname in \$umapfile0" for Type 0)
(Use "fieldname in \$umapfile" for Type 2)

Query File: format

Override Qry:

Map Cond: true

Add to Map List: true

Ready Response 0.70 draw 0.60 insert datamap.g[db.search] [US]

Figure 3-12: Type 1 record for the format file—subformat mapping

formatctrl File

- This is the Type 0 record for the formatctrl file. Any other Data Map definitions that have a value of formatctrl in the Query File field will use this Data Map record.

Note: The query in the Map Query field retrieves a unique record from that file.

The screenshot shows a window titled "ServiceCenter - [Search Data Map Records]". The window has a menu bar (File, Edit, View, Format, Options, List Options, Window, Help) and a toolbar with icons for Back, Add, and Search. The main area is titled "Data Map" and contains the following fields:

- Desc: map the formatctrl record itself
- Filename: formatctrl
- Type: 0
- Field Name: name
- Array?: f
- Arrayed Structure?: f AS Field No: []
- Map Query: "name=\"+str(\$umapdata)+\""
- Query File: []
- Override Qry: []
- Map Cond: true
- Add to Map List: true

Below the Map Query field, there are three lines of italicized text:

- (Use \$umapdata to reference the Field Name contents for Types 0 and 1)*
- (Use "fieldname in \$umapfile0" for Type 0)*
- (Use "fieldname in \$umapfile" for Type 2)*

The status bar at the bottom shows "Ready", "Response 0.70 draw 0.60 insert", and "datamap.g(db.search)[US]".

Figure 3-13: Type 0 record for the formatctrl file

- This is the Type 1 record for the formatctrl file that maps any defined subroutines.

The screenshot shows a web-based interface for managing data maps. The window title is "ServiceCenter - [Search Data Map Records]". The main content area is titled "Data Map" and contains the following fields and values:

- Desc: map any subroutines
- Filename: formatctrl
- Field Name: subroutines
- Type: 1
- Array?: f
- Arrayed Structure?: t
- AS Field No.: 4
- Map Query: (empty)
- Query File: enclapplication
- Override Qry: (empty)
- Map Cond: false
- Add to Map List: true

Below the fields, there are three lines of instructional text:

- (Use \$umapdata to reference the Field Name contents for Types 0 and 1)*
- (Use "fieldname in \$umapfile0" for Type 0)*
- (Use "fieldname in \$umapfile" for Type 2)*

The status bar at the bottom of the window displays: "Ready", "Response 0.70 draw 0.60 insert", and "datamap.g[db.search] [US]".

Figure 3-14: Type 1 record for the formatctrl file—subroutine mapping

enclapplication File

- This is the Type 0 record for the enclapplication file. Any other Data Map definitions that have a value of **enclapplication** in the **Query File** field will use this Data Map record.

Note: The query in the **Map Query** field retrieves a unique record from that file.

The screenshot shows the 'Data Map' configuration window in ServiceCenter. The window title is 'ServiceCenter - [Search Data Map Records]'. The menu bar includes File, Edit, View, Format, Options, List Options, Window, and Help. The toolbar contains icons for Back, Add, and Search. The main area is titled 'Data Map' and contains the following fields:

- Desc:** map enclapplication record for an application
- Filename:** enclapplication
- Type:** 0
- Field Name:** application.name
- Array?** f
- Arrayed Structure?** f
- AS Field No.:** (empty)
- Map Query:** "application.name='\"+str(\$umapdata)+'\""
- (Use \$umapdata to reference the Field Name contents for Types 0 and 1)*
- (Use "fieldname in \$umapfile0" for Type 0)*
- (Use "fieldname in \$umapfile" for Type 2)*
- Query File:** (empty)
- Override Qry:** (empty)
- Map Cond:** true
- Add to Map List:** true

The status bar at the bottom shows 'Ready', 'Response 0.70 draw 0.60 insert', and 'datamap.g(db.search)[US]'.

Figure 3-15: Type 0 record for the enclapplication file

- This is a Type 1 record for the enclapplication file that maps the associated application records of a RAD routine.

The screenshot shows a window titled "ServiceCenter - [Search Data Map Records]" with a menu bar (File, Edit, View, Format, Options, List Options, Window, Help) and a toolbar (Back, Add, Search). The main area is titled "Data Map" and contains the following fields:

- Desc: map the application records associated with an enclapplic
- Filename: enclapplication
- Field Name: application.name
- Type: 1
- Array?: f
- Arrayed Structure?: f
- AS Field No.:
- Map Query:

Below the fields, there are three instructions:

- (Use \$mapdata to reference the Field Name contents for Types 0 and 1)*
- (Use "fieldname in \$mapfile0" for Type 0)*
- (Use "fieldname in \$mapfile" for Type 2)*

The "Query File" field contains the text "application" and is highlighted with a red border. Other fields include "Override Qry:", "Map Cond: true", and "Add to Map List: true".

At the bottom of the window, the status bar shows: Ready, Response 0.70 draw 0.60 insert, datamap.g[db.search] [US]

Figure 3-16: Type 1 record for the enclapplication file—application record mapping

- This is a Type 1 record for the enclapplication file that maps the associated *code* record of a RAD routine.

The screenshot shows the 'Data Map' window in ServiceCenter. The window title is 'ServiceCenter - [Search Data Map Records]'. The menu bar includes File, Edit, View, Format, Options, List Options, Window, and Help. The toolbar contains icons for Back, Add, and Search. The main area is titled 'Data Map' and contains the following fields:

- Desc: map code record associated with an enclapplication
- Filename: enclapplication
- Field Name: application.name
- Type: 1
- Array?: f
- Arrayed Structure?: f
- AS Field No.: [empty]
- Map Query: [empty]

Below the fields, there are three instructions:

- (Use \$umapdata to reference the Field Name contents for Types 0 and 1)*
- (Use "fieldname in \$umapfile0" for Type 0)*
- (Use "fieldname in \$umapfile" for Type 2)*

The 'Query File' field contains the value 'code' and is highlighted with a red box. Other fields include 'Override Qry:', 'Map Cond: true', and 'Add to Map List: true'. The status bar at the bottom shows 'Ready', 'Response 0.70 draw 0.60 insert', and 'datamap.g[db.search] [US]'.

Figure 3-17: Type 1 record for the enclapplication file—code record mapping

- This is a Type 1 record for the *enclapplication* file that maps the associated *format* record of a RAD routine.

The screenshot shows the ServiceCenter application window titled "[Search Data Map Records]". The main area is titled "Data Map" and contains the following fields and values:

- Desc: map the format associated with an enclapplication
- Filename: enclapplication
- Field Name: application.name
- Type: 1
- Array?: f
- Arrayed Structure?: f
- AS Field No.: (empty)
- Map Query: (empty)
- Query File: format
- Override Qry: (empty)
- Map Cond: true
- Add to Map List: true

Below the Map Query field, there are three instructional notes:

- (Use \$umapdata to reference the Field Name contents for Types 0 and 1)*
- (Use "fieldname in \$umapfile0" for Type 0)*
- (Use "fieldname in \$umapfile1" for Type 2)*

The status bar at the bottom shows "Ready", "Response 0.70 draw 0.60 insert", and "datamap.g[db.search] [US]".

Figure 3-18: Type 1 record for the enclapplication file—form at mapping

application File

- This is the Type 0 record for the *application* file. Any other Data Map definitions that have a value of application in the Query File field will use this Data Map record.

Note: The query in the Map Query field retrieves a unique record from that file.

Data Map

Desc:

Filename: Type:

Field Name:

Array?:

Arrayed Structure?: AS Field No.:

Map Query:

(Use \$umapdata to reference the Field Name contents for Types 0 and 1)
(Use "fieldname in \$umapfile0" for Type 0)
(Use "fieldname in \$umapfile" for Type 2)

Query File:

Override Qry.:

Map Cond.:

Add to Map List:

Ready Response 0.70 draw 0.60 insert datamap.g(db.search)[US]

Figure 3-19: Type 0 record for the application file

- This is a Type 2 record for the application file. It finds all rio panels of a given application and maps the format name defined in each panel.

Note: You could also define the appropriate Type 2 records to find all print panels for a given application and unload the forms that they reference.

ServiceCenter - [Search Data Map Records]

File Edit View Format Options List Options Window Help

Back Add Search

Data Map

Desc: map formats defined on application rio panels

Filename: application Type: 2

Field Name: name

Array? f

Arrayed Structure? f AS Field No:

Map Query: "format='rio\' and application=''+str(application in \$umapfile)+'"

(Use \$umapdata to reference the Field Name contents for Types 0 and 1)
(Use "fieldname in \$umapfile0" for Type 0)
(Use "fieldname in \$umapfile" for Type 2)

Query File: format

Override Qry:

Map Cond: true

Add to Map List: true

Ready Response 0.70 draw 0.60 insert datamap.g(db.search) [US]

Figure 3-20: Type 2 record for the application file

code File

- This is the Type 0 record for the *code* file. Any other Data Map definitions that have a Query File of *code* will use this Data Map record.

Note: The query in the Map Query field retrieves a unique record from that file.

ServiceCenter - [Search Data Map Records]

File Edit View Format Options List Options Window Help

Back Add Search

Data Map

Desc: map code record for an application

Filename: code Type: 0

Field Name: application

Array? f

Arrayed Structure? f AS Field No:

Map Query: "application=\"+str(\$umapdata)+\""

(Use \$umapdata to reference the Field Name contents for Types 0 and 1)
(Use "fieldname in \$umapfile0" for Type 0)
(Use "fieldname in \$umapfile" for Type 2)

Query File:

Override Qry:

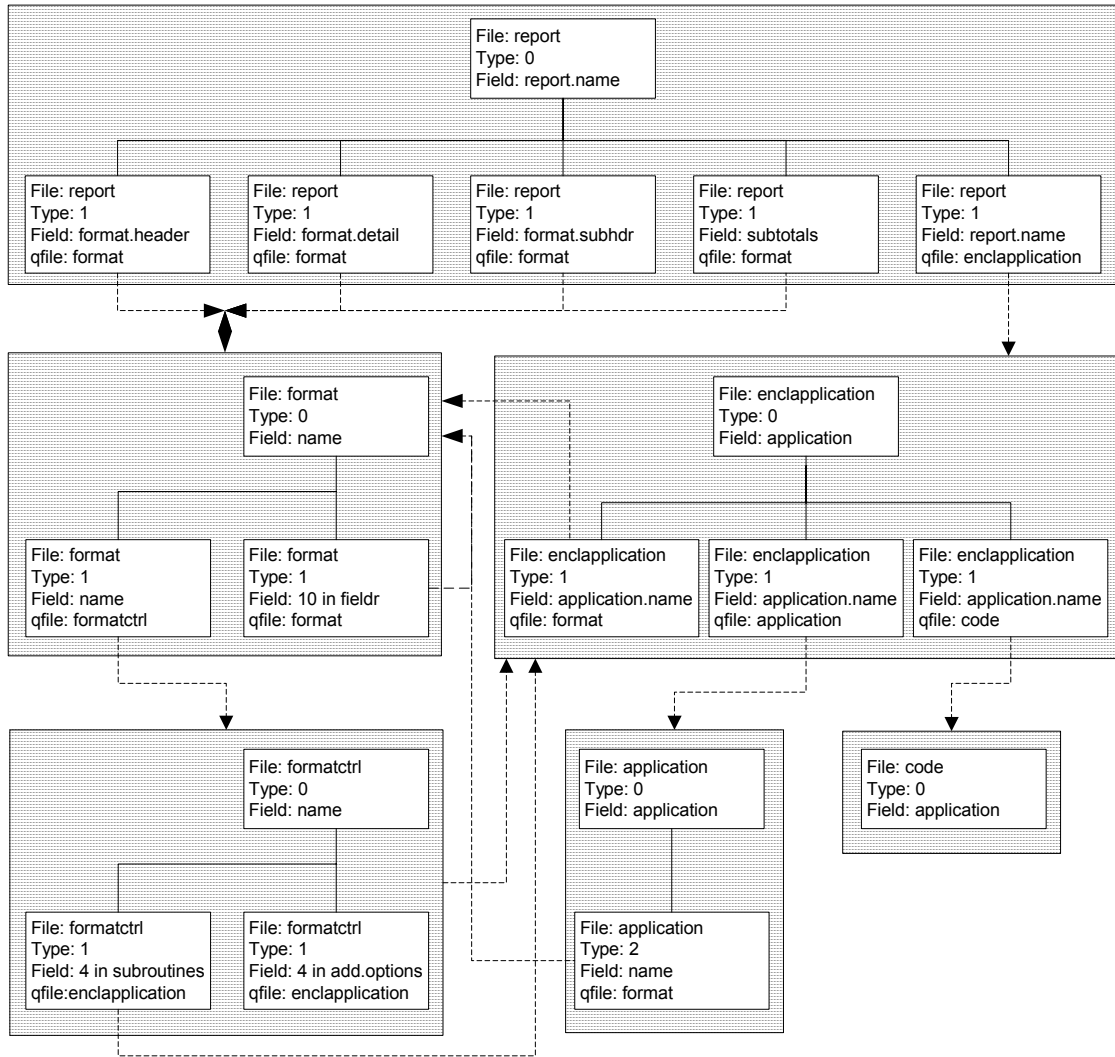
Map Cond: true

Add to Map List: true

Ready Response 0.70 draw 0.60 insert datamap.g(db.search) [US]

Figure 3-21: Type 0 record for the code file

Graphical View of Example Maps



This flowchart summarizes the example map illustrated in this section. The shaded areas represent the map definitions for a given file. The solid lines represent the relationship of the map definitions for a given file. The dashed lines represent the relationship between a Type 1 or Type 2 record to a Type 0 record. The arrows point to the shaded area itself, however, the map always starts with the Type 0 record within the shaded area.

Unloading with a Data Map

We will now unload all the records associated with the report *cm3.change.committee* as defined by the Data Map in the previous section.

Important: To unload these records, you must access the report file from the Database Manager and not from a command line.

To execute a standard database unload using a Data Map:

- 1 Select the Toolkit tab in the system administrator's home menu.
- 2 Click Database Manager.
The Database Manager dialog box is displayed.
- 3 Type =report in the Form field.
- 4 Press Enter.
A blank report definition form is displayed in read-only mode.
- 5 Click Search.
A QBE list of available reports is displayed.
- 6 Select *cm3.change.committee* from the list.

The report definition of *cm3.change.committee* is displayed.

The screenshot shows the 'Report Maintenance' window in the ServiceCenter - [Report Writer] application. The window is divided into several sections:

- Basic:**
 - Name: cm3.change.committee
 - Title: All Requests Planned to Start
 - Application: Within the Next 30 Days
 - Initial Form: (empty)
 - Primary File: cm3r
 - Query Form: (empty)
 - Author: alan (09/25/91 14:34:47)
 - Editor: falcon (04/22/03 12:49:44)
 - Save Report
 - Summary: (selected)
 - Detail: (selected)
 - Auto: (selected)
 - Manual:
- Formats:**
 - Header: cm3.change.committee.header (Print Master Header: Yes)
 - Subheader: cm3.change.committee.subhdr (Print Subheader: Yes)
 - Detail: cm3.change.committee.detail (Subheader Page Break: (empty))
- Select:**
 - Primary Query: true
 - Stacked Queries: header,last=true and header,open=true
 - Sort Sequence: header,planned.start; header,planned.end
 - Lines Per Page: 60
 - Width (80/132): 80
- Subtotals:**

Format Name	Field/Variable Names to Total	Calculation	Variable Names on Format
(empty)	(empty)	(empty)	(empty)
(empty)	(empty)	(empty)	(empty)
(empty)	(empty)	(empty)	(empty)

At the bottom of the window, a status bar indicates: Record updated in the report file. Response 0.140 draw 0.100 insert report.g [US]

Figure 3-22: Report definition for *cm3.change.committee*

7 Select Options > Export/Unload.

The Unload/Export Facility form is displayed.

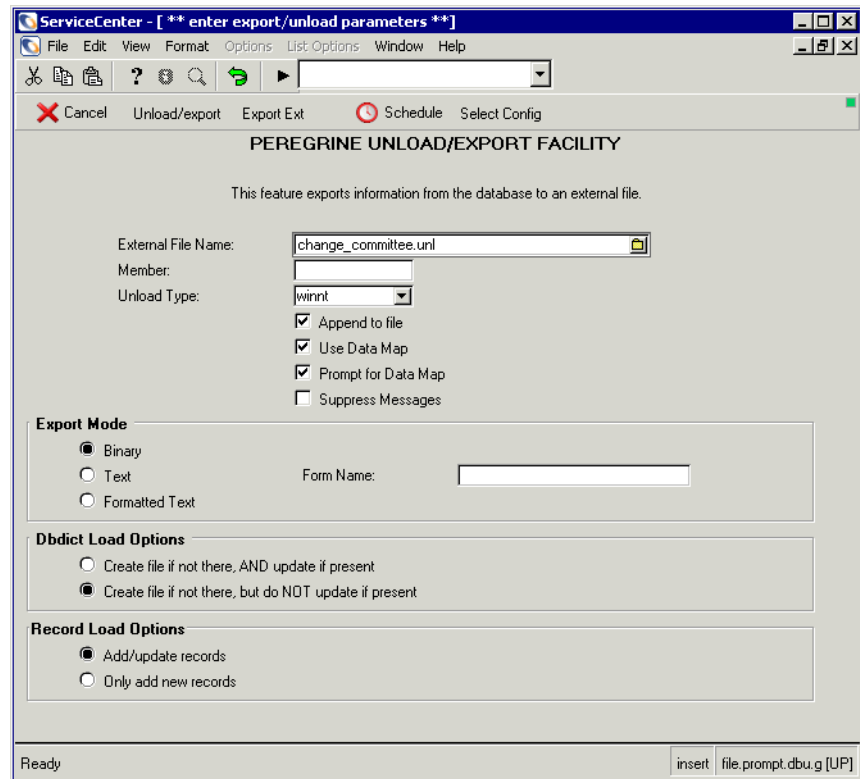


Figure 3-23: Unload/export utility form

8 Enter a name for the external file.

You may also include a fully qualified path in the **External File Name** field. If you want the unload file to be saved to the RUN directory of the client, select **File > Client side Load\Unload**. If this option is not enabled, the unload file for a client is saved in the RUN directory of the server.

9 Select the appropriate Data Map options.

- Click **Use Data Map** to unload all the records defined by the Data Map for the cm3.change.committee report. If this option is not selected, the system will unload the report only.
- Click **Prompt for Data Map** to display a confirmation list of records to unload. If this option is not selected, the system will perform the unload without asking for confirmation.

10 Click Unload/export.

The Data Map Input form is opened, showing a field called **Enter the Last Modify Date** in which the current date and time is displayed. This date is maintained in the variable *\$last.mod.date*. None of the map definitions we have defined in this example reference this field.

Note: If you are executing the unload in the foreground, the Data Map utility executes a script named *unload.map*. If the unload is performed in the background or if the **Prompt Data Map** condition is *false*, the following prompt screen is not displayed.

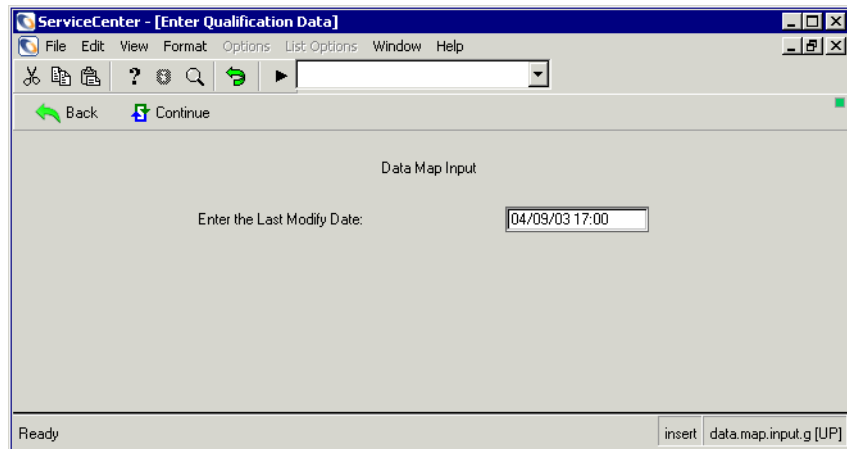


Figure 3-24: Last modification date

11 Click Continue.

A list of records defined by the Data Map is displayed. You can delete records from this list that you do not want to unload at this time by placing the cursor on a line and clicking **Delete**.

Note: If you are executing the unload in the foreground, the Data Map utility executes a script named *unload.map*. If the unload is performed in the background or if the **Prompt Data Map** condition is *false*, the following prompt screen is not displayed.

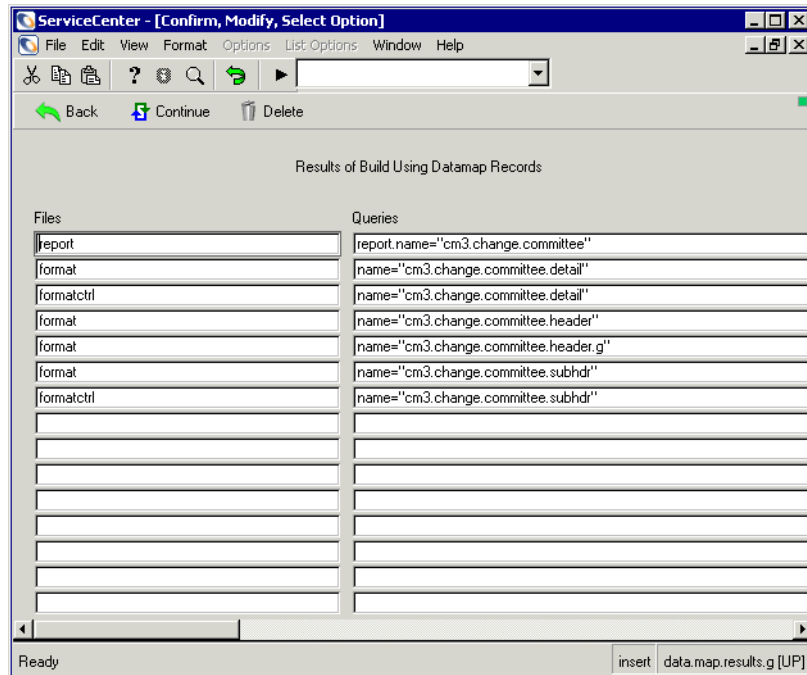


Figure 3-25: Data Map list of records to unload with this report

12 Click **Continue** again.

You are returned to the report definition record (Figure 3-22 on page 87). The following message is displayed in the status bar: *<n> records from report unloaded to: <external file name>, 00:00:00 elapsed.*



- 13 Click the message icon to display all the messages associated with this unload (blue indicates a call to action, black indicates informational only, and red indicates that it's an error message).

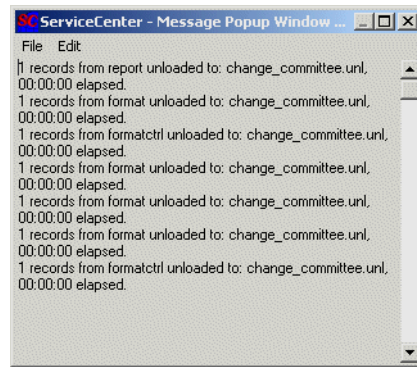


Figure 3-26: Unload messages

Special Considerations

The Data Map definitions described up to this point show how to map *all* records associated with some primary records. However, in some instances, you may want to unload only those records that have been modified since a particular date. For example, a report is in production status, and needs some enhancements. From your production system, you would probably unload all records related to the primary record definition (i.e., the report) regardless of when they were modified. This unload file would then be loaded into your test system. After making the necessary changes in your test system you could either unload all related records back to production, or unload only those records that were modified after being loaded to the test system.

The Data Map utility allows you to control which records to map with the **Map Cond** field. This is a boolean field that you can set to *true* or *false* or to a condition that evaluates to *true* or *false*. Data Map executes a Script that prompts you to enter a *last modified date* (Figure 3-24 on page 89). It defaults this date to the current date/time and stores it in the global variable *\$last.mod.date*.

You can use the *\$last.mod.date* variable in the **Map Cond** field for Type 0 records. For instance, if you only wanted to map *formats* that have been modified after a certain date, then define the **Map Cond** as:

```
$umap.modtime>$last.mod.date
```

The variable `$umap.modtime` is a global variable that contains the `modtime()` of the file variable `$umapfile`. The `modtime()` function is a standard ServiceCenter function that returns the date/time of the record contained in the indicated file variable that was last modified. Refer to *System Language* in *System Tailoring* for more information.

You can also define the **Map Cond** to examine other data fields in the record. For instance, you could define the Type 0 record for the `enclapplication` file (this is the file that contains the application encyclopedia record for a RAD routine) to check the compile date. This condition would be written as:

```
last.compile.date in $umapfile>$last.mod.date
```

The **Map Cond** is also evaluated for all Type 1 and Type 2 records before they are executed. Therefore, if you did not want to map the forms associated with a report, but did want to map the forms associated with a `cm3rcatphase`, you could set the **Map Cond** for the Type 1 format records associated with the report file to `false`.

4 Unload/Load

CHAPTER

Overview

Load/Unload was designed to aid ServiceCenter System and Database administrators in performing load and unload functions on multiple records within a database.

This chapter is divided into the following sections:

- *Unloading ServiceCenter Data* on page 93.
- *Loading ServiceCenter Databases* on page 127.

Unloading ServiceCenter Data

The Database Manager *unload* and *load* functions allow the user to write records to and read records from an external OS/390, Unix or Windows NT/2000 file. The contents of the external dataset or file are generally indecipherable to any software other than ServiceCenter, therefore the user needs to use the import/export functions to transfer data to or from a source other than ServiceCenter. Refer to *Export Utility* on page 137 and *Import Utility* on page 171 for more information.

The unload function writes data from the current ServiceCenter system to an external dataset or file (as specified by the user) for future loading into the same or another ServiceCenter system via the load function. Load options governing how ServiceCenter later loads the data are also included in the *unload* file.

Note: The unload file is saved to the server's Run directory, unless otherwise specified. If you cannot locate the exported file on the client machine, refer to the server Run directory.

Note: To control the destination of unload files, enable/disable the **Client Side Load/Unload** option in the File menu and set the server control parameter in the sc.ini file of the server, as outlined in the following steps.

To turn on the Client Side Load/Unload option:

- 1 From the **File** menu, select **Client side Load/Unload**.
- 2 Make sure the option is checked.

If you are already in Database Manager, exit to the menu, then start Database Manager again. Export file names are related to your client machine, not the server. For example, C:\export.txt will write to the C: drive of your local machine, not to the C: drive of the ServiceCenter server.

- 3 Add the following server control parameter to the sc.ini file of the server: `clientsideunload:n` where *n* indicates whether or not **File > Client side load/unload** is enabled.
 - a 1—**File > Client side Load/Unload** is *enabled*. Client load and unload files are located/sent to the **client RUN** folder.
 - b 0—**File > Client side Load/Unload** is *disabled*. Client load and unload files are located/sent to the **server RUN** folder.

The unload function always writes out the database dictionary of the record(s) being unloaded. When the file is loaded at a later date, records are directed automatically to the database specified in the unloaded database dictionary, i.e., records unloaded from a database are loaded into a database of identical name.

This section discusses the various options available within the Database Manager for unloading records from one ServiceCenter file at a time. In addition, the ServiceCenter Unload Script Utility lets you unload unrelated or related records from many ServiceCenter files at once. The unloaded data formats are identical, whether you use Database Manager Export/Unload or the Unload Script Utility, and the same Database Manager Load process is used to load data unloaded via either method. See *Creating an Unload Script* on page 14 for more details.

Unloading a Single Record from a Database

You may want to copy one particular *device* record from your production ServiceCenter environment into your development system. Accomplishing this involves an unload of the desired record from the production system, followed by a load from within the development system.

To unload a single record:

- 1 Access the device record you want to unload.

Note: The **Export/Unload** option only performs on a single record when selected from the **Options** menu.

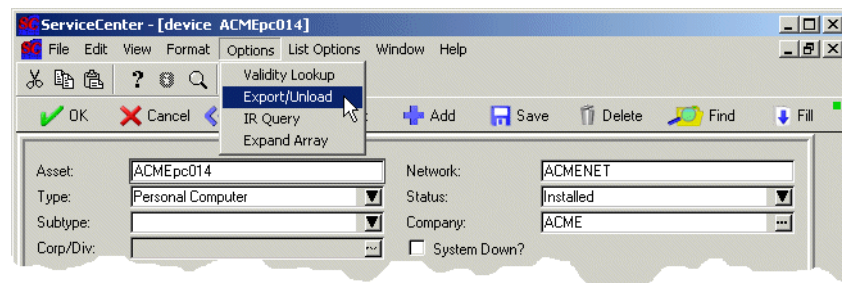


Figure 4-1: Preparing to unload a record from the Database Manager

- 2 Select **Options > Export/Unload**.

The Unload/Export Facility form is displayed.

Unload parameters fall into two categories:

- Those which control how data is written to an external file (unload options).
- Those which control how the same data is to be read back into ServiceCenter via the Load utility (load options).

Both categories of options are set prior to the actual unload.

Unload options control the immediate allocation of the external dataset and the format of the data written to the external dataset. The load options are written to the external dataset along with the unloaded record(s). These load options remain dormant until the external dataset is accessed by the Load utility, at which time they control how the unloaded data is read back into ServiceCenter. See *Unload Messages* on page 98.

3 To continue with this example, using the default unload parameters, enter the fully-qualified dataset name to which ServiceCenter will write this single device record.

- Enter <qualifier>.example1, where <qualifier> conforms to your standard naming conventions for OS/390 datasets.

-or-

- Enter <path>.example1, where <path> conforms to your standard naming conventions for Unix or Windows NT/2000 files.

Note: Select File > Client side Load/Unload to have the unload file saved in the client RUN directory. If you have selected this option, it is not necessary to define a qualified path. If you do not select the client-side unload option from a client and do not define a path for the unload file, the system will save the file in the server RUN directory.

4 Click Unload/export to begin the unload.

ServiceCenter - [** enter export/unload parameters **]

File Edit View Format Options List Options Window Help

F3 - Cancel F1 - unload/export F2 - Export Ext F7 - Schedule F8 - Select Config

PEREGRINE UNLOAD/EXPORT FACILITY

This feature exports information from the database to an external file.

External File Name:

Member:

Unload Type:

Append to file
 Use Data Map
 Prompt for Data Map
 Suppress Messages

Export Mode

Binary
 Text
 Formatted Text

Form Name:

Dbdict Load Options

Create file if not there, AND update if present
 Create file if not there, but do NOT update if present

Record Load Options

Add/update records
 Only add new records

Ready Response 0.160 draw 0.90 insert file.prompt.dbu.g [S]

Figure 4-2: Configuring unload settings

Unload Complication Messages

- A successful unload of records generates the following messages when unloading to an external dataset that does not already exist:
 - * *unable to allocate*
 - * *device file unloaded to external file: <system.filename>*
 - <nn> records from device unloaded to: <system.filename> HH:MM:SS elapsed
- A successful unload generates the following messages when unloading to preallocated external dataset:
 - **device file unloaded to external file: <system.filename>*
 - <nn> records from device unloaded to: <system.filename> HH:MM:SS elapsed
- The following messages are generated when ServiceCenter cannot connect to the named external file:
 - **device file cannot be unloaded to external file: <system.filename>*
 - **unable to allocate unable to allocate*

Often, a dataset that cannot be allocated lacks a certain high level qualifier required by a security facility such as Top Secret, RACF, or ACF2. Check with your internal Systems Programmer or Security Officer for the required qualifier, and try the unload again.

The unload is executed as a foreground process, meaning the ServiceCenter terminal session is devoted to executing the unload, and terminal control is returned after the unload is completed. The Database Manager unload form remains on your screen while the unload is processed.

Upon completion, the *format* record list is re-displayed along with several possible messages displayed at the bottom of your screen (unload messages are also written to the msglog file).

The three components of the report (report, form, and Format Control records) now reside in the external dataset and can be loaded into your development ServiceCenter system with the **Load** function. Refer to [Loading ServiceCenter Databases](#) on page 127.

Unload Messages

Message	Description
Invalid unload append/create parameter.	Parameter must be <i>append</i> or <i>create</i> 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 an export descriptor record is specified, the unload type must be <i>export</i> 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 file is available to load.	Occurs when the unload function is accessed without the proper parameters (i.e., from the call panel). Access the unload function from Database Manager instead.
No unload file name.	Unload requires an external file name.
The unload format must be binary for Purge/Archive.	When unloads are accessed via Purge/Archive, the unload format must be binary. Refer to <i>Purge/Archive</i> on page 35.
<database name> file unloaded to external file: <OS/390 file or Unix filename>.	Unload processing the named database to the named OS/390 dataset or Unix file complete.
config record: <config name> is not set up to be used for database unloads.	The true/false field Used for database loads/unloads: on the configuration record database load/unload.mvs is set to <i>false</i> . It must be set to <i>true</i> . Refer to the <i>Benchmark Utility</i> on page 337 for information on configuration records.

Message	Description
External file: <MVS.FILENAME> or <UNIX.filename> is locked; try again later.	Another task has exclusive control over the named external dataset.
Invalid number of records found in config file.	The unload utility found two or more hardware configuration records named <i>database load/unload.mvs</i> . There should only be one such record. Refer to <i>Benchmark Utility</i> on page 337 for information on configuration records.
The config record (database load/unload) is missing.	The hardware configuration record <i>database load/unload.mvs</i> , <i>database load/unload.unix</i> , or <i>database load/unload.winnt</i> required for database loads and unloads has been deleted or renamed. It must be replaced.
The config record (database create) is missing.	The hardware configuration record <i>database create.mvs</i> or <i>database create.unix</i> required for the dynamic allocation of an external OS/390 dataset or Unix/NT filename has been deleted or renamed. It must be replaced.
<nn> records unloaded so far from <database name> file.	The unload utility generates this message every hundredth record it attempts to unload.
<nn> records from <database name> unloaded into external file: <MVS.FILENAME> or <UNIX/NT.FILENAME>.	Unload processing nn records from the named ServiceCenter database to the named OS/390 dataset or Unix/NT file completed.
<database name> file CANNOT be unloaded to external file: <MVS.FILENAME> or <UNIX/NT.FILENAME>.	ServiceCenter could not connect to the named external file. Often, a dataset that cannot be allocated lacks a certain high-level qualifier required by a security facility such as Top Secret, RACF, or ACF2.

Message	Description
Unable to allocate.	<p>This message occurs in two distinct situations:</p> <p>(1) The unload utility attempted to connect to a non-existent external dataset causing the message to be generated. However, the unload utility will attempt to dynamically allocate the external OS/390 dataset or Unix/NT filename. If the dynamic allocation is successful, confirmation messages will follow. In this situation, it does not indicate an error condition.</p> <p>(2) The unload utility attempted to connect to a non-existent external dataset causing the message to be generated. The subsequent attempt to dynamically allocate the external OS/390 dataset or Unix/NT filename failed causing the message to be generated a second time. Other error messages will follow indicating the error condition.</p>
Application:file.unload has completed with errors; check other messages.	An error exit was taken; 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 Form Fields

Field	Description
External File Name	Fully qualified name of the external file to which you want the information written. In the OS/390 environment, you must enter an existing high level qualifier, or the process will wait on execution for action at the system console. In the Unix and Windows NT/2000 environment, the full path name should be entered if the file is being written to a directory other than the one in which the ServiceCenter files are located.
Member	Name of the member if the external file is an OS/390 partitioned dataset. Otherwise, leave the field blank.
Unload Type	Determines the output format of the data, based on the system to which the data will be written. Options include mswin, winnt, unix, OS/390, sun and other common platforms. In the current release, the system takes care of this automatically, depending on the platform where the unload file resides.
Append to Existing File	<p>Default is yes, which indicates that the exported information will be appended to any existing file of the same name, indicated in the External File Name field. If a file of the same name does not exist, one will be created. If the field is set to no, any existing file of the same name will be replaced with the exported information or, if none exists, a file of that name will be created.</p> <p>Note: In an OS/390 environment, remember that you cannot append to a partitioned dataset (PDS) member. When a member is defined, the application ALWAYS creates a new dataset without respect to the parameter entered.</p>
Use Data Map	Default is <i>false</i> . If this box is checked (<i>true</i>), Datamap executes for every record that is unloaded. Refer to tData Mapping on page 55 for information on data mapping.
Prompt Data Map	Default is <i>false</i> . If this box is checked (<i>true</i>), and the Use Data Map field is <i>true</i> , the Datamap application displays the standard Datamap confirmation screen for every record that is unloaded.
Suppress Messages	Allows you to ignore all messages produced during the unload process, including the final confirmation message.

Field	Description
Binary	Writes the file in ServiceCenter internal binary formatted mode, which can only be read by the Database Manager's Load facility.
Text	Writes the file in ServiceCenter internal text mode. Exported text can be read by humans, unlike Binary mode. However, it is not formatted for data transfer to external applications. It is only readable by another ServiceCenter system, using the Database Manager's Load facility.
Formatted Text	Exports information in a format that can be read by external programs, and structured to match an existing ServiceCenter text (not Windows) format.
Form Name	Name of the format that describes the information to be exported via the Formatted Text Unload/export option. If the format does not exist, you will be prompted to create it.
Use Current File Description	Used when loading ServiceCenter internal binary or text formatted information. In later releases it is also used in the Export and Import (load) utilities. Warning: When loading a file exported with the above check box activated, the file is loaded using the database dictionary file name of the new database dictionary into which the records will be loaded (imported), rather than the file name referenced at the time the file was created.
Update Duplicate Records	Used when loading ServiceCenter internal binary or text formatted information. It is not used in the Export and Import (load) Utilities. Activating this check box automatically updates any records in the current file with values from the loaded file when unique key values (e.g. logical.name, number, etc.) between the records are the same.

The unload is executed as a foreground process, meaning that the ServiceCenter terminal session will be devoted to executing the unload and terminal control is returned after the unload has completed. The Database Manager *unload* form remain on-screen while the unload is processed.

Upon completion, the device record is re-displayed along with several possible messages displayed at the bottom of the screen (unload messages are also written to the *msglog* file). In Unix and Windows NT/2000 the following message is displayed in the status bar: *<n> records from device unloaded to: <unload file name>, <HH:MM:SS> elapsed.*

After a successful unload, the file is available for loading it into another system. Refer to *Loading ServiceCenter Databases* on page 127.

Unloading Multiple Records from One Database

A user may want to copy several device database records from your *production* ServiceCenter environment into *development*. Accomplishing this via the QBE method involves using a QBE format to select the desired records and unloading them from production.

To unload multiple records:

- 1 Access the Database Manager.
- 2 Click **Search** to perform a *true* query and retrieve a list of all database formats and files.
- 3 Select **AlertDef** format from the QBE list of records.
- 4 Click **Search** to bring up a list of Alert file names.

The *AlertDef.qbe* form is displayed.

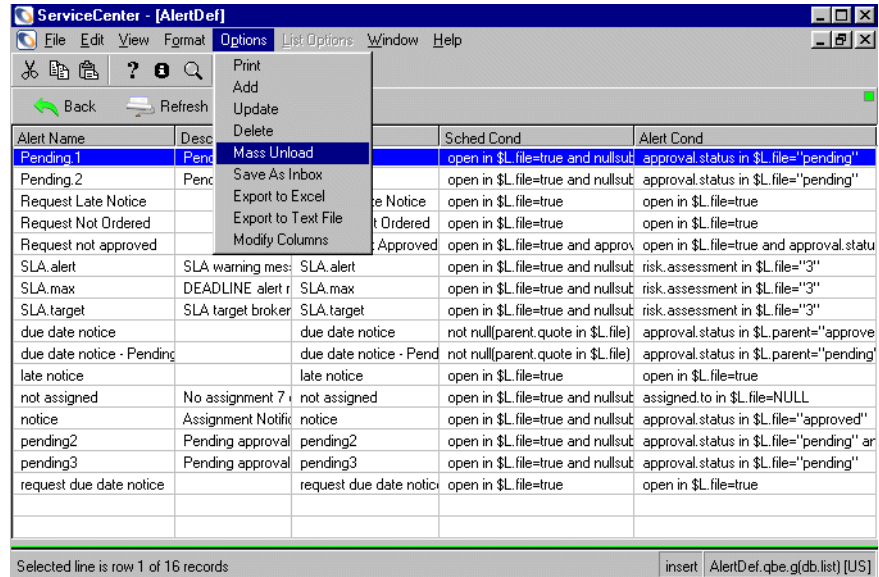


Figure 4-3: Unloading a list of records

- 5 Select **Options** > **Mass Unload** to begin the unload process for this list of records.

Note: If you are using a record list, select **List Options** > **Mass Unload**.

The default unload/export screen is displayed with a prompt requesting unload parameters (Figure 4-2 on page 96).

Unload parameters fall into two categories:

- Those which control how data is written to an external dataset (unload options).
- Those which control how the same data is to be read BACK into ServiceCenter via the load utility (load options).

Both categories of options are set prior to the actual unload.

Unload options control the immediate allocation of the external dataset and the format of the data written to the external dataset. The load options are written to the external dataset along with the unloaded record(s). These load options remain dormant until the external dataset is accessed by the Load Utility at which time they control how the unloaded data is read back into ServiceCenter.

- 6 Enter the fully-qualified dataset name to which ServiceCenter will write this group of device file records.
 - Enter <qualifier>.example2, where <qualifier> conforms to your standard naming conventions for OS/390 datasets.
 - or-
 - Enter <path>.example2, where <path> conforms to your standard naming conventions for Unix or Windows NT/2000 files.
- 7 Click **Unload/export** to begin the unload.

The unload is executed as a foreground process, meaning that the ServiceCenter terminal session will be devoted to executing the unload and terminal control is returned after the unload has completed. The Database Manager *unload* form remains on-screen while the unload is processed.

Upon completion, the device file list is re-displayed along with several possible messages displayed at the bottom of your screen (unload messages are also written to the *msglog* file).

Note: When scrolling unload messages from a list, enter a scroll setting of half or page, otherwise a scroll setting of recd causes the list itself to be scrolled, causing the unload messages to be lost.

After a successful unload, the file is available for loading it into another system. Refer to *Loading ServiceCenter Databases* on page 127.

Unloading Multiple Records from a Single File Using Value-list

A user may want to copy several device database records from a production ServiceCenter environment into a development environment.

Accomplishing this via the *Value-List* method involves entering the list of desired records, unloading them from production, then loading them from within the development system.

The *Value-List* method is superior to the *query* method in situations where a list of desired records cannot be generated practically via a query. For example, a user may want to unload device records with the following logical names:

ACME Phone 0001	ACME Phone 0002
ACME Phone 0003	ACME Phone 0004
ACME Phone 0005	ACME Phone 0006
ACMEpc012	ACMEpc013
ACMEpc014	ACMEpc015
ACMEpc016	ACMEprinter
ACMEserver002	

The only queries capable of producing such a list (exclusive of other logical.names) are complex and awkward, whereas the *Value-List* method allows the user to enter the field name and the list of values for that field name to specify the list of records to export or unload.

To unload multiple records from a single file using a value-list:

- 1 Access the device form.
- 2 Without querying for records select **Options > Export/Unload**.
An Export/Unload utility form is displayed (Figure 4-4 on page 107) in which you can specify a field name and an array of field values (value list).
- 3 Enter logical.name in the **Field Name** field.
- 4 Enter the desired logical.name values in the **Field Values** array.

For this example, enter the following asset names:

- ACME Phone 0001
- ACMEpc012
- ACMEprinter
- ADMEserver002

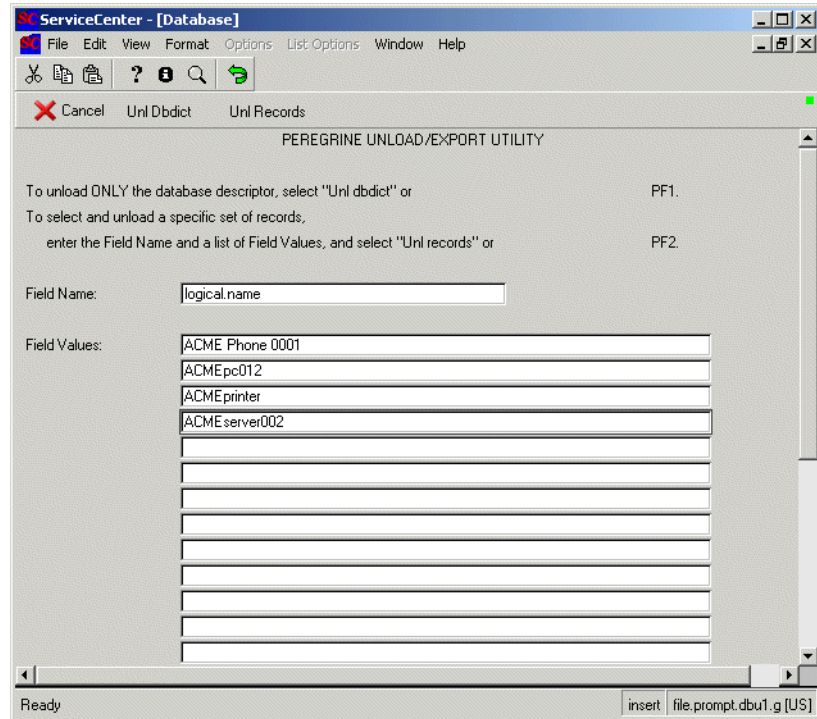


Figure 4-4: Value list record selection

5 Click Unl Records.

This displays the list of records which meet the search criteria defined on the previous screen (Figure 4-5 on page 108).

6 Verify that the QBE list of records is accurate.

- 7 Select **Options > Mass Unload** from the menu bar, to begin the process of unloading the list of records.

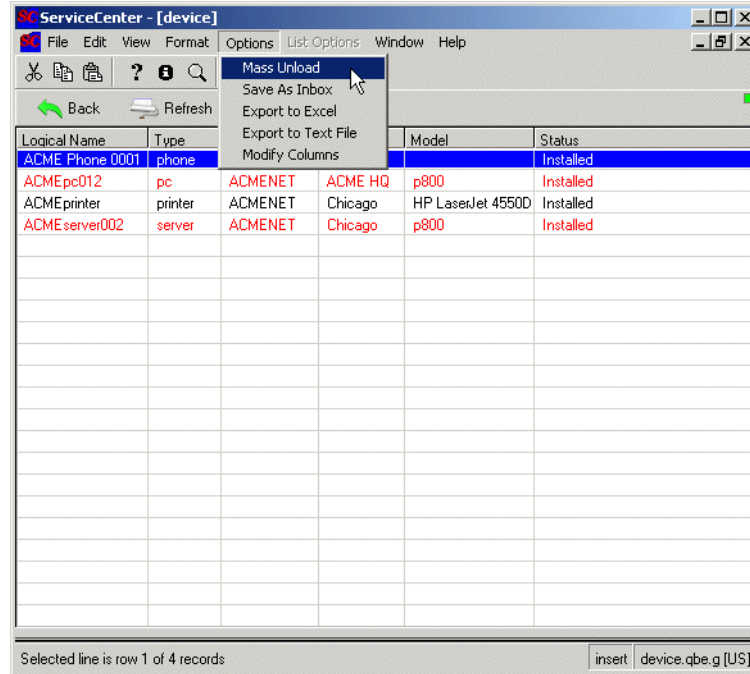


Figure 4-5: List of assets matching the search criteria

The Unload/Export Facility form is displayed with a prompt requesting unload parameters.

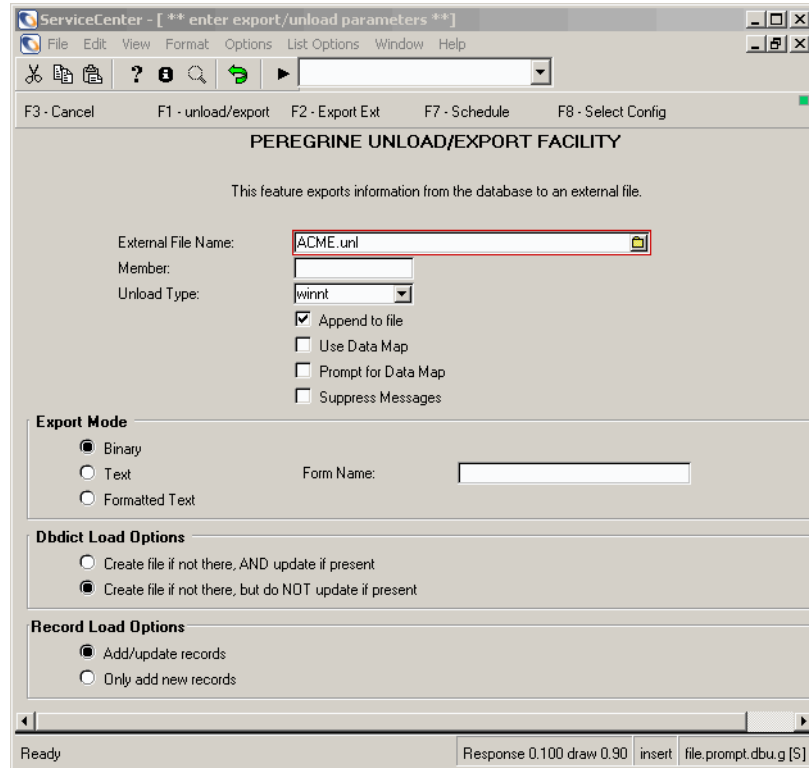


Figure 4-6: Configuring multiple record unload

Unload parameters fall into two categories:

- Those which control how data is written to an external dataset (unload options), and
- Those which control how the same data is to be read BACK into ServiceCenter via the load utility (load options).

Both categories of options are set prior to the actual unload.

Unload options control the immediate allocation of the external dataset and the format of the data written to the external dataset. The load options are written to the external dataset along with the unloaded record(s). These load options remain dormant until the external dataset is accessed by the Load Utility at which time they control how the unloaded data is read back into ServiceCenter.

- 8 To continue with this example, using the default unload parameters, enter the fully-qualified dataset name to which ServiceCenter will write this single device record.
 - Enter <qualifier>.example3, where <qualifier> conforms to your standard naming conventions for OS/390 datasets.
 - or-
 - Enter <path>.example3, where <path> conforms to your standard naming conventions for Unix or Windows NT/2000 files.
- 9 Click **Unload/export**, to begin the unload.

Unload options control the immediate allocation of the external dataset and the format of the data written to the external dataset.

The load options are written to the external dataset along with the unloaded record(s). These load options remain dormant until the external dataset is accessed by the Load Utility, at which time they control how the unloaded data is read back into ServiceCenter.

The unload is executed as a foreground process, i.e. session is devoted to executing the unload, and terminal control is returned only after the unload has completed. The Database Manager unload format remains on-screen while the unload is processed.

Upon completion, the device record list is re-displayed along with several possible messages displayed at the bottom of the screen (unload messages are also written to the *msglog* file).

Note: When scrolling unload messages from a QBE list, enter a scroll setting of **half** or **page**, otherwise a scroll setting of **recd** will cause the QBE list itself to be scrolled, causing the unload messages to be lost.



The message icon can also be used to display messages (blue indicates a call to action, black indicates informational only, and red indicates that it's an error message).

After a successful unload, the file is available for loading it into another system. Refer to *Loading ServiceCenter Databases* on page 127.

Unloading Records from Multiple Databases

A user may want to copy several device database records and the entire *tpdevice* database from a production ServiceCenter environment into a development environment.

Accomplishing this efficiently involves an unload of the desired device records followed by an unload of the *tpdevice* records into the same OS/390, Unix, or Windows NT/2000 file, followed by a load of the OS/390, Unix, or Windows NT/2000 file from within the development system.

Records from different databases can be unloaded into the same OS/390, Unix, or Windows NT/2000 file, because destination information is included with each unload. The destination information consists of the dbdict of the unloaded database.

At load time, the **load** function examines the unloaded dbdict for a group of records to determine where to route them. Using the example of *device* and *tpdevice* records mixed within one OS/390, Unix, or Windows NT/2000 file, at load time the *device* records are directed to the *device* database and the *tpdevice* records are directed to the *tpdevice* database automatically.

An important consideration when unloading multiple databases into one OS/390, Unix, or Windows NT/2000 file is that at load time, the entire external file is loaded. There is no option to selectively choose to load some records and not others.

Important: If the user needs to be selective about when and where records are to be loaded, the records must be unloaded into separate OS/390, Unix, or Windows NT/2000 files.

Part I — Unload from the First File

To unload from the first file:

- 1 Access the *device.g* form.
- 2 Bring up a list of device records to be unloaded.
For this example call up all records for ACME assets.
- 3 Enter ACME in the Asset field.

4 Click Search.

Note: Use a query that is appropriate for your database.

A list of device records satisfying the query is displayed (Figure 4-5 on page 108).

5 Select **Options > Mass Unload**, to begin the unload process for this list of records.

Note: If you are using a record list, select **List Options > Mass Unload**.

The Unload/Export Facility form is displayed with a prompt requesting unload parameters (Figure 4-6 on page 109).

Unload parameters fall into two categories:

- Those which control how data is written to an external dataset (unload options), and
- Those which control how the same data is to be read BACK into ServiceCenter via the load utility (load options). Both categories of options are set prior to the actual unload.

Unload options control the immediate allocation of the external dataset and the format of the data written to the external dataset.

The load options are written to the external dataset along with the unloaded record(s). These load options remain dormant until the external dataset is accessed by the Load utility at which time they control how the unloaded data is read back into ServiceCenter.

6 Enter the fully-qualified OS/390 dataset name, Unix, or Windows NT/2000 file name to which ServiceCenter will write these device records

- Enter `<qualifier>.EXAMPLE4`, where `<qualifier>` conforms to your standard naming conventions for OS/390 datasets, Unix or Windows NT/2000 files.

7 De-select the **Append to File** check box, giving it a *false* value to be sure the records will be unloaded to an empty OS/390 dataset name, Unix, or Windows NT/2000 file.

8 Click **Unload/export**, to begin the unload.

The unload is executed as a foreground process, meaning the ServiceCenter terminal session is devoted to executing the unload, and terminal control is returned after the unload is completed. The Database Manager *unload* format remains on your screen while the unload is processed.



Upon completion, the device QBE list is re-displayed along with several possible messages displayed at the bottom of your screen (unload messages are also written to the *msglog* file).

You can also click the message icon to display messages (blue indicates a call to action, black indicates informational only, and red indicates that it's an error message).

After a successful unload, the file is available for loading it into another system. Refer to *Loading ServiceCenter Databases* on page 127.

Part II — Unload from the Second File

To unload from the second file:

- 1 Exit back to the Database Manager dialog box.
- 2 Enter =tpdevice in the Form field.

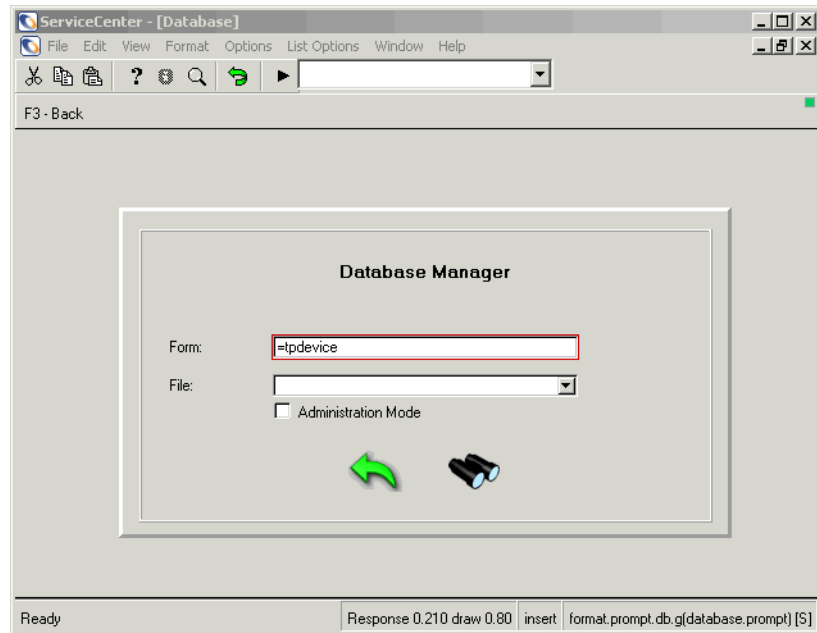
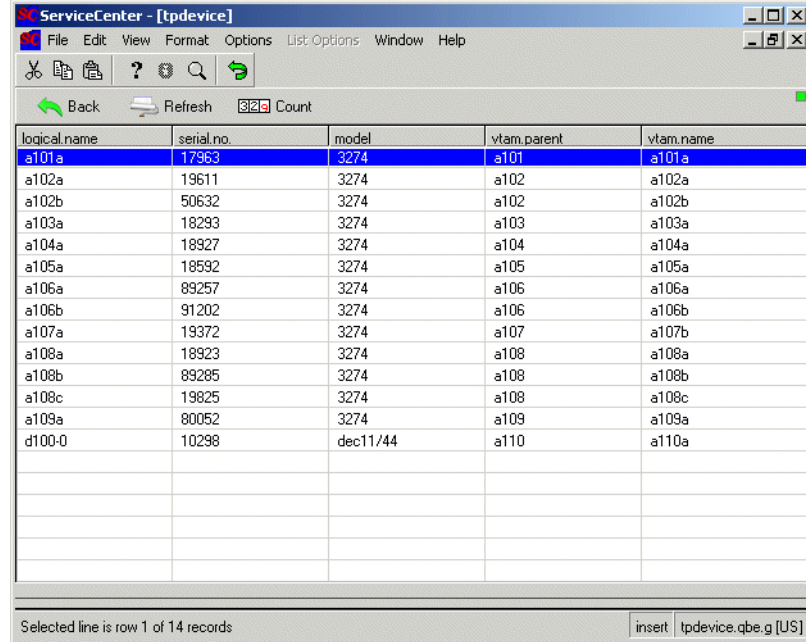


Figure 4-7: Database Manager dialog box

- 3 Press Enter.
The *tpdevice* (Teleprocessing Devices) form is displayed.

- 4 Pass a *true* query by pressing Enter to return a list of all records (or use any query to retrieve the desired records).



logical_name	serial.no.	model	vlam.parent	vlam.name
a101a	17963	3274	a101	a101a
a102a	19611	3274	a102	a102a
a102b	50632	3274	a102	a102b
a103a	18293	3274	a103	a103a
a104a	18927	3274	a104	a104a
a105a	18592	3274	a105	a105a
a106a	89257	3274	a106	a106a
a106b	91202	3274	a106	a106b
a107a	19372	3274	a107	a107b
a108a	18923	3274	a108	a108a
a108b	89285	3274	a108	a108b
a108c	19825	3274	a108	a108c
a109a	80052	3274	a109	a109a
d100-0	10298	dec11/44	a110	a110a

Selected line is row 1 of 14 records

insert tpdevice.qbe.g [US]

Figure 4-8: Unloading a second group of records

- 5 Select **Options > Mass Unload** to begin the unload process on these *tpdevice* records.

Note: If you are using a record list, select **List Options > Mass Unload**.

This unloads every record in the *tpdevice* file.

The default unload screen is displayed with a prompt requesting unload parameters.

Note: In addition to the default unload parameters, the previous external file name is retained. If this file name does not appear, re-enter it as the external file name.

- 6 The *tpdevice* records are to be unloaded to this same external dataset, so activate the **Append to (Existing) File** check box, setting its value to *true*.
- 7 Click **Unload/export**, to append the *tpdevice* records to the end of the OS/390 dataset, or Unix or Windows NT/2000 file containing the device records.

The unload is executed as a foreground process, meaning the ServiceCenter terminal session is devoted to executing the unload, and terminal control is returned after the unload is completed. The Database Manager *unload* format remains on your screen while the unload is processed.

Upon completion, the *tpdevice* QBE list is re-displayed along with several possible messages displayed at the bottom of the screen (unload messages are also written to the *msglog* file).

After a successful unload, the file is available for loading it into another system. Refer to *Loading ServiceCenter Databases* on page 127.

Unloading a QBE Only (No Data Records) to Excel

A user may want to export only the list of records which meet a particular search criteria to an Excel file. This list is obtained by passing your query into the Database Manager, which returns a QBE list. ServiceCenter Database Manager uses Microsoft's (MS) DDE to export the QBE into an Excel format.

Note: The Export to Excel option is available only on Windows clients.

To unload a QBE list to Excel:

- 1 Access the device form.
- 2 Enter the query values in the appropriate fields to search for records containing specific information.
- 3 Press Enter to pass the query.

This query returns a list of records in the device file containing values which match the criteria.

4 Select Options > Export to Excel.

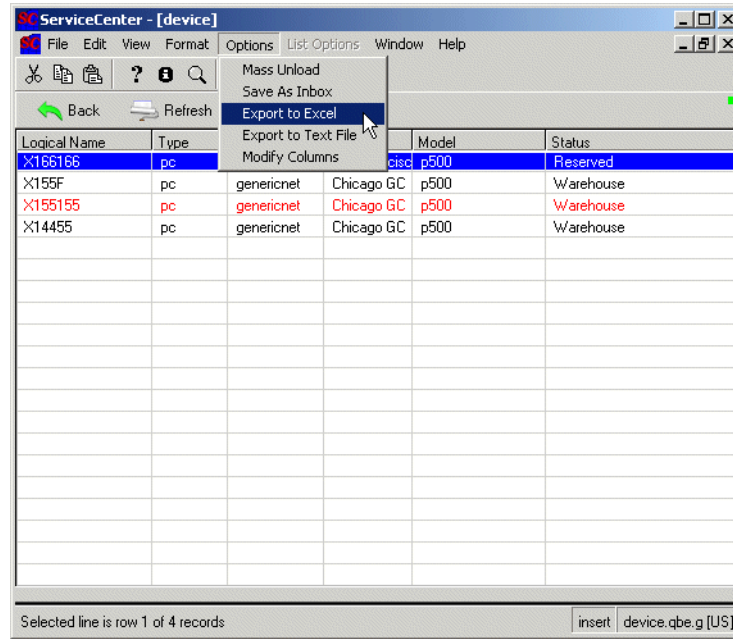


Figure 4-9: Unloading data records to Excel

Note: When running Microsoft Office, the Microsoft Excel application opens, and the data from the ServiceCenter record list is displayed in a table within the Excel application.

- 5 Save this file in Excel.
- 6 Return to ServiceCenter to pass additional queries or to perform other functions.

Unloading a Database Dictionary Only (No Data Records)

A user may want to move a database file tested and created in a *development* ServiceCenter system into a *production* system. The development database may contain test records, therefore only the database dictionary (the structure of the file) should be migrated and not the records within.

The user may also want to use a load to make additions to a *production* database dictionary file.

Important: As a precautionary measure, backup the unmodified database dictionary file (i.e., the database dictionary which will be replaced on the target ServiceCenter system) before proceeding.

Both examples involve unloading a database dictionary only.

To unloading the device Database Dictionary without unloading any device records:

- 1 Access the device form.
- 2 Without querying for records at this point, pull down the **Options** menu.
- 3 Select the **Export/Unload** option, to begin the unload process.

The unload file form is displayed, requesting the fields values of the records you wish to unload or export.

- 4 *Do not* enter values in this form. These values refer to actual values found in a specific field, as defined in records in this Database Dictionary file. Since you are going to unload the structure of the Database Dictionary file but not the records, there is no need to specify either a field or field values.

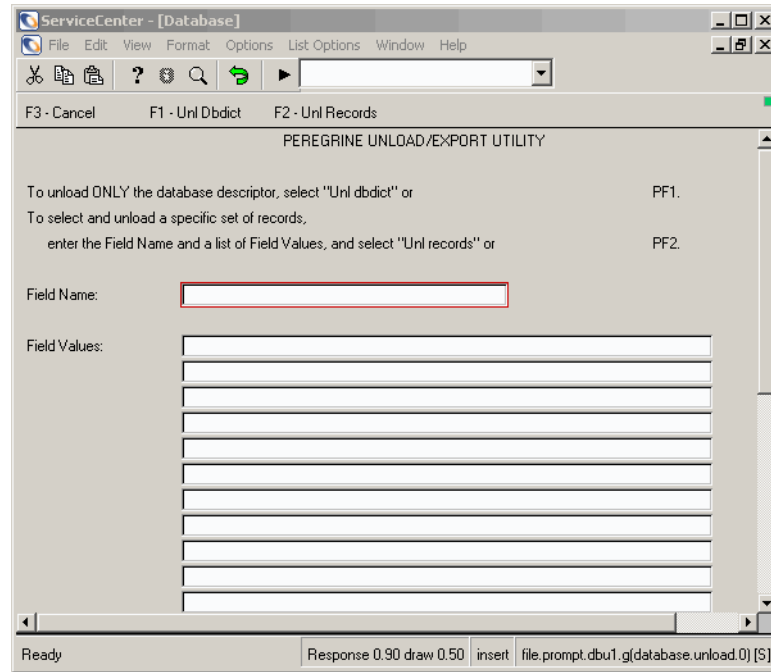


Figure 4-10: Unloading a dbdict

- 5 Click **Unl Dbdict**.

The Unload/Export Facility form (Figure 4-6 on page 109) is displayed requesting unload parameters.

- 6 To continue with this example, enter the fully-qualified OS/390 dataset name, or Unix or Windows NT/2000 file name to which ServiceCenter will write the device dbdict.

Enter `<qualifier>.EXAMPLE5`, where *<qualifier>* conforms to the standard naming conventions for OS/390 datasets or Unix files.

- 7 De-select the **Append to File** check box, giving it a value of *false* to be sure it is unloaded to an empty OS/390 dataset or Unix file.

8 Click Unload/export.

The unload is executed as a foreground process, meaning the ServiceCenter terminal session is devoted to executing the unload, and terminal control is returned after the unload is completed. The Database Manager *unload* format remains on your screen while the unload is processed.

Upon completion, the blank device form is re-displayed with a message at the bottom of your screen stating that *0 records were unloaded*. This is fine, since none should have been because you were unloading the structure instead of the data records.

After a successful unload, the file is available for loading into another system. Refer to *Loading ServiceCenter Databases* on page 127.

Unloading a ServiceCenter Report Writer Report

Reports created with the ServiceCenter ReportWriter Utility consist of records from at least two files: *report* and *format* with possible Format Control record(s) in the *formatctrl* file and possible link records in the *link* file.

Another, easier way to unload all records from these four related files is by using the ServiceCenter Unload Script Utility. See Chapter 10, *Unload Script Utility* for details.

Part I — Report Structure Unload

To unload the *invoice.approved.acct* report using the Database Manager Unload/Export capability:

- 1 Access the Database Manager.
- 2 Enter `=report.prompt.rw` in the **Form** field of the Database Manager dialog box.
- 3 Press Enter.

The Report Writer dialog box is displayed.

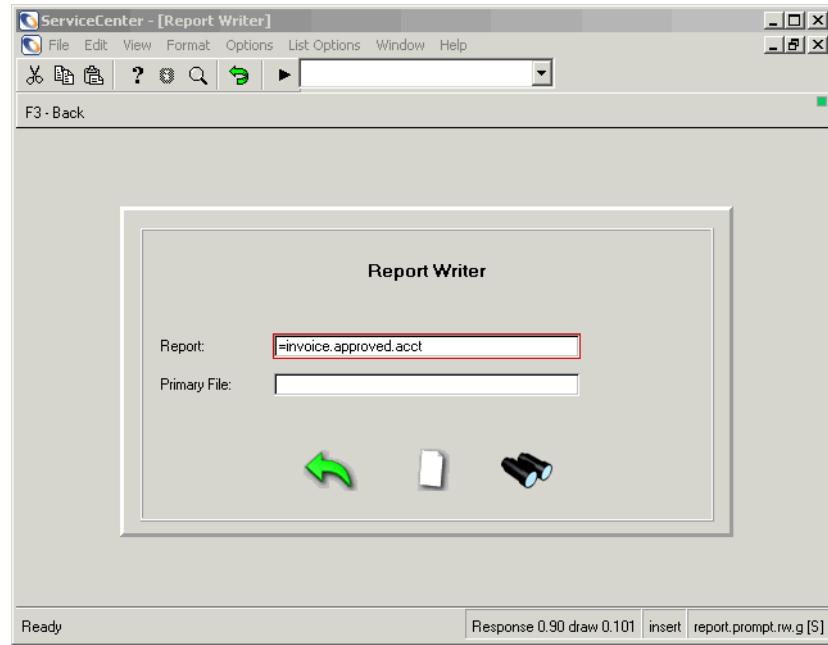


Figure 4-11: Report Writer dialog box

- 4 Enter =invoice.approved.acct in the **Report** field.
- 5 Click Search.

The report record name and primary file designation are displayed.

6 Select **Options > Export/Unload** to unload the report record.

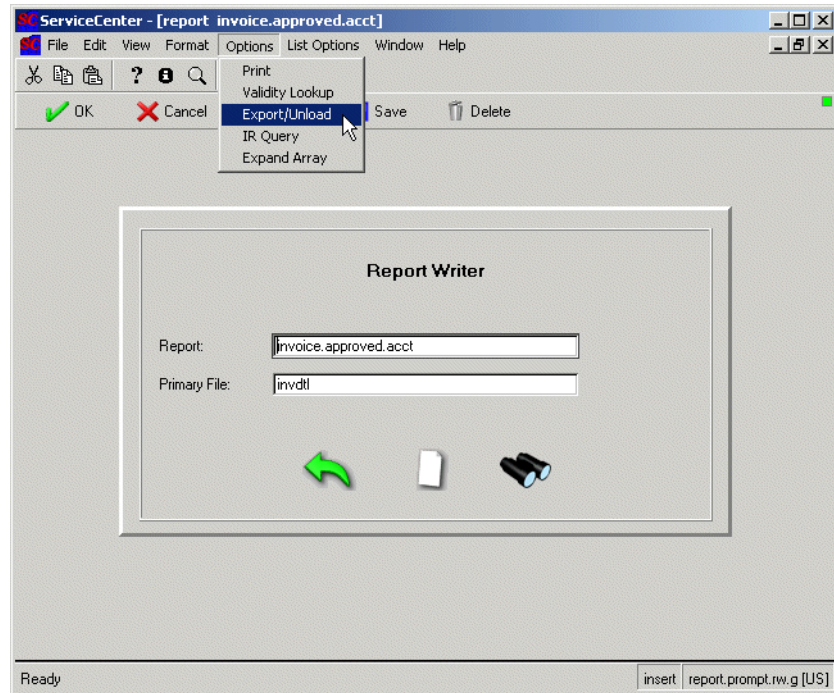


Figure 4-12: Report Writer prompt

The *Unload/Export Utility* screen (Figure 4-6 on page 109) is displayed with a prompt requesting unload parameters.

- 7 To continue with this example, enter the fully-qualified OS/390 dataset name, Unix or Windows NT/2000 file name to which ServiceCenter will write the report record
 - Enter *<qualifier>.EXAMPLE6*, where *<qualifier>* conforms to your standard naming conventions for OS/390 datasets, Unix, or Windows NT/2000 files.
- 8 De-select the **Append to File** check box, giving it a value of *false* to be sure it will be unloaded to an empty OS/390 dataset, Unix, or Windows NT/2000 file.

9 Click Unload/export.

The unload is executed as a foreground process, meaning the ServiceCenter terminal session is devoted to executing the unload, and terminal control is returned after the unload is completed. The Database Manager *unload* format remains on your screen while the unload is processed.

Upon completion, several possible messages are displayed at the bottom of your screen (unload messages are also written to the *msglog* file).

After a successful unload, the file is available for loading it into another system. Refer to *Loading ServiceCenter Databases* on page 127.

Part II — Report Record Unload

The next step of the process is to exit from the report form and add records associated with this report from the *format* file.

To add records:

- 1 Return to the Database Manager dialog box.
- 2 To unload forms associated with this report, enter =format in the **Form** field, and click **Search**.

The format form is opened in the Database Manager.

The six forms referenced by this report can be unloaded via the *Value-List* method (see *Unloading Multiple Records from a Single File Using Value-list* on page 105).

- 3 Do not place data in any of the fields or query for any specific records.

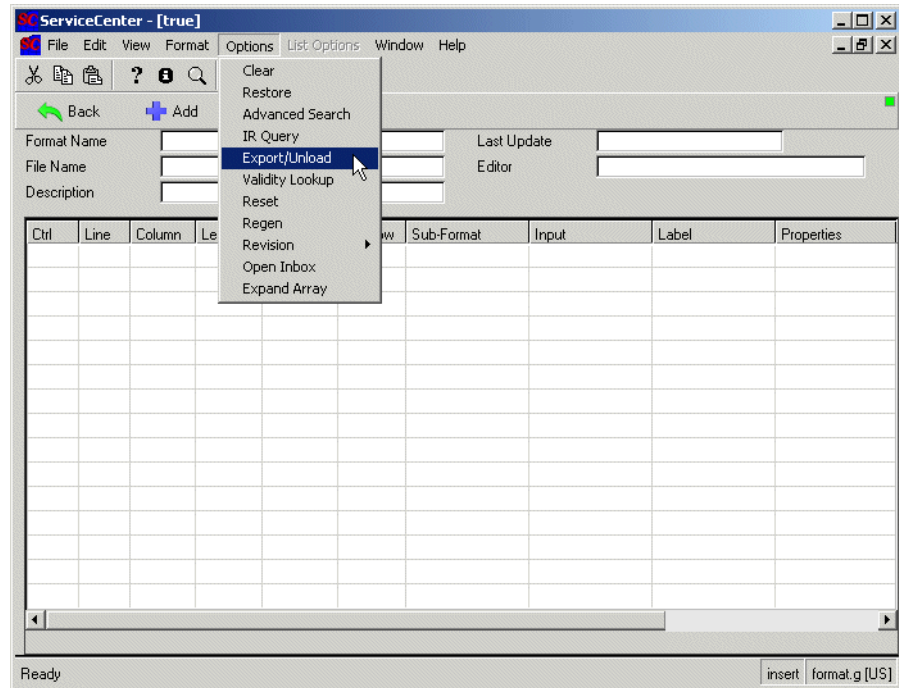


Figure 4-13: Format record

- 4 Select **Options > Export/Unload** from a blank *format* form.
The Unload/Export Facility form is displayed.
- 5 Enter name in the **Field Name** field.

6 Enter the following six format names in the Field Values array:

invoice.acct.header	invoice.acct.grand
invoice.acct.details	invoice.details
invoice.acct.totals	invoice.select

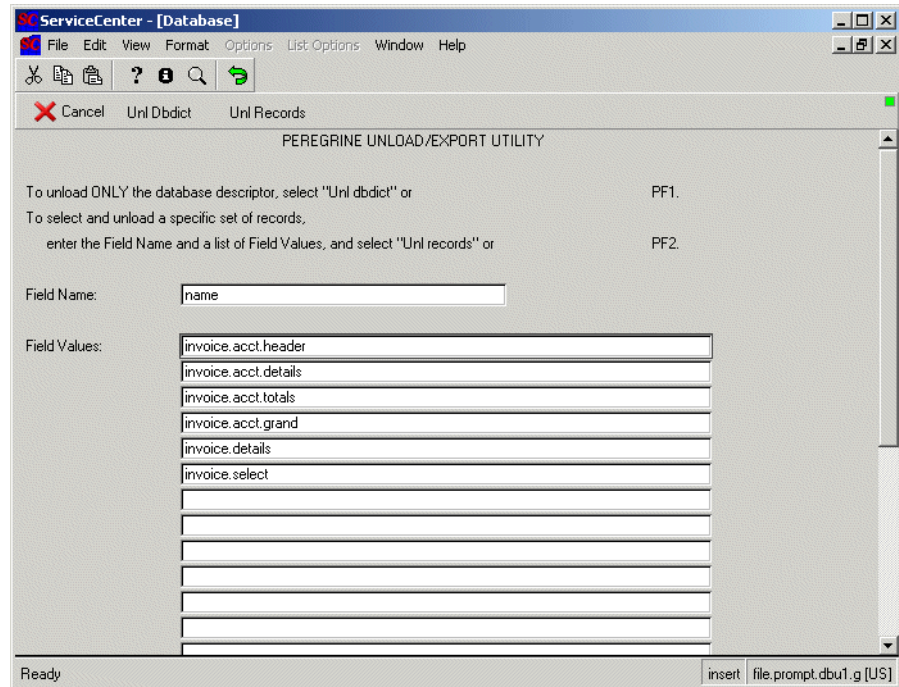


Figure 4-14: Listing reports to unload

7 Click **Unl** records.

A record list of the selected forms is displayed.

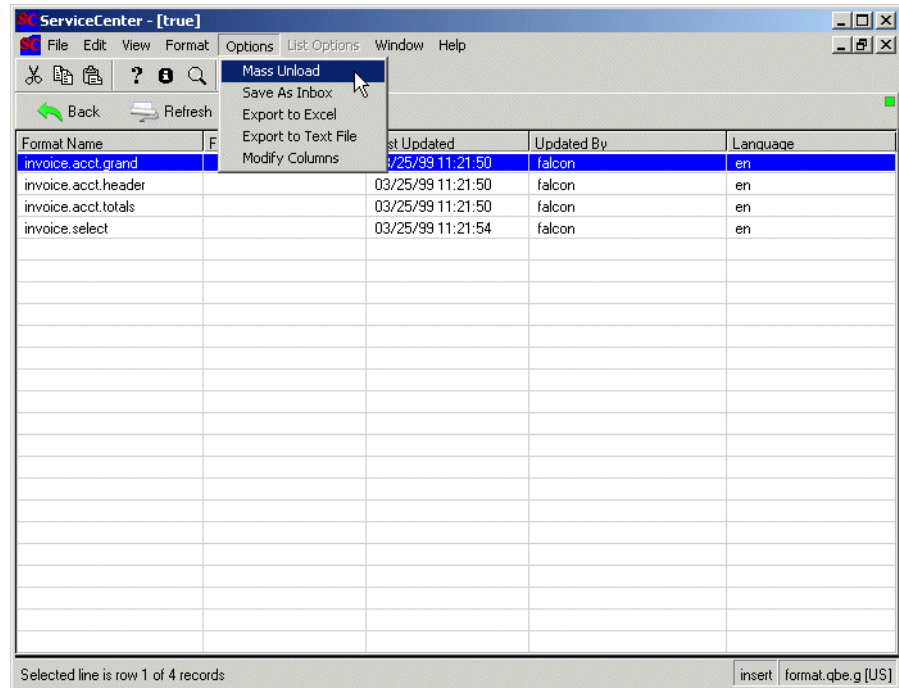


Figure 4-15: QBE list of formats to unload

8 Select **Options > Mass Unload** to unload the *format* records.

The Unload/Export Facility screen is displayed requesting unload parameters.

- 9 Leave the **Append to File** check box activated since the format records should be appended to this same file.
- 10 Click **Unload/export**.

Part III—Format Control

To unload Format Control records associated with this report:

- 1 Return to the Database Manager dialog box.
- 2 Enter =formatctrl in the **Form** field, and click **Search**.

Note: The Format Control records referenced by this report can be unloaded via the *Value-List* method.
- 3 Without placing data in any of the fields, select **Options > Export/Unload**. The Unload/Export Utility descriptor screen is displayed requesting unload values (Figure 18-37).
- 4 Enter name in the **Field Name** field.
- 5 Enter the following two Format Control names in the **Field Values** array.
 - invoice.detail
 - invoice.select

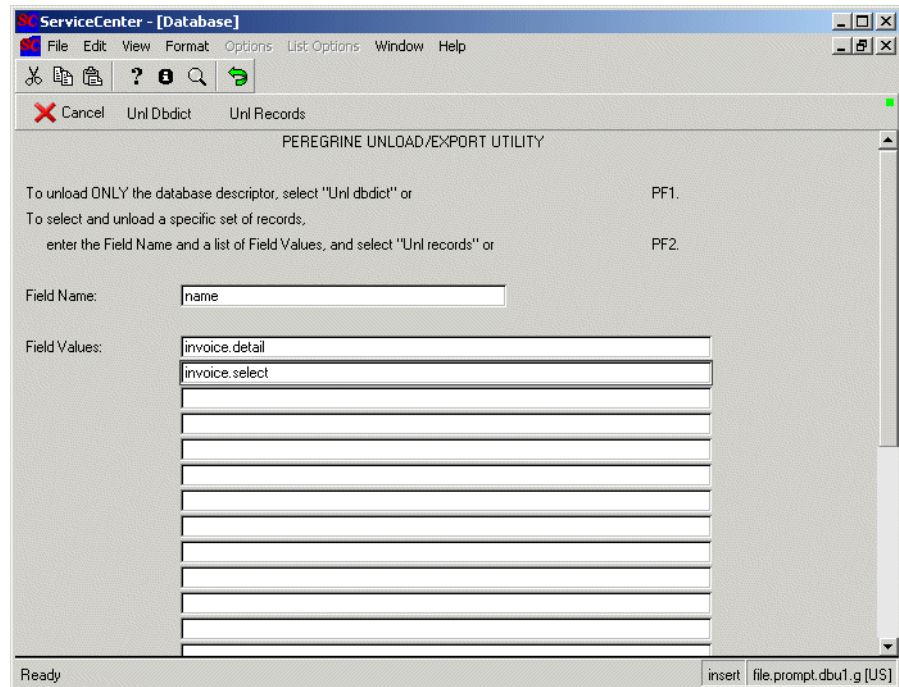


Figure 4-16: Specifying Format Control records to unload

6 Click **Unl records**.

A record list of the selected Format Control records is displayed.

7 Select **Options > Mass Unload** to unload the `formatctrl` records.

The Unload/Export Facility form is displayed with a prompt requesting unload parameters.

Note: The **External File Name** value is retained from the previous unload.

8 Leave the **Append to File** check box activated since the `formatctrl` records are to be appended to this same file.**9** Click **Unload/export**.**Part IV—Link Records**

The next step is to unload link records associated with this report. Repeat the same process you used to unload the `formatctrl` records, but access the link file instead. The two link records you need to unload are *invoice.header* and *invoice.detail*.

Loading ServiceCenter Databases

The **Load** function reads data into the current ServiceCenter system from an existing dataset or file containing data unloaded from a ServiceCenter P4 file system. The **Load** function will not read any external dataset that is not in ServiceCenter unload format.

Note: Load function security is enabled via Format Control attached to the *format.prompt.db* form.

The options chosen when executing a load of an external dataset are:

- Name of the external dataset.
- Run the load in *foreground* or *background*.
- View contents of load file.

All other load options were defined when the file was originally unloaded and these options are included in the external dataset itself. These load options govern how the data is loaded and cannot be overridden at load time.

The preceding section of this chapter describes how to unload records from a ServiceCenter database or databases into an external dataset or file. The following section describes how to load such a file, or a file created by the ServiceCenter Unload Script Utility, back into ServiceCenter.

Executing the Load as a Foreground Process

To load an external file:

- 1 Access the Database Manager.
- 2 Leave all fields in the Database Manager dialog box blank.
- 3 Select **Options > Import/Load**.
- 4 When the new form opens, enter the fully qualified dataset or file name of the external file you wish to load.
<qualifier>. EXAMPLE1.
- 5 If you specified a member for the unload, enter its value in the **Member** field.
- 6 Leave the **Import Descriptor** record field empty. This is used only for imports.
- 7 Choose the appropriate message activity level during the load process. Loads that include many large files can generate many messages, so you can choose whether to receive *All Messages*, *Totals Only*, or *None* for the entire load process.

- 8 To view the contents of the file(s) to be loaded, click **List contents**.

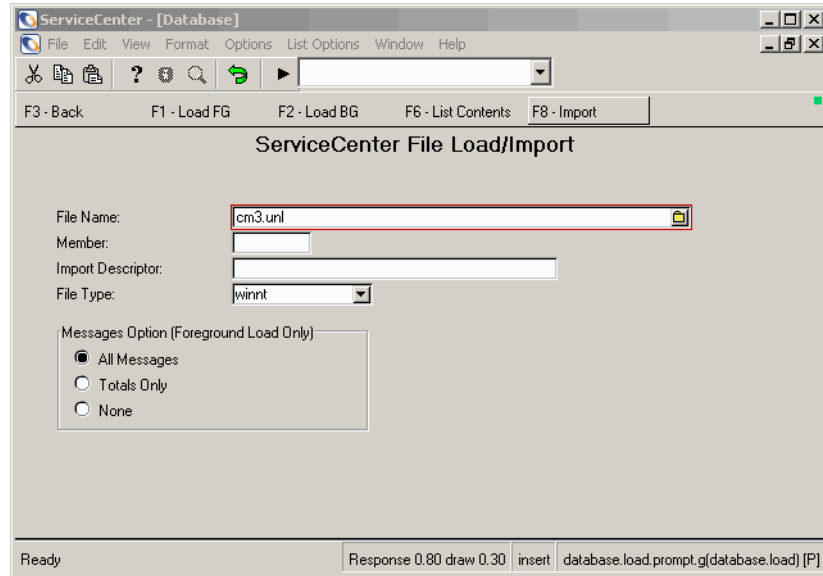


Figure 4-17: Selecting a ServiceCenter unload file to load

The field values for each record unloaded are displayed in a table. This provided you the opportunity to scroll through the data unload to verify it has been formatted properly.

- 9 If an error has occurred or the incorrect data has been loaded into the file, repeat the unload process again.

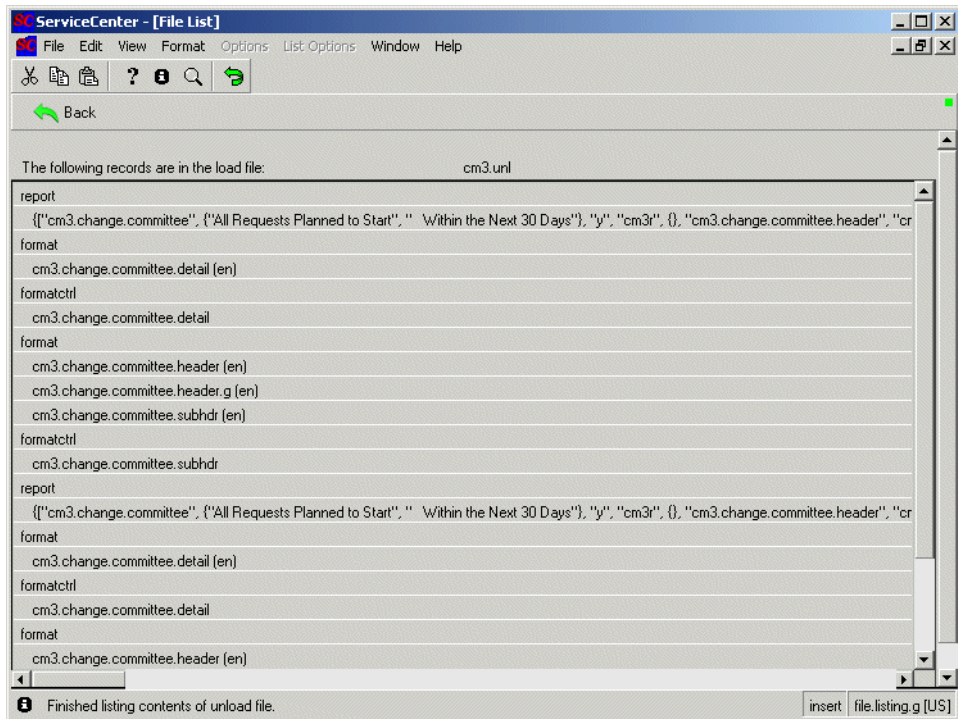


Figure 4-18: Contents table of unload file data records

- 10 Click **Back** to return to the load/import screen.
 - 11 To run the load processes, click **Load fg**.
- Note:** The import button functions are described in *Export Utility* on page 137 and *Import Utility* on page 171.

The load is executed as a foreground process, meaning the ServiceCenter terminal session is devoted to executing the load, and terminal control is returned after the load is completed. The Database Manager *load* format remains on your screen while the load is processed.

The system first attempts to connect to the external dataset and immediately generates a message if it cannot.

- If the dataset does not exist, the message *Error occurred while loading external file: <qualifier>.EXAMPLE1* is displayed.
- The same message is displayed if the dataset exists but cannot be accessed.
- In an OS/390 environment, this is usually due to the intervention of a security facility such as Top Secret, RACF, or ACF2 which require a certain high-level qualifier on the dataset name.
- In a Unix or Windows NT/2000 environment, it can mean that permissions on the directory or file do not allow access for this user.

After the external file is successfully connected, the load utility proceeds with loading the contents of the external file.

When it is completed, the Database Manager dialog box is displayed with messages indicating the following pieces of data:

- Number of records loaded
- Files into which records were loaded



Any other load messages appear on the screen and in the *msglog* file, which can be seen by clicking the message icon black indicates information only, blue indicates a call to action, and red indicates an error message).

Load Complication Messages

Usually, several messages are generated by a load.

- In the OS/390 environment, the following load messages are produced when loading the dataset *<qualifier>.example1*:
 - **<qualifier>.example1 file loaded.*
 - **File(s) loaded from external file:<qualifier>.example1*
 - 1 record loaded into device file from external file: <qualifier>.example1*
- In the Unix environment, the following load messages are produced when loading the file *path>.example1*:
 - **<path>.example1 file loaded.*
 - **File(s) loaded from external file:<path>.example1*
 - 1 record loaded into device file from external file: <path>.example1*

- In the Windows NT/2000 environment, the following load messages are produced when loading the file *path>.example1*:

**<path>.example1 file loaded.*

**# total records processed in 00:00:00*

**# records from <path>.example1 updated to device*

**# records from <path>.example1 added to device*

**# errors detected loading to device from <path>.example1*

**# records from <path>.example1 loaded to device, 00:00:00 elapsed*

Loading device file for external file: <path>.example1

Note: The on-line message facility displaying the load messages on the screen displays the most recent message first. As you scroll through the load messages on-line, they are therefore displayed in reverse chronological order.

Message	Description
<MVS.FILENAME>, <WINNT.FILENAME> or <UNIX.FILENAME> file loaded.	Load processing of the named dataset is complete.
File(s) loaded from external file: <MVS.FILENAME>, <WINNT.FILENAME> or <UNIX.FILENAME>	Load processing of all the files within the dataset or file completed.
<nn> records loaded into <database name> from external file: <MVS.FILENAME>, <WINNT.FILENAME> or <UNIX.FILENAME>	Load processing of nn records into the named ServiceCenter database from the named dataset or file completed.
Loading <database name> from external file: <MVS.FILENAME>, <WINNT.FILENAME> or <UNIX.FILENAME>.	Load processing of the named ServiceCenter database from the named dataset or file begins.
<nnn> 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 ServiceCenter database.

Message	Description
The config record (database load/unload is missing).	The hardware configuration record <i>database load/unload.unix</i> required for database loads and unloads has been deleted or renamed. It must be replaced. Refer to <i>Unload/Load</i> on page 93 for more information.
Invalid number of records found in config file.	The load utility found two or more hardware configuration records named <i>database load/unload.mvs</i> . There should be only one such record.
Config record is not set up to be used for database loads.	The true/false field Used for database loads/unloads on the hardware configuration record <i>database load/unload.mvs</i> is set to <i>false</i> . It must be <i>true</i> .
The external file <MVS.FILENAME>, <WINNT.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 exclusive control on it.
Invalid end of file encountered in external file <MVS.FILENAME>, <WINNT.FILENAME> or <UNIX.FILENAME>.	The load utility encountered an incomplete dataset, i.e., the unload which originally created this dataset was interrupted before completion.
All null keys:file: <database name>;record;<nn> in external file:<MVS.FILENAME>, <WINNT.FILENAME> or <UNIX.FILENAME>.	The load utility attempted to add record number <nn> to the named ServiceCenter database and encountered all null keys.

Message	Description
Invalid null key;file:<database name>;rec no:<nn> in external file <MVS.FILENAME>, <WINNT.FILENAME> or <UNIX.FILENAME>.	The load utility attempted to add record number <nn> to the named ServiceCenter database and encountered an invalid null key.
Invalid duplicate key;file:<database name>;rec no:<nn> in external file <MVS.FILENAME>, <WINNT.FILENAME> or <UNIX.FILENAME>.	The load utility attempted to add record number <nn> to the named ServiceCenter database and encountered an invalid duplicate key.
External file <MVS.FILENAME>, <WINNT.FILENAME> or <UNIX.FILENAME> is not in Peregrine Four Unload Format.	The named dataset is not in the proprietary format required by the load utility.

Executing the Load as a Background Process

A background load is executed independently of the terminal session, making the terminal session available for other functions as soon as the load has been scheduled.

To execute a load:

- 1 Select the Toolkit tab in the System Administrator's home menu.
- 2 Click **Database Manager**.
The Database Manager dialog box is displayed.
- 3 Select **Options > Import/Load**.
- 4 Enter the fully qualified dataset or file name of the external file you wish to load, <qualifier>. EXAMPLE1.
- 5 If you specified a member for the unload, enter its value in the **Member** field.
- 6 Leave the **Import Descriptor** field blank (for imports only).
- 7 Click **Load bg**.

Note: The import button functions are described in *Export Utility* on page 137 and *Import Utility* on page 171.

OS/390 Note: The system first attempts to connect to the external dataset and immediately generates a message if it cannot.

- If the dataset does not exist, the message *Error occurred while loading external file: <qualifier>.example1* is displayed.
- The same message will appear if the dataset exists but cannot be accessed, usually due to the intervention of a security facility such as Top Secret, RACF, or ACF2, which requires a certain high-level qualifier on the dataset name.

After the external file is successfully connected, the Schedule Background Load form is displayed. The **Expiration Date** value defaults to the current date and time which may be overwritten.

The screenshot shows a window titled "ServiceCenter - [ServiceCenter]" with a menu bar (File, Edit, View, Format, Options, List Options, Window, Help) and a toolbar. Below the toolbar is a status bar with "End", "Load BG", and "Menu" buttons. The main area is titled "SCHEDULE BACKGROUND LOAD" and contains the following fields:

- Expiration Date:
- Repeat Interval:
- External Filename:
- Member:
- Suppress Messages?:

The status bar at the bottom left shows "Ready" and the bottom right shows "insert | bgload.prompt.g [US]".

Figure 4-19: Scheduling background load

- 8 Enter a time in the **Repeat Interval** field to repeat the load automatically at specified times.

The example of 1:00:00 causes the load to occur at intervals of one hour, as well as at the current time (or time shown in the **Expiration Date** field).

- 9 Change the path and/or name of the file in the **External Filename** field if necessary.

This value defaults to the entry provided on the previous form (Figure 4-19 on page 135).

- 10 Click **Load BG**.

The Database Manager prompt screen re-displays with the message *Background load schedule to run at <date/time>*.

Note: Load messages are written to the msglog file when the actual load is executed.

5 Export Utility

CHAPTER

This chapter introduces the Export Utility and explains how to export any field in any ServiceCenter file, including multiple lines in arrays.

This chapter is organized into the following main sections.

- *Overview of the Export Function* on page 138.
- *Rules Used in the Export Utility* on page 139.
- *Accessing the Export Utility* on page 140.
- *Exporting Records Example* on page 141.
- *Creating an Export Descriptor Record* on page 148.
- *Exporting Records in the Foreground* on page 153.
- *Exporting Records in the Background* on page 159.
- *Printing an Export Descriptor Record* on page 161.
- *Using Format Control with Export* on page 162.

Overview of the Export Function

Although ServiceCenter may be the central repository for your enterprise infrastructure data, you may need to use ServiceCenter information in external programs, such as reporting packages. ServiceCenter provides multiple ways to export text data in formats that are readable by non-ServiceCenter applications. In addition to the Export Utility, ServiceCenter also supports:

- Text file and Excel file exports via DDE and DDE-like capabilities. You access the Export to Text File and Export to Excel options via the List Options menu (when viewing a QBE list) or via the Options menu (when not viewing a QBE list).

For more information, see the DDE chapter in *ServiceCenter System Tailoring*. Also see *Exporting to Excel in a Different Language* on page 169 to modify your display options to export a file to Excel in a different language.

Note: The Export to Excel option is available only on Windows clients.

- Formatted text unload option. This lets you use a ServiceCenter text (not Windows) format to define both the fields to export and how to format the external data. This facility is accessed in the Database Manager via the Unload facility, Formatted Text Export mode.

The Export Utility allows you to export any field in any ServiceCenter file, including multiple lines in arrays.

The following rules generally apply to the Export Utility. You can:

- Export in either foreground or background.
- Use Format Control to initialize data and perform calculations prior to writing the record.
- Use field-specific output routines to manipulate data.
- Export single or multiple line formats.
- Export both labels and data.

Using the Export Utility involves defining the data output via either an export descriptor record or a ServiceCenter format, and then executing the task.

To export data:

- 1 Either build an export descriptor record or use the **Format Utility** to build an exact image of what the output should look like.
- 2 Use **Format Control Maintenance** to define any special processing or validation to be done prior to exporting the data.
- 3 Use the **Format Utility** to apply any output routines required to manipulate field values (for format-controlled exports only).
- 4 Use **Database Manager** to select the data to export.
- 5 Use the **Database Manager Export/Unload** option to access the export function.
- 6 The **Export/Unload** function performs an export using the procedures described in *Rules Used in the Export Utility* on page 139.

Rules Used in the Export Utility

The Export Utility outputs information in different forms, depending on the environment in which it is running.

- In an OS/390 environment, ServiceCenter exports one QSAM record for each line of a Formatted Text unload or standard Export (i.e., using an export descriptor record). ServiceCenter exports one line per record if the record can fit in the logical record length (lrecl) defined for the dataset. If the data is longer than the lrecl, then it will use more than one line. Peregrine recommends using only one line per exported record, including using only a one-line format for Formatted Text unloads and ensuring that the total export length is less than the logical record length of the target external dataset.
- In a Unix environment, an entire format is treated as a single record without regard to the number of lines in the format. The data is written as a concatenated string of all lines, with an ASCII *0a* delimiter between records.
- In a Windows NT/2000 environment, an ASCII text file is written with each record displayed on a separate line.

In all cases, the fields are fixed length. Fields are normally separated by a space (since that is the character read from the format) but can be separated instead by another character (such as a comma) by replacing the space between fields in the format with a comma (.). There is no requirement for a field separator, however, and you can build a format that has no separation between fields.

Accessing the Export Utility

The Export Utility uses the Database Manager's **unload** function. It is available from QBE lists and data records depending on security parameters established in the `formatctrl` record for the *format* in use.

To export information, you must first select a file using standard ServiceCenter Database Manager procedures.

There are two ways to access the Database Manager:

- Menu option
- Command line

To access the Database Manager from the menu:

- 1 Select the Toolkit tab in the System Administrator's home menu.
- 2 Click Database Manager.

To access the Database Manager from the command line:

- 1 Make sure that the command line is visible by selecting **Command Line** from the **View** menu.
- 2 Type `db` on the command line, and click the **Start Command** arrow (or press Enter).

Exporting Records Example

In the following example, records from the probsummary file will be exported.

To export records:

- 1 Enter the name of a form or file in the Database Manager dialog box.
For this example, type probsummary in the File field of the Database Manager dialog box.

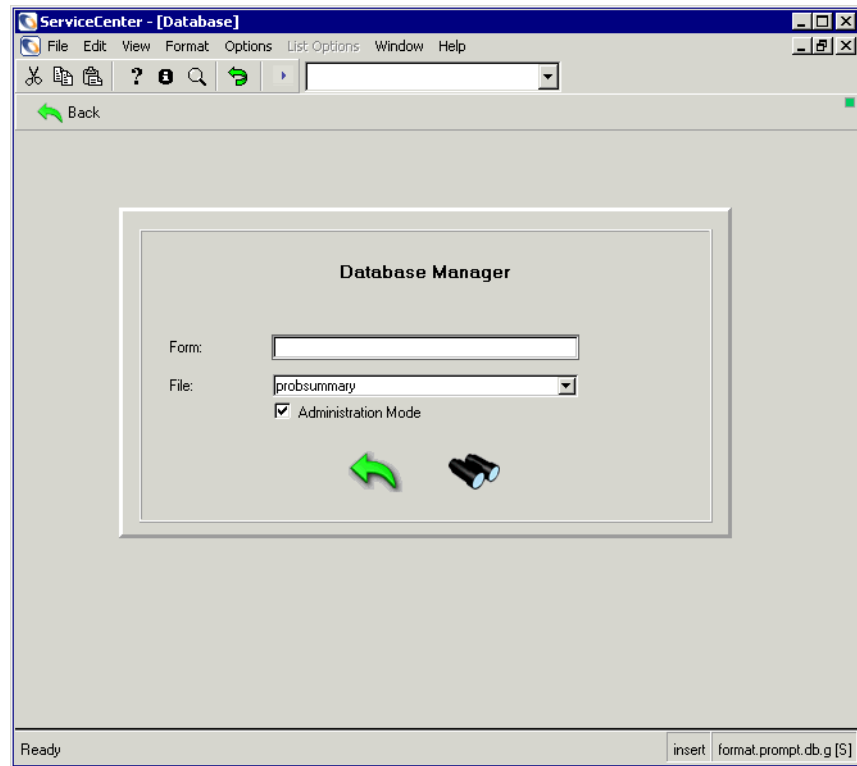


Figure 5-1: Database Manager dialog box

- 2 Click **Search** or press Enter.
A QBE list of formats associated with the probsummary file is displayed.
- 3 Select a format. For this example, select the *apm.search.probsummary* format.

The *search.probsummary.g* form is displayed.

Figure 5-2: Blank probsummary search form opened in Database Manager

- 4 Enter search criteria on the search form, if desired.
For the example, type #IM103 in the Number text box.
- 5 Click Search or press Enter.

The *probsummary.qbe.g* form is displayed.

The screenshot shows the ServiceCenter Database Manager interface with the *probsummary.qbe.g* form filled out. The form is titled "ServiceCenter - [probsummary: IM1030]" and has a menu bar (File, Edit, View, Format, Options, List Options, Window, Help) and a toolbar with buttons for OK, Cancel, Previous, Next, Add, Save, Delete, Views, Find, and Fill. Below the toolbar is a table with columns: ID, Open Time, Update Time, Alert Status, Category, and Title. The table contains two rows of data:

ID	Open Time	Update Time	Alert Status	Category	Title
IM1030	12/23/00 13:4	03/08/01 16:14	alert stage 3	client system	Customer is unable to access draw program. Every time
IM1031	12/29/00 13:4	03/08/01 16:14	alert stage 3	business applic	Customer can't access contact records in SPS by compe

Below the table are tabs for "Basic Incident Search", "Advanced Search", and "IR Query". The "Basic Incident Search" tab is active, showing a search form with the following fields and values:

- Search for Incident Tickets Where:
- Number: IM1030
- Company: PRGN
- Alert Level: alert stage 3
- Category: client system
- Subcategory: software
- Product Type: standard applications
- Problem Type: other
- SLA: 47
- Contact Name: HENNESEY, DAVID
- Contact Location: BLDG2
- Alternate Name:
- Manufacturer: Microsoft Corporation
- Class: class4
- Cost Center:
- Asset ID: DavePC
- Assigned As: Primary (selected)
- Smart Search?:
- Contract:
- Status: Open
- Owned By: BOB.HELPDESK
- Primary Asgn Group: SOFTWARE
- Assignee Name:
- Severity: 4 - Low
- User Priority: Low
- SDU's unable to fix:
- Total Loss of Service:
- Fix that is: Permanent (selected), Temporary, Either
- Tickets that are: Open (selected), Closed, Either

At the bottom of the form, it says "Selected line is row 1 of 10 records" and "insert probsummary.qbe.g [US]"

Figure 5-3: Filled probsummary search form opened in Database Manager

6 If Record List is enabled:

- To export the entire list, select **List Options > Mass Unload**.
- To export a single record, select the record and select **Options > Export/Unload**.

If Record List is disabled:

- To export the entire list, select **Options > Mass Unload**.
- To export a single record, open it and select **Options > Export/Unload**.

- 7 For this example, select IM1037 and select **Options > Export/Unload**.

The Unload/Export Facility is displayed. This utility provides several options for writing data to an external file, including a formatted text unload that generates a text file that other applications can read.

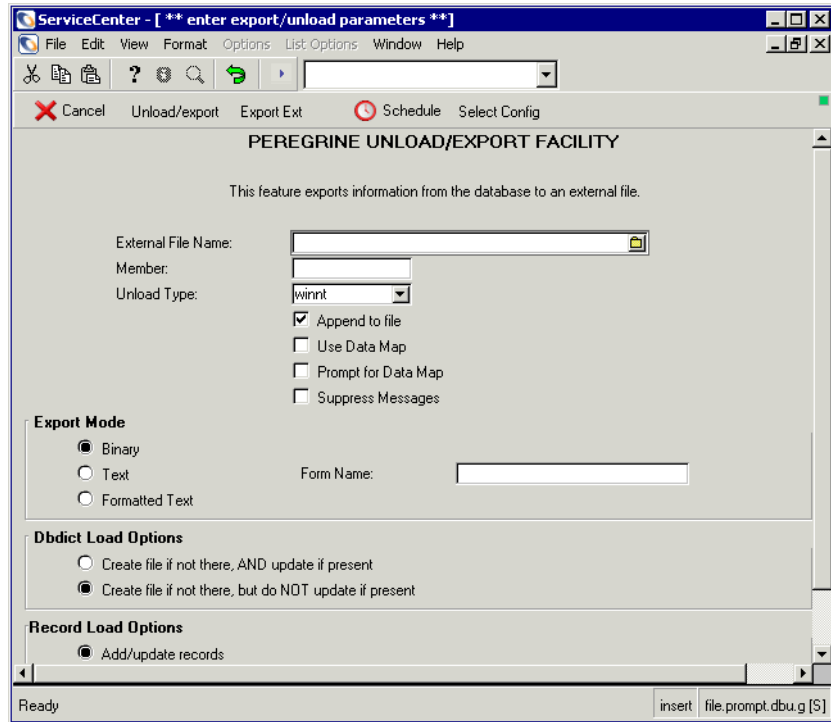


Figure 5-4: The Unload/Export Facility

- 8 Complete all required fields, using the field descriptions in the following table:

- If you are using the export utility with an export descriptor record, complete the first four fields: **External File Name**, **Member** (if any), **Unload Type**, and **Append to file**.
- If you are using the **Formatted Text** option, include the **Form Name**.

For this example, enter `exportsample.txt` in the External File Name text box.

Field	Definition
External File Name	<p>Enter the fully qualified name of the external file to which you want the information written.</p> <p>In the OS/390 environment, you must enter an existing high level qualifier, or the process will wait on execution for action at the system console.</p> <p>In the Unix and Windows NT/2000 environments, include the full path name if you are writing the file to a directory other than the ServiceCenter RUN directory.</p>
Member	If the external file is an OS/390 partitioned dataset, enter the name of the member. Otherwise, leave the field blank.
Unload Type	If the external file is a QSAM dataset, select OS/390. Otherwise, select unix or winnt for a file.
Append to File	<ul style="list-style-type: none"> ■ The default is <i>true</i> (checked), which indicates the exported information is to be appended to any existing file of the same name. If a file of the same name does not exist, one is created. ■ If the field is set to <i>false</i> (unchecked), any existing file of the same name will be replaced with the exported information or, if none exists, a file of that name is created. <p>In an OS/390 environment, remember that you cannot append to a partitioned data set (PDS) member. When a member is defined, the application ALWAYS creates a new data set without respect to the parameter entered.</p>
Use Data Map	Default is <i>false</i> . If this box is checked (<i>true</i>), Datamap executes for every record that is unloaded. Refer to the Data Mapping on page 55 for information on data mapping.
Prompt Data Map	Default is <i>false</i> . If this box is checked (<i>true</i>), and the Use Data Map field is <i>true</i> , the Datamap application displays the standard Datamap confirmation screen for every record that is unloaded.
Suppress Messages	Allows you to ignore all messages produced during the unload process, including the final confirmation message.
Binary	Writes the file in ServiceCenter internal binary formatted mode, which can only be read by the Database Manager's Load facility. This is the preferred mode for unload.

Field	Definition
Text	Writes the file in ServiceCenter internal text mode. Exported text can be read by humans, unlike Binary mode. However, it is a dump of the DBdict, so it can be difficult to read if you are not familiar with the structure of the file. The file is not formatted for data transfer to external applications. It is only readable by another ServiceCenter system, using the Database Manager's Load facility. This is a mode used for unloads and with export options.
Formatted Text	Exports information in a format that can be read by external programs, and structured to match an existing ServiceCenter text (not Windows) format. This is a mode with export options.
Form Name	Name of the format that describes the information to be exported via the Formatted Text Unload/export option. If the format does not exist, you will be prompted to create it.
Dbdict Load Options	When you export a dbdict, you also set the actions for <i>import</i> . The following fields determine those actions. Select one of the following, based on your particular needs. These options are used with an unload. <ul style="list-style-type: none"> ■ Create file if not there, but do NOT update if present ■ Create file if not there, AND update if present
Record Load Options	When you export a record, you also set the actions for <i>import</i> . The following fields determine those actions. Select one of the following, based on your particular needs. These options are used when creating an unload. <ul style="list-style-type: none"> ■ Add/update records ■ Only add new records

9 Launch the export utility.

- To launch the export utility for an unload, click **Unload/export**. For details, see *Unload/Load* on page 93.
- To launch the export utility using an Export Descriptor Record, click **Export Ext**. If an export descriptor record already exists with the name specified in the export parameters, it will appear for verification. If more than one export descriptor record exists for the file you are exporting, ServiceCenter presents a list so you can select the export descriptor record to use.

If no export descriptor record exists for the current file, a message is displayed instructing you to build the export descriptor record. For instructions on building an export descriptor record, see *Creating an Export Descriptor Record* on page 148. For complete instructions on building forms using the Forms Designer Utility, see *ServiceCenter System Tailoring*.

For this example, select **Export Ext**.

- 10 Select an Export Descriptor Record.

For this example, open **probsum.test**.

The File Export External Record Descriptor form is displayed. For a description of the fields on this form, see page 150

Field Name	Field Type	Start Pos	Fld Lng	Array Type	Array Occ	Date Format
number	2	1	9			
open.time	3	10	8			mm/dd/yy
update.time	3	24	8			mm/dd/yy
agreement.id	1	38	20			
close.time	3	58	8			mm/dd/yy

Figure 5-5: The probsum.test Export Descriptor Record for the probsummary file

- 11 Optionally edit the Export Descriptor Record, following the instructions in *Modifying an Export Descriptor Record* on page 152.

12 Click Export.

You'll find the exported text in the ServiceCenter Run directory. The exported text from this example looks like this:

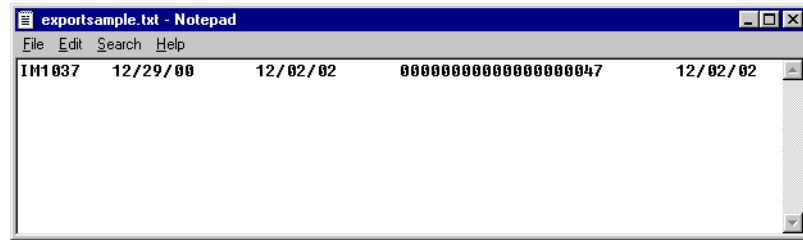


Figure 5-6: Example of exported text

The fields in the example include number, open time, update time, agreement id, and close time, as selected in the export descriptor record shown in Figure 5-5 on page 147.

Creating an Export Descriptor Record

Export descriptor records define the rules for exporting data from ServiceCenter files. Each export descriptor record identifies a source file, the fields from that file to export, and the formatting (start position, width, number of repetitions for array data, and data structure for date/time fields) for each field. Export descriptor records can also identify a related Format Control record, used to validate and format data prior to export.

When you export ServiceCenter data using an export descriptor record, each exported ServiceCenter record produces a single line/record in the resulting external file.

When you create an export descriptor record, the fields on your record are validated against the Database Dictionary for the **ServiceCenter file name** on the export descriptor record. If this validation is unsuccessful, a *Validation failed* message is displayed, along with other messages defining the reason(s) for failed validation. You need to scroll these messages and correct the problems before you can perform the export. ServiceCenter will save your export record regardless of whether the validation is successful or unsuccessful.

Field	Definition
Export Descriptor Record	Enter the name of the export descriptor record. You can enter any value desired, however, it must be unique.
ServiceCenter File Name	Enter the ServiceCenter filename from which the records are to be exported.
Format Control Record Name	Enter the name (if any) of the Format Control record that will be used for the export (refer to <i>Using Format Control with Export</i> on page 162).
Field Names	Enter the field name from this file's Database Dictionary that you want to export. When the field is part of a structure, prefix the field with a structure name (for example, number). If the field is part of an arrayed structure, enter only the field name without the structure.
Field Type	The export validation routine will determine the value for this field from the ServiceCenter Database Dictionary, and apply it automatically.
Start Pos	<p>Enter the starting position for the field in the <i>external</i> file you are exporting. When several lines from an array have been included on the previous line, be sure to take into account the length of each line of the array when determining the start position of the current field, e.g., 5 lines of an array with 80 character spaces for each line, sets aside 400 spaces in a field, making the start of the next field at least 400 spaces later. See <i>Array Occ</i> below. Allow room for date + space + special characters + time. For an example see <i>Export descriptor record for the problem file</i> on page 152.</p> <p>The maximum number of blanks between any two fields exported is 500, but any number below can be used to separate the data from two fields.</p> <p>The validation routine will calculate date and time values.</p>
Fld Lng	Enter the length of the field in the <i>external</i> file you are exporting. If the field is a date/time field, be sure to include the length of both the date portion and the time portion of the field.
Array Type	The export validation routine will determine the value for this field from the ServiceCenter Database Dictionary, and apply it automatically.

Field	Definition
Array Occ	Enter the number of elements in the array to be exported. Be sure to allocate enough space in the external record for arrays by multiplying array occurrence by the field length, e.g., to include 5 lines of an array with 80 character spaces for each line, set aside 400 spaces in this field.
Date Format	<p>Enter the format of the ServiceCenter date/time field. This determines the format of the date and time as they will appear in the external file. The following tokens are used:</p> <ul style="list-style-type: none">■ two digit day dd■ two digit month mm■ two digit year yy■ four digit year yyyy■ hour time hh■ minute time ii■ second time ss <p>Any other character/symbol may be used to separate these variables. They may also be combined in any order or combination, e.g., <i>mm/dd/yyyy hh*ii*ss</i>.</p> <p>If you do not include the time format code the time will not be included in the export file.</p>

- 5 Press **Save** to add the record to the export file and retain the information on your screen. The following message is displayed in the status bar: *export record updated*.

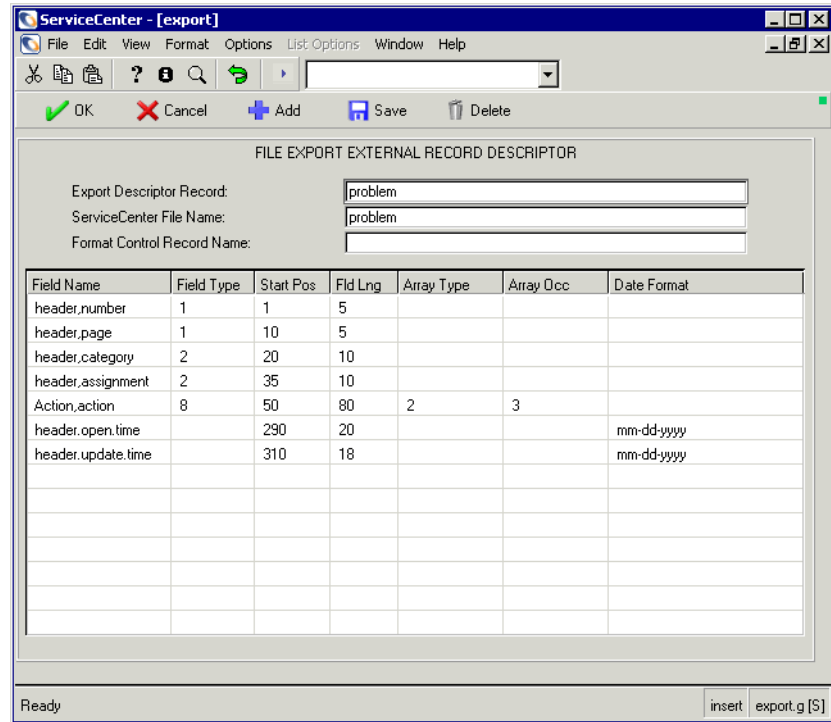


Figure 5-8: Export descriptor record for the problem file



Note: If the message *Validation failed* is displayed, open the messages window by clicking a message icon, and scrolling the recorded messages to identify the message or error source: A blue icon indicates a required action; a black icon indicates information only; a red icon indicates an error message.

Modifying an Export Descriptor Record

You may need to modify an export descriptor record if your validation was unsuccessful or to make any desired changes. When you update a record, ServiceCenter validates the fields on your record against the fields in the Database Dictionary named in the **File name** field of the export descriptor record. If this validation is unsuccessful, a *Validation failed* message will appear along with other messages which will define the reasons for failed

validation. You need to scroll these messages and correct the problems before you can perform the export. ServiceCenter will save your record regardless of whether the validation is successful or unsuccessful the data just may not be retrievable from the file.

To modify an export descriptor record:

- 1 Open Database Manager.
- 2 Enter the name of a form or file in the Database Manager dialog box. For this example, type =export in the **Form** field of the Database Manager dialog box and click **Search** or press Enter.
- 3 To choose an existing record from a QBE list, press Enter.

– or –

To open a specific record, enter the name of the existing record in the **Export Descriptor Record** field and press Enter. For a description of the fields on this form, see page 145.

In either case, if a record exists, ServiceCenter will display the record.

- 4 Using the tab key, move the cursor to the fields you want to change. For a description of the fields on this form, see page 150.
- 5 Enter the new data.
- 6 Press **Save** to update the record to the *export* file.

The message *export record updated* appears at the bottom of your screen.

Exporting Records in the Foreground

When you export records in the foreground (on your client workstation), the time required to export the records will vary depending on how many records are in the category you specify.

The forms below illustrate how to export all Computer Accessories records (category="Computer Accessories") from the *model* file.

To export records in foreground processing:

- 1 Enter the name of a form or file in the Database Manager dialog box. For this example, type `model` in the Form field of the Database Manager.
- 2 Click **Search** or press Enter.
- 3 Select the records you want to export, then click **Search**. For this example:
 - a Select the **model** format.
 - b In the **LI Category** field on the Catalog tab, select **Computer Accessories**.
 This example retrieves all model records where the LI Category value is **Computer Accessories**.

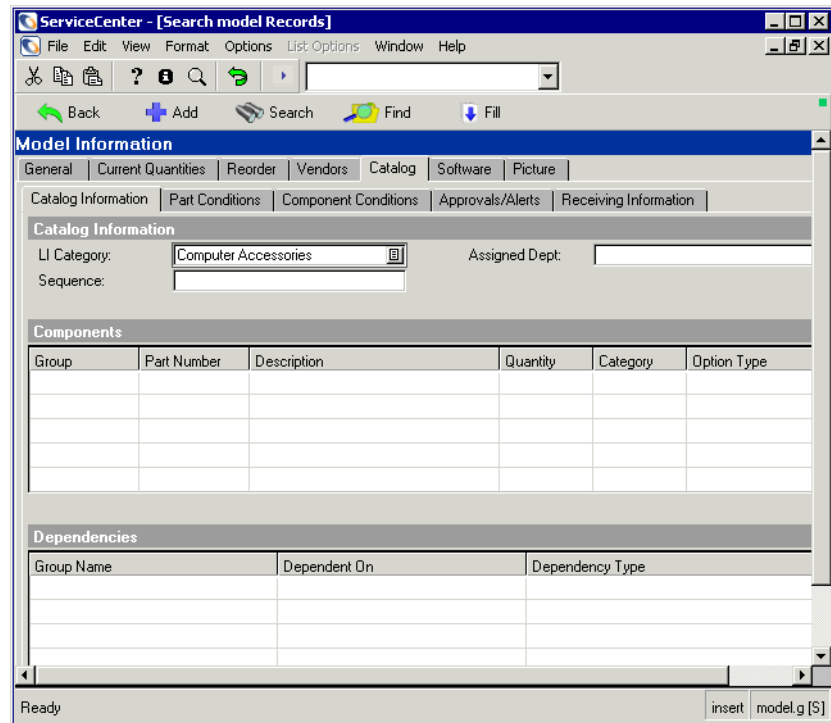


Figure 5-9: Model file—searching for records

- 4 Click **Search** to display model records.

A QBE list of selected records is displayed.

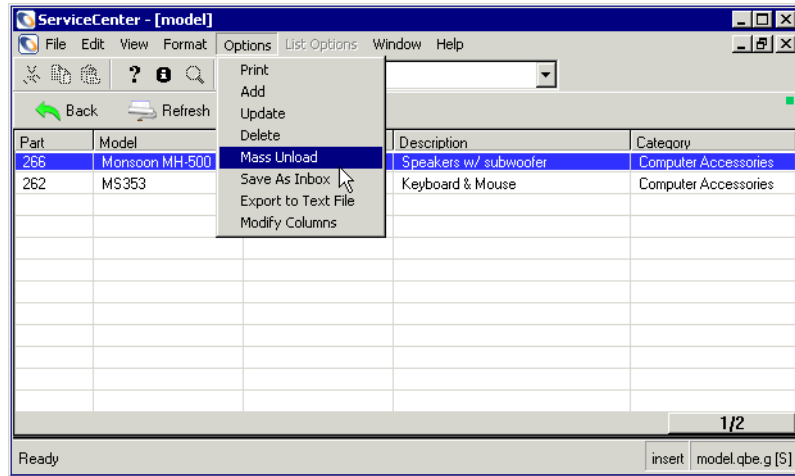


Figure 5-10: QBE list of models

5 Select one of the following:

- If Record List is enabled:
 - To export the entire list, select **List Options > Mass Unload**.
 - To export a single record, select the record and select **Options > Export/Unload**.
- If Record List is disabled:
 - To export the entire list, select **Options > Mass Unload**.
 - To export a single record, open it and select **Options > Export/Unload**.

For this example, select **List Options > Mass Unload** or **Options > Mass Unload**.

- 6 Complete the first four fields on the Export/Unload form. (For Formatted Text Unload, also remember to select Formatted Text for the Export mode, and to enter your export descriptor record name in the Form Name field):

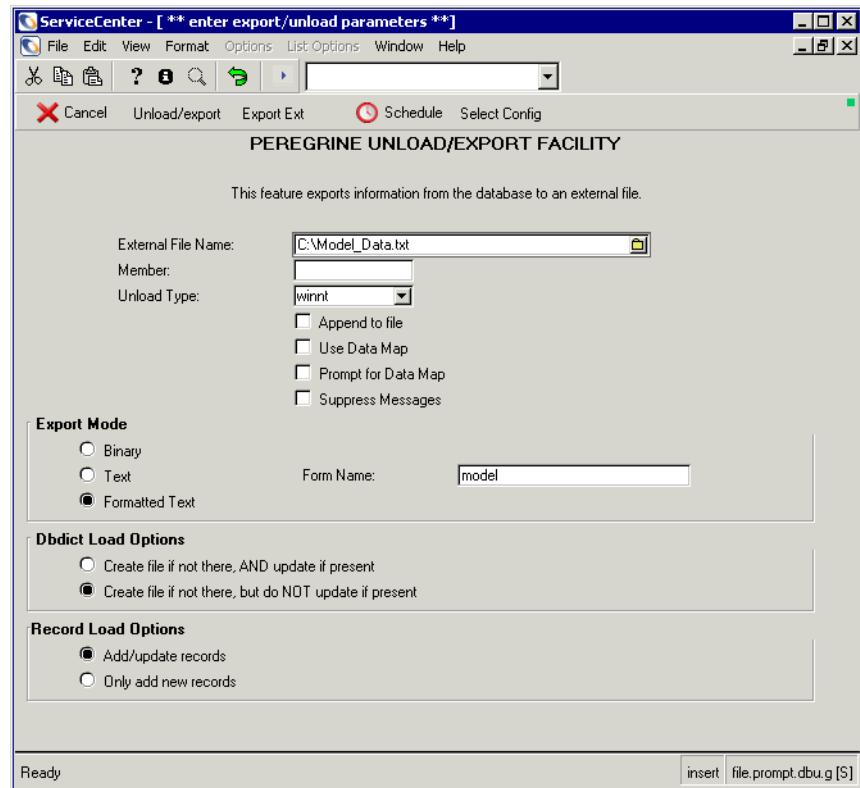


Figure 5-11: Unload/Export Facility using a descriptor record

- 7 Click **Export Ext** to launch the Export Utility (or **Export/unload** for a Formatted Text unload).

The Export External Record Descriptor record for this file is displayed. If more than one Export External Record Descriptor record exists for this file, ServiceCenter presents a list so you can select the record to use.

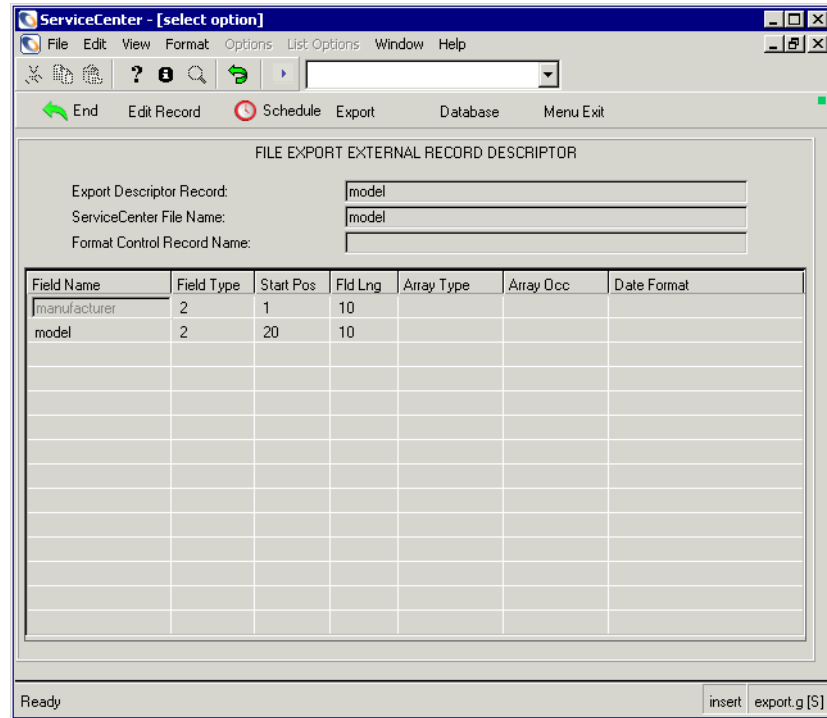


Figure 5-12: Descriptor record for model file

- 8 To make changes to the export descriptor record,
 - a Click **Edit Record** and make the changes.
 - b To save your changes, click **Save**; to exit, click **OK**.

- 9 To begin exporting in the foreground (in your user session), click **Export** on the *export* or *export.g* form.

If the **Schedule** or **Export** buttons do not appear on your screen, the message *Export descriptor has not been validated* will appear, and the validation was not successful for this export record. If this occurs, press **Edit Record** and refer to *Modifying an Export Descriptor Record* on page 152.

Button	Action
End	Exits the export descriptor record and brings you back to the Unload/Export Facility screen.
Edit record	Allows editing of the export descriptor record. For more information, refer to <i>Modifying an Export Descriptor Record</i> on page 152.
Schedule	Displayed if the export record validation was successful. Allows you to schedule the export to run in background. For more information, refer to <i>Exporting Records in the Background</i> on page 159.
Export	Displayed if the export record validation was successful. When this key is pressed, the export process begins in foreground.
Database	Allows you to go directly to Database Manager. When the cursor is placed on the Export Descriptor Record field or File Name field, the system searches for a form matching the contents of the field. If found, you go directly to that form's database. If not found, the following message appears: <i>Format or file not found, please re-enter</i> . Enter the form name of the file you wish to select and continue.
Menu exit	Exits the Export Utility and returns to the originating menu.

When the export is complete, you'll see a QBE list of all records exported, with a message at the bottom of the screen indicating the number of records exported. (You may need to scroll the list to view the records.)



Note: Remember to click the message icon and scroll through all messages when the message in the form footer is preceded with an asterisk (*). The message icons are displayed in the color that indicates the type of message that is available (blue indicates a call to action, black indicates information only, and red indicates an error message).

- 10 To exit, click **End**.

Exporting Records in the Background

When you schedule an export to run as a background operation, you can continue working in ServiceCenter during the export.

To export records in background processing:

- 1 Enter the name of a form or file in the Database Manager dialog box.
- 2 Press Enter or click Search.
- 3 Select the records you want to export, then click Search.

A QBE list of records is displayed.

- 4 Select **Options > Mass Unload**.

If Record List is enabled, the following menu options are available:

- **Options > Export/Unload**
- **List Options > Mass Unload**

Taking the Options - Export/Unload option will unload only one record. If you want to unload the entire file, you have to use List Options - Mass Unload.

If Record List is disabled, select **Options > Mass Unload** to unload the entire list of records

- 5 Select an Export Mode and enter the name of a form in the Form Name field.
- 6 Click Export Ext.

The descriptor record for this file is displayed. If more than one descriptor record exists for this file, ServiceCenter presents a list so you can select the record to use.

- 7 Click **Schedule** to schedule the export for background processing at a specific time.

If the **Schedule** or **Export** buttons do not display on your screen, the message *Export record has not been validated* is displayed because the export record validation was not successful. In this case, click **Edit record and correct the export descriptor record**. See [Modifying an Export Descriptor Record](#) on page 152 for more information.

The Schedule File to be Exported form is displayed.

8 Enter the required data in each field.

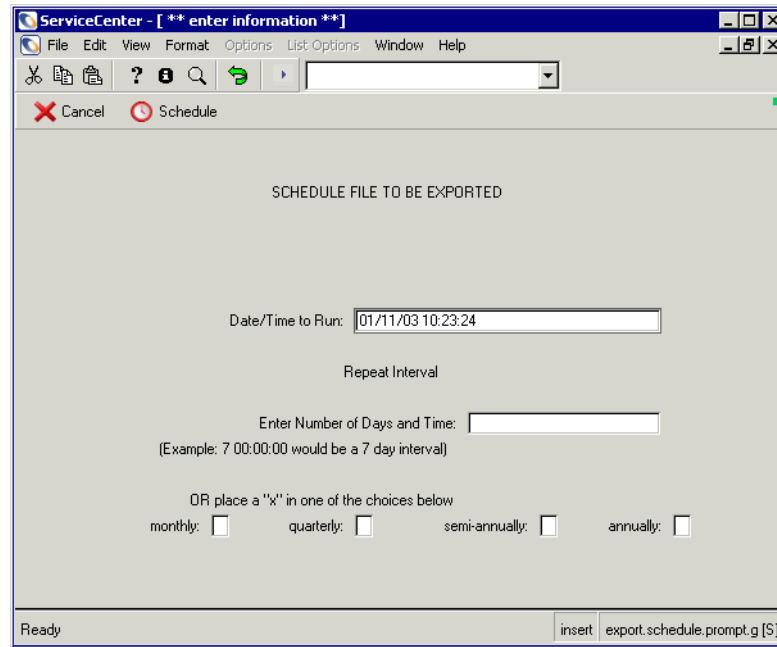


Figure 5-13: Schedule record

The fields for this format are defined below.

Field	Definition
Date/Time To Run	This field defaults to the current date and time. If you do not want the export to process immediately, enter the desired date and time. The format is your current date/time format, for example, MM/DD/YY HH:MM:SS.
Repeat Interval	(Optional) Use the repeat interval if you want the process to repeat on a regular basis. If the repeat interval is less than one day, use HH:MM:SS

Field	Definition
Enter Number Of Days And Time	If you want this export to be rescheduled on a repeat interval, enter the interval in this field. The format is DDD HH:MM:SS.
OR place An "x" In one of the choices below	<p>Enter an x in the desired interval. You can only select one repeat interval option. If you entered a value for the Number of Days and Time, leave these fields blank.</p> <ul style="list-style-type: none"> ■ Monthly—The export occurs every month on the day specified in the Date/Time to Run field. For example, if the Date/Time to Run field contains <i>04/01/98</i>, then the export will occur on the first day of every month. ■ Quarterly—The export occurs four times per year, on the day specified in the Date/Time to Run field. ■ Semi-annually—The export occurs twice a year, on the day specified in the Date/Time to Run field. ■ Annually—The export occurs each year on the day specified in the Date/Time to Run field.

- 9 After you have filled out the form to process the export at the desired time, click **Schedule**. If a repeat interval is selected, a new schedule record will automatically be created for the next interval.

The message in the status bar at the bottom of your screen indicates the time the export is scheduled to run.

Messages for scheduled exports are retained in the `msglog` file.

- 10 Press **Exit** to exit the **Database Manager**.

Printing an Export Descriptor Record

You may want to print your export descriptor record when you encounter a *Validation failed* error message during the export process. You can then check the fields on your export descriptor record against the fields in the Database Dictionary.

To print an export descriptor record:

- 1 Enter `=export` in the **Form** field of the Database Manager dialog box.
- 2 Click **Search** or press **Enter**.

The export record descriptor form is displayed.

- 3 Press Enter to display a QBE of existing records.

~ or ~

Enter the name of an existing record and press Enter.

- 4 You can print the record using either of two methods:

- Select **File > Print Screen**.
- Select **Options > print rec**.

Note: The **print rec** option sends the entire export record to the printer specified in your *operator* record. The export.g form is refreshed, and the following message is displayed in the status bar: *Report: export.print scheduled to run at <mm/dd/yyyy 00:00:00>*.

Using Format Control with Export

Due to the complexity of Format Control, review the *Format Control Manual* thoroughly before proceeding.

Some issues to remember when using Format Control with export are:

- Enter the name of the Format Control record in the *Format Control Record Name* field of the export descriptor record.
- Never assign the Format Control record to the filename unless you want to force its use with **all** formats associated to the file.
- Always enter a value of *true* for display when using validation and calculation routines.

To add Format Control to the *model* export descriptor record created in the previous exercises:

- 1 Retrieve the model export descriptor record created in the previous exercises by using the Database Manager to access the export file via the export form. Then, enter the name of an existing record and press Enter, or search and select the record.
- 2 Move the cursor to the **Format Control Record Name** field and enter the name of the Format Control record to associate with the export descriptor record, *model.fc* in this example.

Note: This Format Control file may not exist on your systems, and an error message may appear in the Message window.

Building the Format Control Record

The screens below illustrate how to add a new Format Control record called *model.fc* to the database using three Format Control options: initialization, calculations, and validations. Below each screen is a brief description of each option.

The following steps progress from the *formatctrl.maint.initial.g* form displayed by the actions in the previous section.

To build the Format Control record:

- 1 Retrieve an existing Format Control record or create a new record:
 - a From the *formatctrl.maint.initial.g* form, press Enter to display a QBE list of all Format Control records.
 - b Select a record by double-clicking the cursor on a record name.
 - c Modify the record to use the values shown below.
 - or –
 - Create a new Format Control record by clicking New to add the record to the database.

Note: More Format Control options appear after you add the Format Control record.
- 2 Set up any Format Control Initialization Expressions:
 - a Enter the following information into the each field of the *formatctrl.maint.initial.g* form. For additional information, see the *Format Control* section in *ServiceCenter System Tailoring, Volume 1*.

Field	Value
Name	model.fc
Initialization Expressions	\$a=tod() Use this expression to assign today's date to a variable (\$a) at the beginning of the job. You can then assign this variable to a field in each record that will be exported to the external file.

ServiceCenter - [Format Control: model.icm]

File Edit View Format Options List Options Window Help

F2 - OK F3 - Cancel F11 - Previous F10 - Next F1 - Add F4 - Save F5 - Delete

Views Queries Calculations Validations Subroutines Add Options Privileges

Format Control Maintenance - Main Information

Name: View: short

File Name:

System:

Query Format:

Default QBE Fmt:

Save Copy Stored Form Name Run Script Use Default Sort

Default Sort sequence for queries

Initialization Expressions

Ready Response 0.100 draw 0.50 insert formatctrl.maint.g(fc.view) [S]

Figure 5-15: Format Control record for the model export descriptor record

- b** Click Save, or click OK to save when prompted.

- 3 Set up any Format Control Calculations:
 - a Choose **Calculations** to create calculation expressions.
 - b Enter the values in the following table in the Calculations form. Refer to *Format Control* in *System Tailoring, Volume 1* for additional information.

Field	Values
Dis [Display]	true
Calculations	<p>warranty.start in \$file=\$a:</p> <p>Use this option with an initialization expression to assign a value to a field to be exported to the external file. In this example, we assign the value \$a to the ServiceCenter field <i>warranty.start</i>; since \$a was defined as <i>tod()</i> in the initialization expression, <i>tod()</i> will be forced into each external record associated with the field <i>warranty.start</i>, regardless of what values exist for this field in the ServiceCenter file.</p>

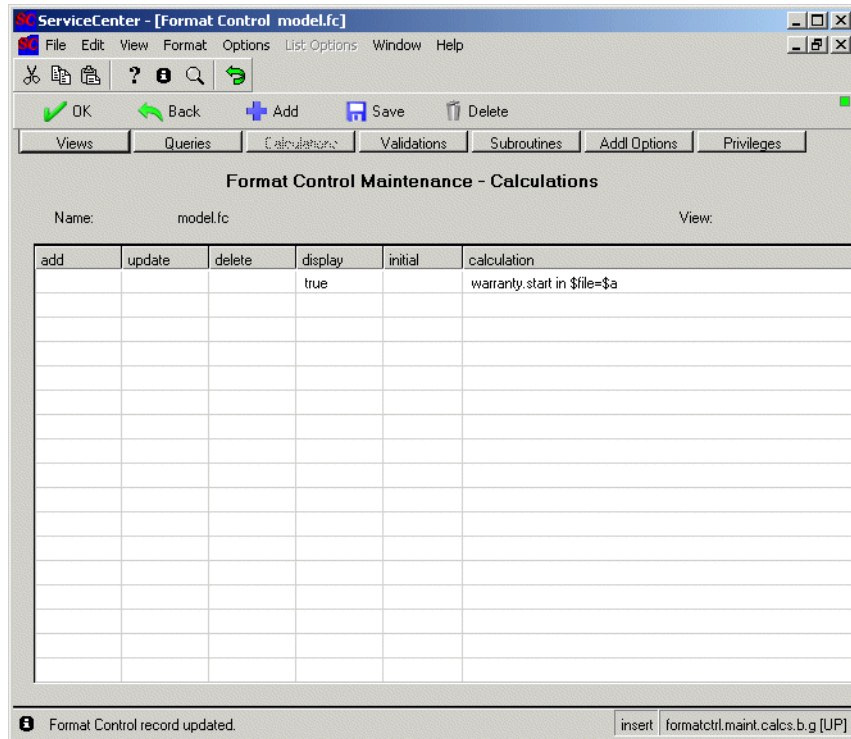


Figure 5-16: Format Control Calculations for the model export descriptor record

Note: The `warranty.start` field must exist in the model Database Dictionary record. Typically (as in this example), but not necessarily, it also exists on the export descriptor record.

- 4 Click **Save** or click **OK** to save when prompted.
- 5 Set up any Format Control Validations:
 - a Choose **Validations** to assign a validation expression.
 - b Enter a validation(s) expression as described in the following table:

Field	Values
Dis [Display]	true
Validations	<p><code>part.no in \$file>"10000"</code></p> <p>Use this option to further restrict the records that are to be exported from the ServiceCenter file. This example exports only records with part numbers above 10,000.</p> <p><code>part.no</code> is a character field, so we use double quotes around the <code>part.no</code> value.</p> <p>Set the Display column to true, and add an appropriate message such as: <i>We are only exporting part numbers above 10,000.</i></p>
Message	<p><i>We are only exporting part numbers above 10,000</i></p> <p>This message will display for each record that does not qualify after the export completes. If the export runs in foreground, the messages will appear in the Message window. If the export runs in background, messages will be added to the <i>msglog</i> file.</p>

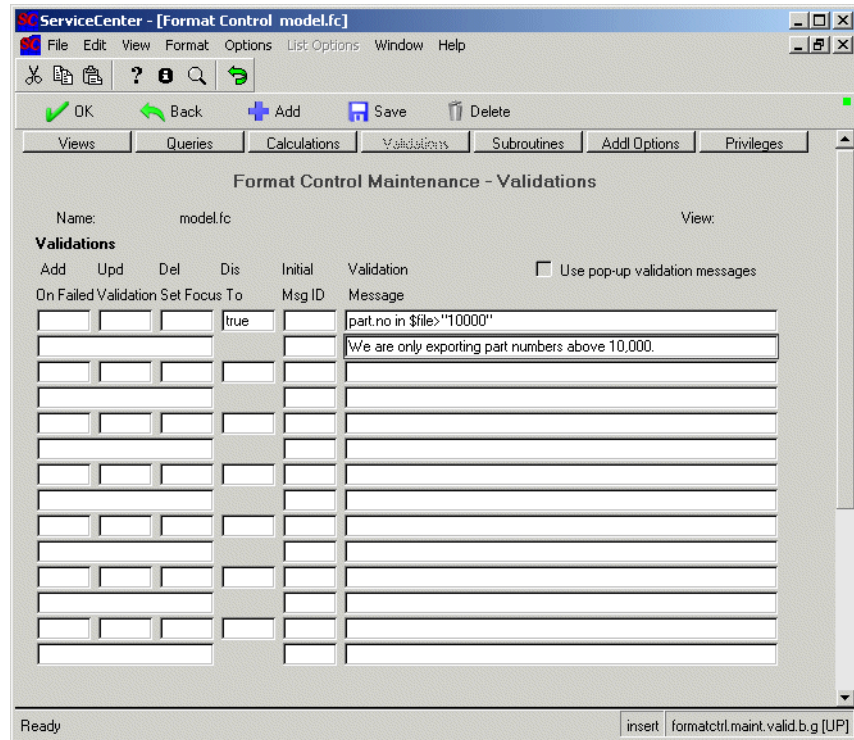


Figure 5-17: Format Control Validations for the model export descriptor record

Note: The `part.no` field must exist in the model Database Dictionary record. Typically, but not necessarily, it also exists on the export descriptor record.

- 6 Click **Save** or **OK** to update the Format Control record.
- 7 Click **Back** to exit the Format Control record.

Exporting to Excel in a Different Language

If you would like to export a file to Excel in a language other than what your system is using, you can do this by making changes to the display options in the Database Manager.

Note: This option is available only on Windows clients.

To change the display options and export a file to Excel in a different language:

- 1 Open Database Manager from the Utilities tab.
- 2 Enter **displayoption** in the Database Manager **Form** field and click Search. A search form is displayed.
- 3 Fill in the **Screen Id** field (for example, enter **apm.edit.problem**) and enter **Export to Excel** in the **Default Label** field to restrict your search.
- 4 Enter **prompt** in the **Name field**.
- 5 Enter the language abbreviation in the **Values** field for the language you want to use to export to Excel.

For this example, enter “ES “for Spanish (be sure to include the quotation marks).

This will set your display options so that when you are ready to export this file to Excel, it will export the file in Spanish.

For information on language abbreviations, see the *ServiceCenter Technical Reference*.

- 6 Click **Save** to save your changes.

To learn more about using Database Manager, see the *Database Management and Administration Guide*.

6 Import Utility

CHAPTER

This chapter introduces the Import Utility and how to import specified fields from an external file into a ServiceCenter file. ServiceCenter comes with the Text Import Wizard tool, which provides a guided front-end for importing from fixed-width or comma-delimited text files. Users can also choose to begin the import in the standard method, but the Import Wizard will complete the process. This tool provides a direct translation from an external file into an existing ServiceCenter database file.

This chapter is organized into the following main sections.

- *Overview of the Import Function* on page 172.
- *Import Text Using the Import Wizard* on page 172.
- *Import records in Foreground* on page 188.
- *Import records in Background* on page 189.
- *Print an Import Descriptor record* on page 191.
- *Using Format Control with Import* on page 192.

Overview of the Import Function

The following are a few points to keep in mind regarding the import function:

- The process relies upon an import descriptor record, which is validated against the ServiceCenter Database Dictionary before it is added or updated. The descriptor record can be created at import time, or prepared ahead of time.
- You can run import in the foreground or schedule it to run in the background.
- The import function uses the mass load function and imports ServiceCenter data in *raw* format.
- The external file must be flat sequential and fixed length or flat sequential and character-delimited. Members of partitioned data sets can also be used (OS/390).
- You can use Format Control with the import function. The most common options are *initialization*, *calculation*, and *validation expressions*.
- With ServiceCenter, all import operations are performed through the Text Import Wizard. Some previous releases used a manual import process.
- You cannot import data into a structured array in ServiceCenter.

Import Text Using the Import Wizard

Text files imported into ServiceCenter include data files exported from ServiceCenter in text format or data files from other systems. These data files will then be incorporated into ServiceCenter or will replace current ServiceCenter data. An example of the latter would be a text file listing your company's employees, which needs to replace the current contacts list in ServiceCenter.

Use the following procedure to import a text file with the Text Import Wizard. This example uses a text file, listing contact information. For different text files, some steps may differ slightly.

Accessing the Wizard

Use the following steps to access the Text Import Wizard

- 1 Review the text file. Identify the order in which the fields appear, file names, and which fields contain data that should be imported into the ServiceCenter file (table).

This file will be imported into ServiceCenter and update the current contacts file.

- 2 Place the text file in a location accessible to your ServiceCenter client.
 - 3 Log onto ServiceCenter as a System Administrator. In the default system, this user is named FALCON. No password is required.
 - 4 Select the Toolkit tab.
 - 5 Click Database Manager.
- The Database Manager prompt screen will appear.

- 6 Select Text Import Wizard from the Options menu.

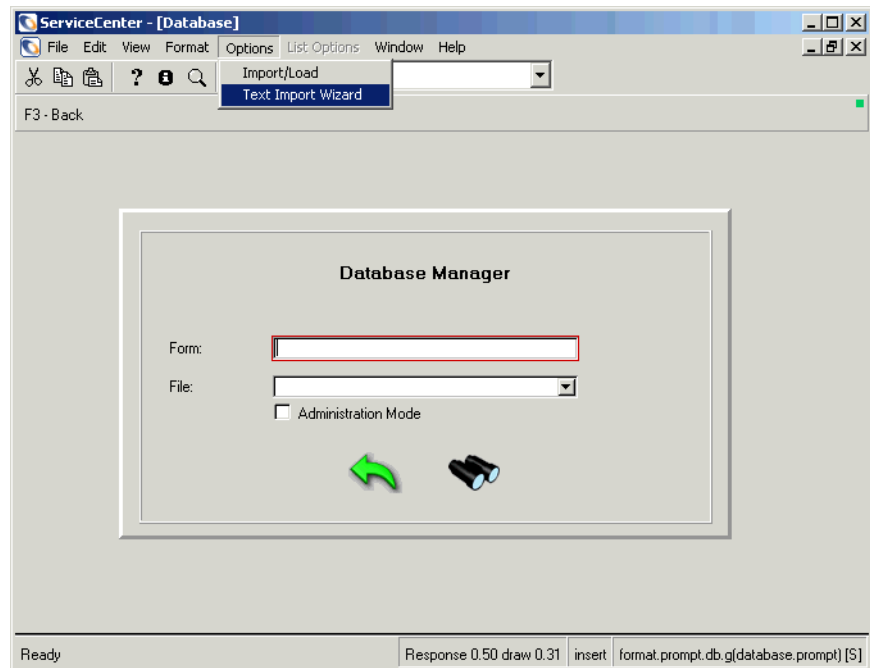


Figure 6-1: Database Manager prompt

The Import Wizard form will appear (Figure 6-2 on page 174).

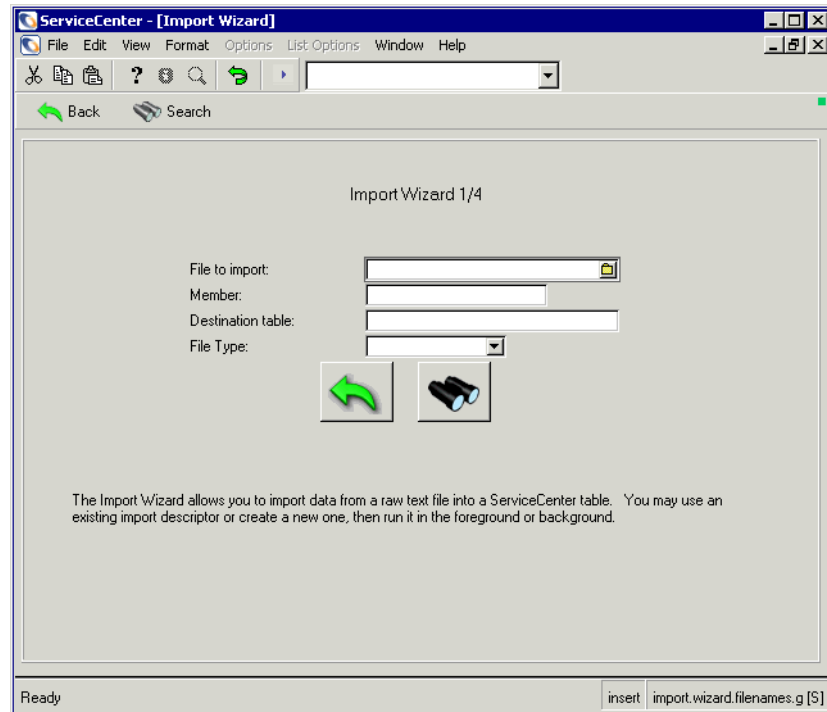


Figure 6-2: Import Wizard

Importing a File with the Wizard

Use the following procedure to import a file with the Import Wizard

- 1 In the **File to Import** field, enter the path and name of the file containing the contact names that you want to import.
- 2 In the **Destination Table** field, enter the name of the table in ServiceCenter into which you want to import your data.
- 3 If the external file is an OS/390 partitioned dataset, enter the name of the member in the **Member** field. Otherwise, leave this field blank.
- 4 In the **File Type** field, select the operating system you are running on: Windows, OS/2, Mac, and so on. This field is optional.
- 5 Click **Search** or press F6.

If more than one import descriptor record exists for that file, a QBE list of records is displayed.

- 6 Select a record from the list, or press **New** to create a new record.

The **Creating a New Import Descriptor** form (Figure 6-3 on page 175) is displayed if no import descriptor record currently exists. Otherwise, the descriptor record is displayed.

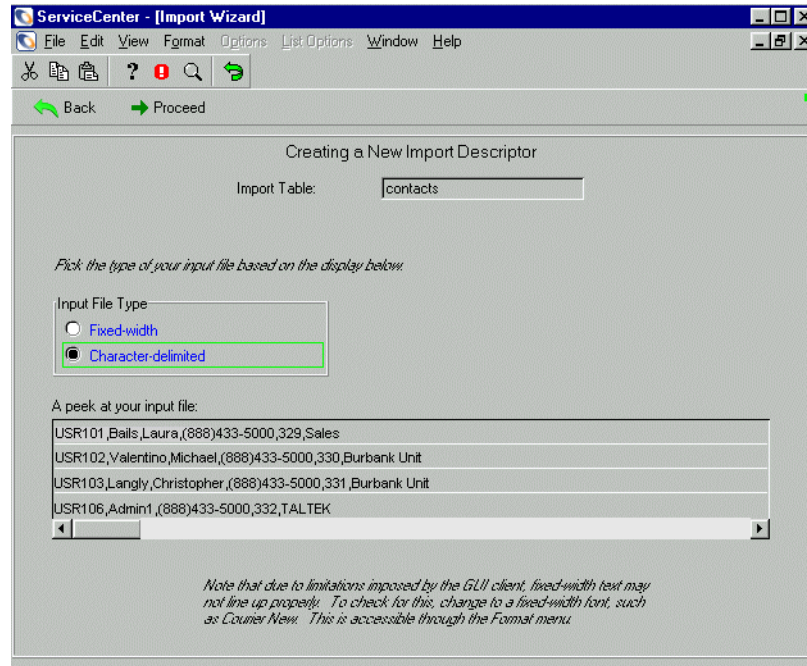


Figure 6-3: New Descriptor Form

- 7 Select an **Input File Type** by clicking on the appropriate option button:
- **Fixed-width** - indicates the text in the file is separated by a specified width. For example:

```

1  2  3  4  5  6  7  8  9  10 11 12 13 14
a  l  p  h  a      b  e  t  a  e  c  h  o
t  e  s  t      t  h  i  s  h  e  r  e
  
```

- **Character-delimited** - indicates the text in this file is separated by a character, such as a comma. For example:
alpha, beta, echo, bravo

The first few lines of the text file are displayed in the table below the option buttons.

Note: The Input File Type value selected determines the appearance of the next screen. If selecting **Character-delimited**, delimiter options and text qualifier fields are displayed.

8 Click **Proceed** or press F5.

The third Import Wizard form will appear (Figure 6-4 on page 176).

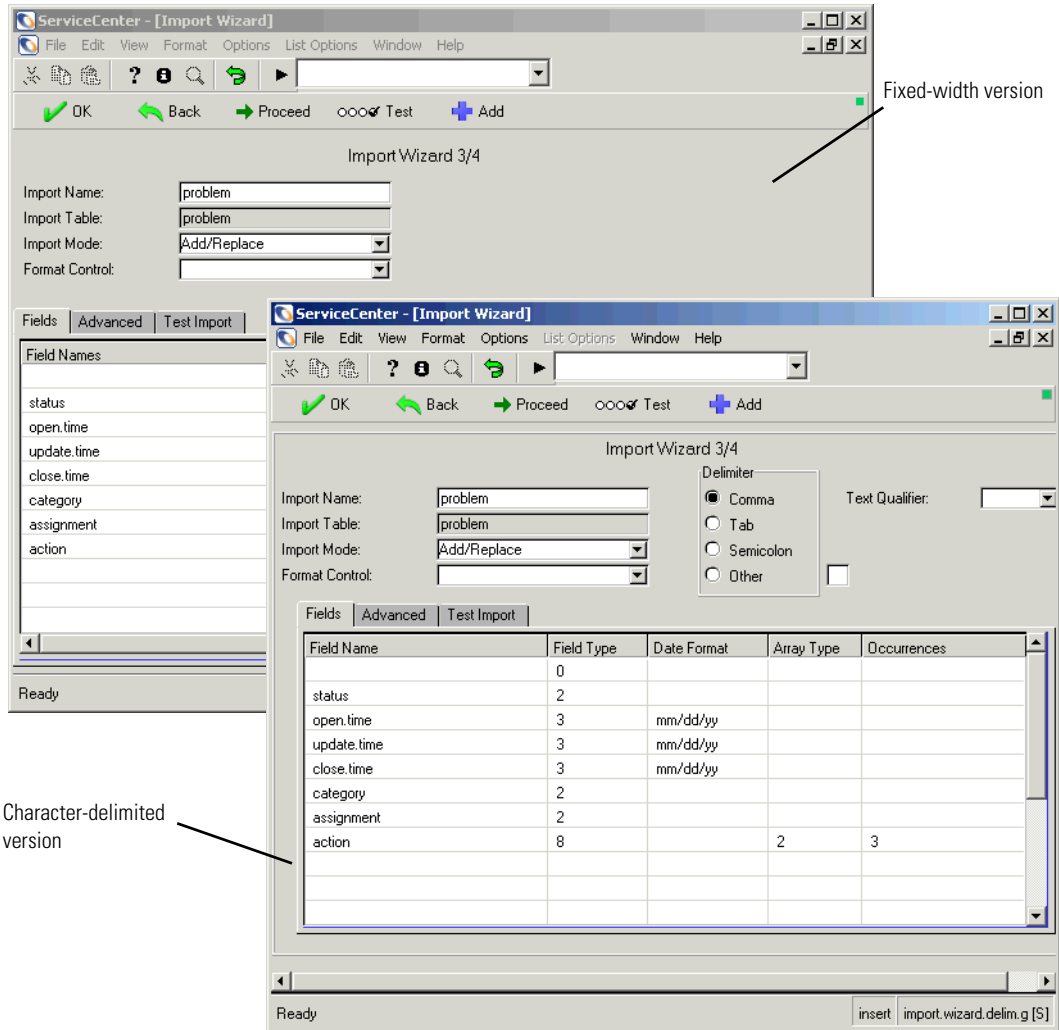


Figure 6-4: Import Wizard Data Definition Forms

Note: In the format control dropdown list you'll see format control records with file names equal to the one to import into. If you do not see the format control record you want to use, be sure to type the correct file name for the format control record.

- 9 Enter an import descriptor title in the **Import Name** field. This title can be used later, if you want to reuse this import description. For example, you are going to import this text file regularly and you do not want run the Import Wizard and describe the file each time.
- 10 Select an **Import Mode** from the drop-down list
Use **Add/Replace** to create a new contact record if one does not exist, or replace the existing record with a new record. To update existing records in the target table, use **Add/Replace** in conjunction with the update field names. For further information about UPDATE field names, see *Modifying an Import Descriptor record* on page 183.
If you selected the Character-delimited file type on the previous screen, you'll need to provide the **Delimiter** selection and **Text Qualifier** input values.
- 11 In the **Format Control** field, enter the name (if any) of the Format Control record that will be used for the import (refer to *Using Format Control with Import* on page 192).
- 12 Select the **Delimiter** character in your text file.
Most common text dump formats use either tabs or commas as separators.
- 13 Enter a qualifying character in the **Text Qualifier** field if one is used in the import text file; otherwise, leave the field blank.

This field allows the user to specify what (if any) qualifying marks are found around the delimited data to be imported. For example, the quotes in the following string are the text qualifiers, "abc","123","ABC". These marks are used when parsing the imported data file onto the system.

Field Name	Field Type	Date Format	Array Type	Occurrences
	0			
status	2			
open.time	3	mm/dd/yy		
update.time	3	mm/dd/yy		
close.time	3	mm/dd/yy		
category	2			
assignment	2			
action	8		2	3

Figure 6-5: Import Wizard Data Definition Form (Character-delimited version)

14 Enter the table configuration information in the Fields tab.

You enter the configuration information in the arrays in the Fields tab: **Field Name**, **Date Format**, and **Occurrences**. **Field Type** and **Array Type** are entered automatically. These arrays allow you to relate the columns from the import file (such as, the `contacts.txt` file) to the columns in your destination table (or file). In this case the `contacts` table (or file). Fields are read in the import file from left to right and placed into the table's columns as you list them in the arrays.

When you place the cursor in an editable array, a drop-down list of valid entries is displayed. Use the down arrow button to select the entry for that array.

Field Name	Field Type	Date Format	Array Type	Occurrences
contact.name	2			
contact.name	2			
contact.name.attach	2			
user.id	2			
group	2			
shift	2			
dept	2			
last.name	2			

Figure 6-6: Fields Arrays

- For example, if you look at the sample file shown in Figure 6-3 on page 175, the name used to identify the contact is listed in the first column of the import table, i.e., **USR101**, **USR102**, etc. In the first **Field Name** array, enter **contact.name**. This name will be placed in the first column of the contacts table in ServiceCenter.
- Next in the text sample is the contact's last name. Place the cursor in the second array and select **last.name**.
- Next in the text sample is the contact's first name. Place the cursor in the third array and select **first.name**.
- Next in the text sample is the contact's telephone number. Place the cursor in the fourth array and select **contact.number**.
- Next in the text sample is the contact's extension number. Place the cursor in the fifth array and select **extension**.
- Finally in the text sample is the contact's department. Place the cursor in the sixth array and select **dept**.

Note: The **Field Type** column is filled in automatically and cannot be modified.

- 15 If a column in your import file contains a date/time value, such as, *12/25/2000*, specify the format of the data *in your text file* in the **Date Format** column. For example, *12/25/2000* is a date format of *mm/dd/yyyy*.

Note: The **Array Type** and **Occurrences** in the example in Figure 6-5 on page 178 are not used, as it is assumed you are not importing an array.

- 16 If your ServiceCenter display font is not a fixed font, you may optionally select **Font** from the **Format** menu and switch to *Courier* font. Doing so makes test output easier to read.

In the next step, you will test the import descriptor. A fixed space font is necessary to properly view the results, which are displayed in the **Test Import** tab.

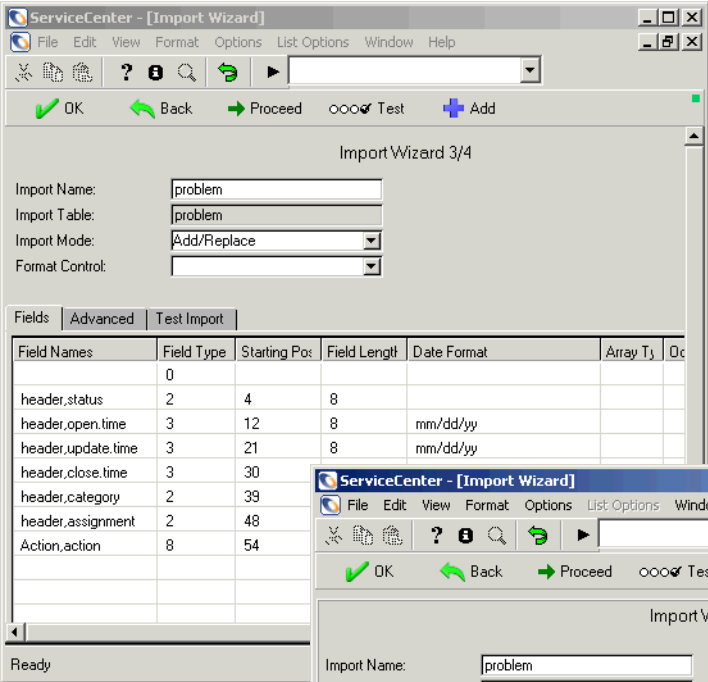
Note: When you create an import descriptor record, the fields on your record are validated against the fields in the dbdict named in the *File name* input field on the import descriptor record. If this validation is unsuccessful, a *Validation failed* message will appear along with several other messages, which define the cause of the failed validation. You need to scroll through these messages and correct the problems before you can perform the import. ServiceCenter will save your import record regardless of whether the validation is successful or unsuccessful.

Verifying the Import

To verify the import:

- 1 Click **Test** or press **F6** to test your import descriptor.

The file will appear in the Test Import tab.



Fixed-width version

Character-delimited version

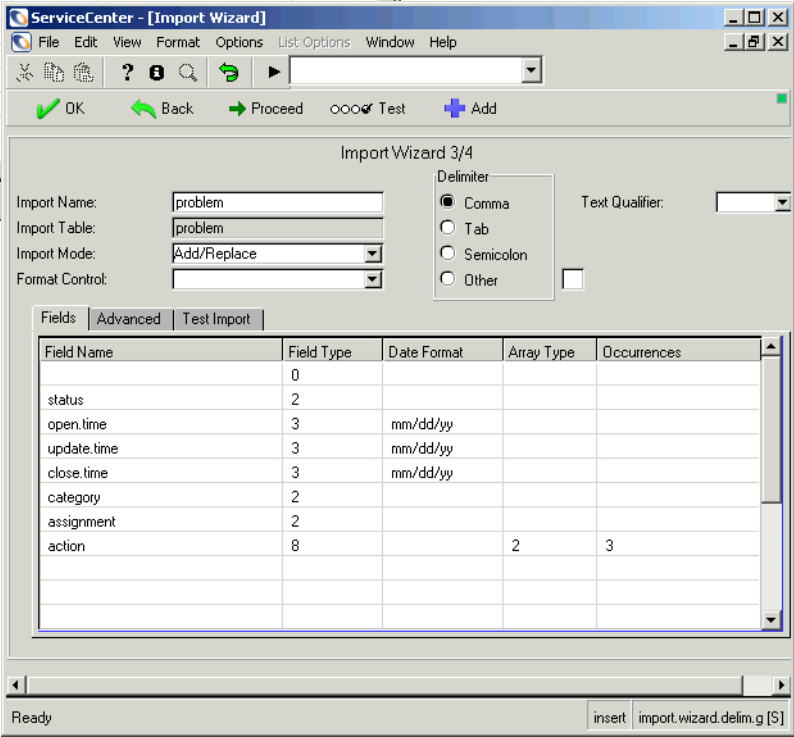


Figure 6-7: Import Wizard Test form

- 2 If the Test Import tab is not displayed, select that tab to view your import.
The first row acts as a header row to indicate into which of the columns your data will be imported. The subsequent rows contain actual data from your import. For example, in Figure 6-7 on page 181, the first record to be imported saves *USR101* as a **contact.name**.
- 3 Click **Add** or press **F1** to save your import descriptor.
Note: You do not have to save the import descriptor. However, if the results are satisfactory, you can save your import descriptor for later use. If you do not save the descriptor, you will have to build another descriptor the next time you want to import a similar text file.
- 4 Click **Proceed** or press **F5** to begin the actual import process.
- 5 When you want to run the import, do one of the following:
 - Click **Run** or press **F5** to run the import immediately. The system begins importing the file. When it completes, you will receive a message to that effect.
 - To run the import at regular intervals, enter the **Scheduling Options** information:
 - Enter the date and time when you want the first import to take place into the **Expiration** field. The format must be **mm/dd/yy hh:mm:ss**.
Note: Scheduled imports always run on the server, so they can *only* be used to load files accessible to the server. A scheduled import of a file on your local hard drive will not work because the background process will try to find that file on the ServiceCenter application server using the path you provide.
 - Click **Schedule** or press **F6**.

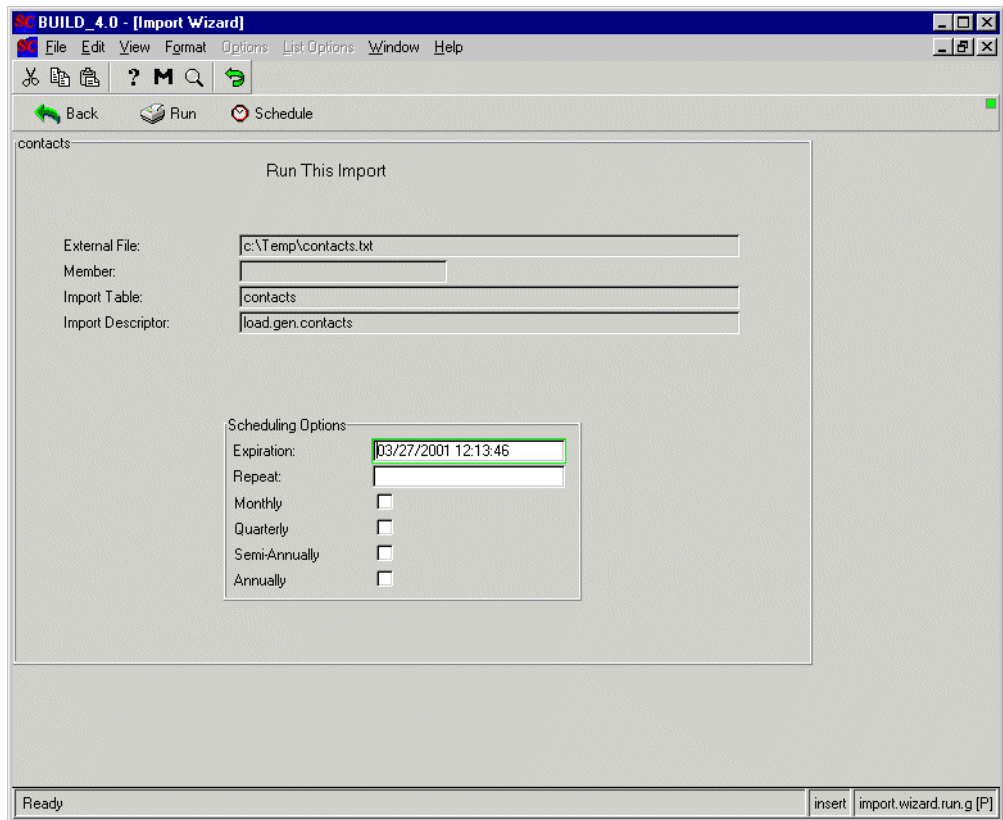


Figure 6-8: Import run form

Modifying an Import Descriptor record

To modify an Import Descriptor record:

- 1 Select the Toolkit tab.
- 2 Click Database Manager.
The Database Manager prompt screen will appear.
- 3 Select Text Import Wizard from the Options menu.
- 4 In the File to Import field, enter the path and name of the file containing the contact names that you want to import.
- 5 In the Destination Table field, enter the name of the table in ServiceCenter into which you want to load your data.

- 6 If the external file is an OS/390 partitioned dataset, enter the name of the member in the **Member** field. Otherwise, leave this field blank.
- 7 In the **File Type** field, select the operating system you are running on: Windows, OS/2, Mac, and so on. This field is optional.
- 8 Click **Search** or press **F6**.

If more than one import descriptor record exists for that file, a QBE list of records is displayed.

The screenshot shows the 'ServiceCenter - [import]' window with the 'Import Wizard 2/4' dialog open. The dialog has a menu bar (File, Edit, View, Format, Options, List Options, Window, Help) and a toolbar with icons for OK, Cancel, Previous, Next, Add, Save, Delete, Views, Find, and Fill. Below the toolbar is a title bar 'Import Wizard 2/4' and a subtitle 'Select an import descriptor from the list below, or create a new one'. A table lists several import descriptors, with 'RB company' selected. Below the table is the 'IMPORT DESCRIPTOR RECORD' section, which includes fields for 'Import Name', 'File Name', 'Format Ctrl', 'UPDATE', 'Mode(add/replace)', 'Unique Key', and 'Field Names'. A table at the bottom shows the field definitions for 'customer.id' and 'company'. The status bar at the bottom indicates 'Selected line is row 1 of 32 records retrieved' and 'insert import.qbe.g [S]'.

Import Name	Delim/Fixed	Mode	Format Control	Query	Expressions	Update Field Name
RB company	delim	replace				
clientIP	delim	replace			type in \$file="pc"	
contacts.1	delim	replace				
contacts.dodds	delim	replace				
contacts.erp.hr	fixed	replace	contacts.erp			

IMPORT DESCRIPTOR RECORD

Import Name: Mode(add/replace): Translate:

File Name: Unique Key:

Format Ctrl:

UPDATE: Field Names:

If field names are specified to select records for update, the fields must be defined in the descriptor below.

Field Name	Field Ty	Start F	Flid Lni	Array Type	Array Occ	Date Fmt	Time Inc	Time Lng
customer.id	2							
company	2	9	50					

Selected line is row 1 of 32 records retrieved insert import.qbe.g [S]

Figure 6-9: Import descriptor record for edit

- 9 Move the cursor to the field(s) you want to change and enter the new data. (For field descriptions, see [Fields](#) on page 185.)
- 10 Press **Save** to update the record in the import file.

The message *Record updated in the import file* will appear at the bottom of your screen.

Fields

The following table provides information for how to modify your import descriptor record.

Field	Description
Import Name	Enter the name of the import descriptor record.
File Name	Enter the ServiceCenter filename to which the records are to be imported. (Import now supports delimited text files.)
Mode (Add/Replace)	Use the ADD option if you want to add records that do not already exist in the ServiceCenter database. Use the REPLACE option if you want to replace records that do exist in the ServiceCenter database. Use ADD/REPLACE to add new records and replace existing ones. To update existing records, use ADD/REPLACE in conjunction with UPDATE Field Names. Note: Replace does not work on a file without a <i>unique</i> or <i>no duplicates</i> key.
Format Ctrl	Enter the name (if any) of the Format Control record that will be used for the import (refer to <i>Using Format Control with Import</i> on page 192).
Text Qualifier	This field allows the user to specify what (if any) qualifying marks are found around the delimited data to be imported. For example, the quotes in the following string are the text qualifiers, "abc","123","ABC". These marks are used when parsing the imported data file onto the system.
Delimiter	Selection of field separators. Select one to match the format of the import text file.
Field Names	Enter the field name from the dbdict that you want to import. When the field is part of a structure, prefix the field with a structure name (for example, number).
Field Type	The import validation routine will determine the value for this field from the ServiceCenter dbdict.
Start Pos	Enter the starting position for the field in the external file that you are importing.
Fld Lng	Enter the length of the field in the external file that you are importing. If the field is a date/time field, enter only the date portion of the field.

Field	Description
Array Type	The import validation routine will determine the value for this field from the ServiceCenter dbdict. Be sure to allocate enough space in the external record for arrays by multiplying array occurrence by field length.
Array Occ	Enter the number of elements in the array to be imported.
Date Format	<p>Enter the format of the ServiceCenter date/time field. The following options are available (Import supports arbitrarily structured date formats for incoming dates. Any combination of the tokens yyyy, yy, mm, dd, hh, ii, ss can be constructed for incoming dates):</p> <p>mm/dd/yy mm-dd-yy mm.dd.yy mmdyy dd/mm/yy dd-mm-yy dd.mm.yy ddmmyy yy/dd/mm yy-dd-mm yy.dd.mm yyddmm yy/mm/dd yy-mm-dd yy.mm.dd yymmdd yy.ddd yyddd</p> <p>Note: The date formats defined above describe the format of the date as it is displayed in the external file.</p>
Case Conversion	Use upper case to translate characters to upper case; lower case to translate characters to lower case; (none) to leave characters as-is.
UPDATE Field Names	<p>List of field names which, when used to build a query using data in the external file, select a single record from the ServiceCenter file for update. The field names and their values are <i>ANDed</i> to build a query.</p> <p>For example: To uniquely select an Incident record, you must provide both an Incident number and a page number. The field names list would include number and page. These field names <i>must</i> exist in the import descriptor.</p> <p>Note: Do not enter the list of fields to update into this array. Rather, enter the field names that are included in the import that uniquely identify the record to update.</p>

Field	Description
Expressions	<p>These fields are used to initialize variables needed to build a query for use in selecting records for update. The external import record is always referred to by the variable <i>\$import.string</i>.</p> <p>Assume, for example, that an import is being defined to update records in the location file. The external file contains the location name beginning in position 12 for a length of 20.</p> <p style="padding-left: 40px;">1324ZX-12 detroit 04/14/95 12:152325 E. Main St.</p> <p>You want to build a query for each record that selects by location name. Your expressions will be:</p> <pre style="padding-left: 40px;">\$import.loc.name=substr(\$import.string, 12, 20) \$n=strtrmr(\$import.loc.name)</pre> <p>The first statement assigns the value detroit to <i>\$import.loc.name</i>. The second statement truncates the trailing spaces, with the result that <i>\$import.loc.name</i> is detroit.</p>
Query	<p>This query is performed against the ServiceCenter file to select a record for update. One and ONLY one record must be selected in order for update to occur; if no record or more than one record is selected, the Mode option is used to determine action.</p> <p>Using the example above for the location file, the query would be:</p> <pre style="padding-left: 40px;">"location=\""+\$import.loc.name+"\""</pre> <p>for detailed information about forming statements and using functions, see <i>ServiceCenter System Tailoring</i>.</p>
Skip	<p>This condition is evaluated each time a record cannot be selected using the Query; if true, the record is skipped. If the condition is false or unknown, the Mode field is used to determine whether to add or replace.</p> <p>Using the location example, if no locations with a location code containing ZX should be added or replaced, the Skip condition might be:</p> <pre style="padding-left: 40px;">index("ZX", substr(\$import.string, 1, 11))>0</pre>
Note:	<p>The following fields appear on both import and import.extended forms.</p>
Test Import	<p>Displays the result of the test import. Using a fixed font for test output makes it easier to interpret test results.</p>

Deleting an Import Descriptor record

In most cases, you will reuse an existing import record to perform an import numerous times. However, you may want to delete an import record that you will no longer need from your database.

The screens below illustrate how to delete the `abends.problem` import record from the problem file.

To delete an Import Descriptor record:

- 1 Open Database Manager.
- 2 Enter `=import` at the Form name prompt.
- 3 Press **Enter** or **Search**.
- 4 To display an existing record, press **Enter** to display a QBE list of import records,
 ~ or ~
 Enter the name of an existing record (e.g., `abends.problem`) and press **Enter**.
 In either case, if a record exists, ServiceCenter will display the record.
- 5 Press **Delete** to delete this record from the import file.
 You will be prompted to confirm the delete operation.
- 6 You can cancel the delete by pressing **No**.
 To confirm the delete, press **Yes**.
 The message *Record deleted from the import file* will appear, along with a QBE list of records that still exist in the import file.

Import records in Foreground

When you import records in the foreground, the amount of time it takes to import the records will vary, depending on how many records are in the external file.

The screens below illustrate how to import records from the *problem* file with the category *abends*.

To import records in foreground processing:

- 1 Open Database Manager.
- 2 Select **Options > Text Import Wizard**.
- 3 Enter the required data in the fields provided.
- 4 Press **Import**.

The first screen of the Import Wizard appears, as described under *Importing a File with the Wizard* on page 174. The record you previously created is now available for use.

Import records in Background

When you schedule an import to run as a background operation, you can continue working in ServiceCenter during the import.

To import records in background processing:

- 1 Open Database Manager.
- 2 Select **Options > Text Import Wizard**.
- 3 Provide the information requested at each Import Wizard screen as described under *Importing a File with the Wizard* on page 174.
- 4 At the **Run this Import** screen (import.wizard.run.g) (Figure 6-10 on page 190), enter a date and time in the **Expiration** field.

5 Click Schedule.

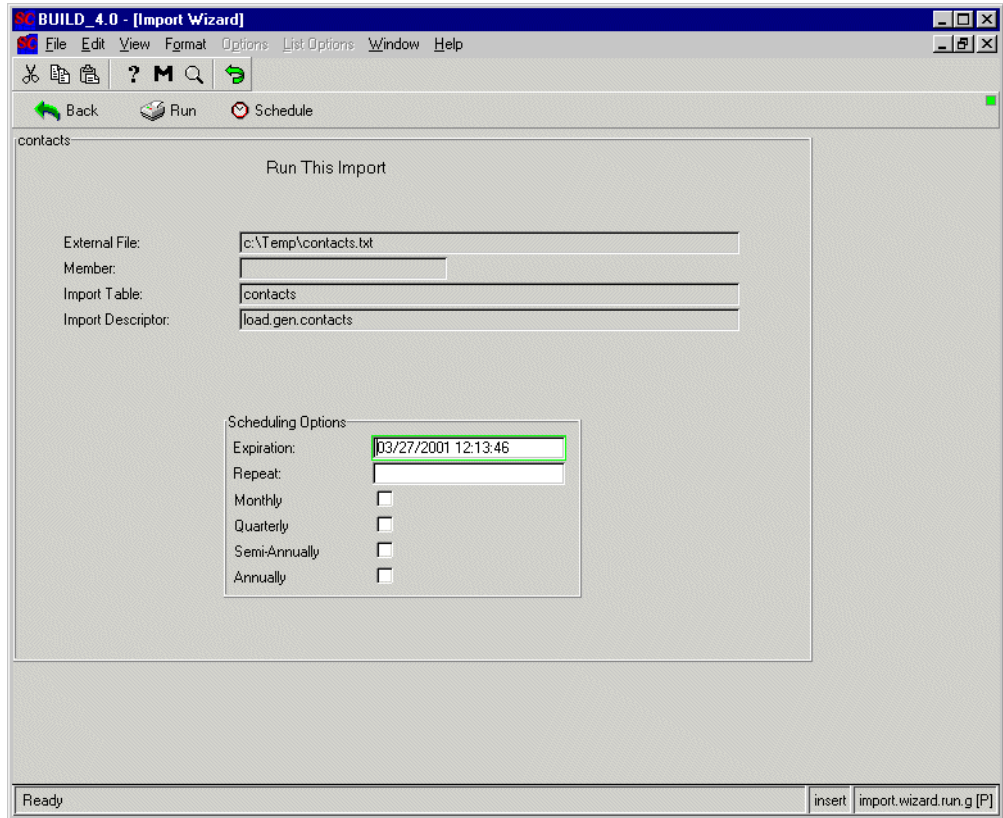


Figure 6-10: Scheduling import run

You will be returned to the Database Manager prompt screen.

A message will appear at the bottom of the screen indicating the *import has been scheduled for future processing* and display the scheduled time.

Print an Import Descriptor record

You may want to print your import descriptor record when you encounter a *Validation failed* error message during the import process. You can then check the fields on your import descriptor record against the fields in the Database Dictionary.

To print an Import Descriptor record:

- 1 Open Database Manager.
- 2 Enter =import at the Form name prompt.
- 3 Press **Enter** or **Search**.
- 4 To display an existing record, press **Enter** to display a QBE list of import records, or enter the name of an existing record (*abends.problem*) and press **Enter**.

In either case, if a record exists, ServiceCenter will display the record.

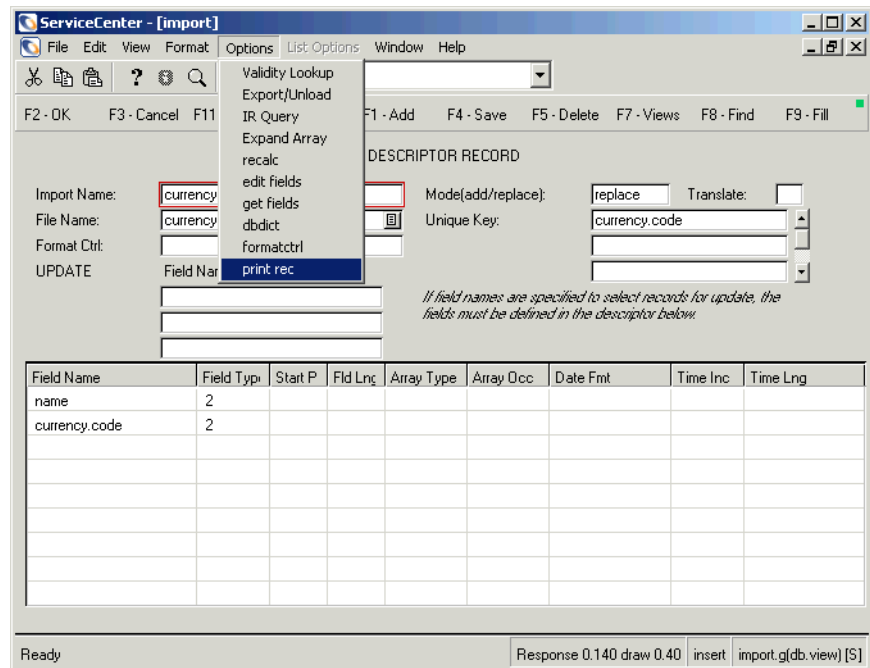


Figure 6-11: Import descriptor record—print option

- 5 Select **Options** > **print rec**.

Notice the message stating the scheduled time for the print job.

Using Format Control with Import

Due to the complexity of Format Control, review the Format Control section of *ServiceCenter System Tailoring* thoroughly before proceeding. Some of the issues to remember when using Format Control with import are:

- Always enter *true* for the **Add Format Control option** when using validation and calculation routines with import.
- Enter the name of the Format Control record in the appropriate field of the import descriptor record.

To add Format Control records:

Note: The following steps use the *abends.problem* file as an example.

- 1 Select the Toolkit tab.
- 2 Click **Database Manager**.
The Database Manager prompt screen will appear.
- 3 Select **Text Import Wizard** from the **Options** menu.
- 4 In the **File to Import** field, enter the path and name of the file containing the contact names that you want to import.
- 5 In the **Destination Table** field, enter the name of the table in ServiceCenter into which you want to load your data.
- 6 If the external file is an OS/390 partitioned dataset, enter the name of the member in the **Member** field. Otherwise, leave this field blank.
- 7 In the **File Type** field, select the operating system you are running on: Windows, OS/2, Mac, and so on. This field is optional.
- 8 Click **Search** or press F6.

If more than one import descriptor exists for that file, a QBE list of records is displayed.

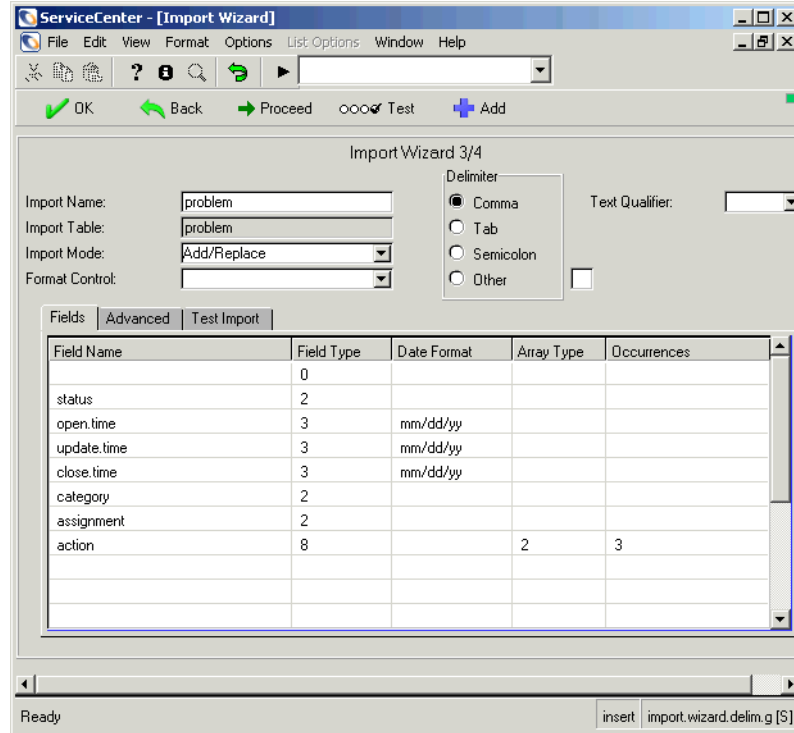


Figure 6-12: Import Descriptor record—defining Format Control

- 9 Move the cursor to the **Format Control** field.
- 10 From the dropdown list, select the name of the Format Control record you want to use for this import descriptor record (*language.import* in this example).
- 11 Press **Save** to record this change and update the record in the import file.

The message *Record updated in the import file* will appear at the bottom of your screen, as long as the Format Control record indicated exists.

Important: If the record does not exist, it needs to be created before the import descriptor record can be saved correctly.

7 Telephony

CHAPTER

Overview

ServiceCenter Telephony (SCT) provides options for computer-telephone integration, so that your PC can be set up to answer incoming telephone calls. Using Microsoft's TAPI add-on feature as an interface for the 32-bit ServiceCenter client, such as clients running on Windows NT, will allow you to use ServiceCenter Telephony. By simply clicking on a button with your mouse, you can answer a call, place a caller on hold, return to that call or hang-up.

A common method for integrating third-party software is through the use of DDE protocol with ServiceCenter System Events. For specific information on ServiceCenter DDE support and System Events, see the *System Tools* section of *ServiceCenter System Tailoring*.

This chapter discusses the TAPI aspect of ServiceCenter Telephony.

TAPI

Prerequisites

SCT is compliant with *Microsoft's Telephony API (TAPI)*, but you must have installed a Telephony Service Provider from your switch vendor that supports TAPI. Each vendor has software for their specific switches, which is installed on individual workstations. You can set this up by going to the Windows Control Panel and selecting the Telephony button. The Hardware and Software requirements depend upon each company and each switch. The benefit of using TAPI is that the underlying physical or logical connections are hidden from the application. Be sure you have completed the following before you begin using SCT.

- Checked your operating platform requirements to make sure that the *service provider* software will run on your system.
- Installed a Telephony Service Provider from a switch vendor that supports Microsoft's TAPI.
- Installed TAPI on your PC.
- Connected your PC to a telephone, as described by your telephone switch vendor.
- Made sure that you have a TAPI compliant switch. See [Connectivity](#) on page 201 for a partial list of TAPI compliant switches.

Configurations

Configurations goes over the basics to help you to learn about and see:

- Four basic connection diagrams
- The flow a call takes, depending upon how your company has set up its phone system
- The architecture of a system, from PC/Workstation down to *Telephony Service Provider* (phone switch company)
- What you need to get the phone connected to your PC/Workstation
- Enabling TAPI for ServiceCenter
- Receiving and making calls
- Debugging tools provided to help diagnose problems you may encounter

Figure 7-1 on page 197 diagrams the four basic connections you will find. The phone-centric model, which uses a physical connection between the phone and the workstation, is the easiest to set up and the most common initial implementation. The voice-server model, which uses a logical connection to associate a call with a workstation, is more robust and scalable and is best suited for large implementations. You can find further detail in the Providing Connection Model Independence section in Microsoft's TAPI 2.1 White Paper located at <http://www.microsoft.com/communications/tapi21learnmore.htm>.

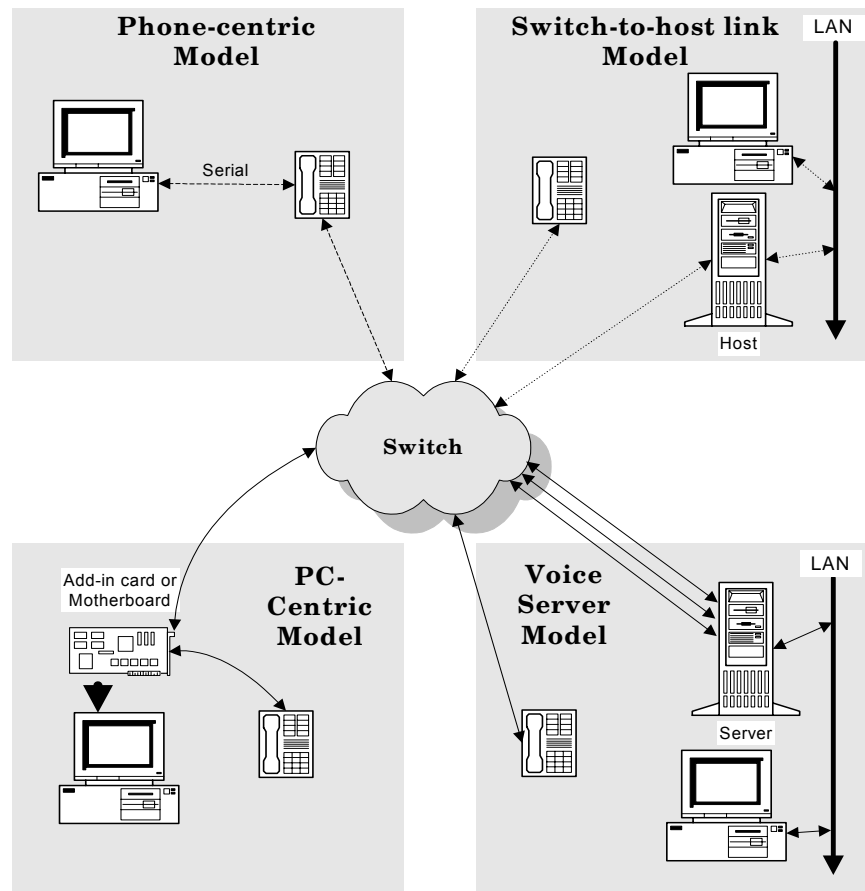
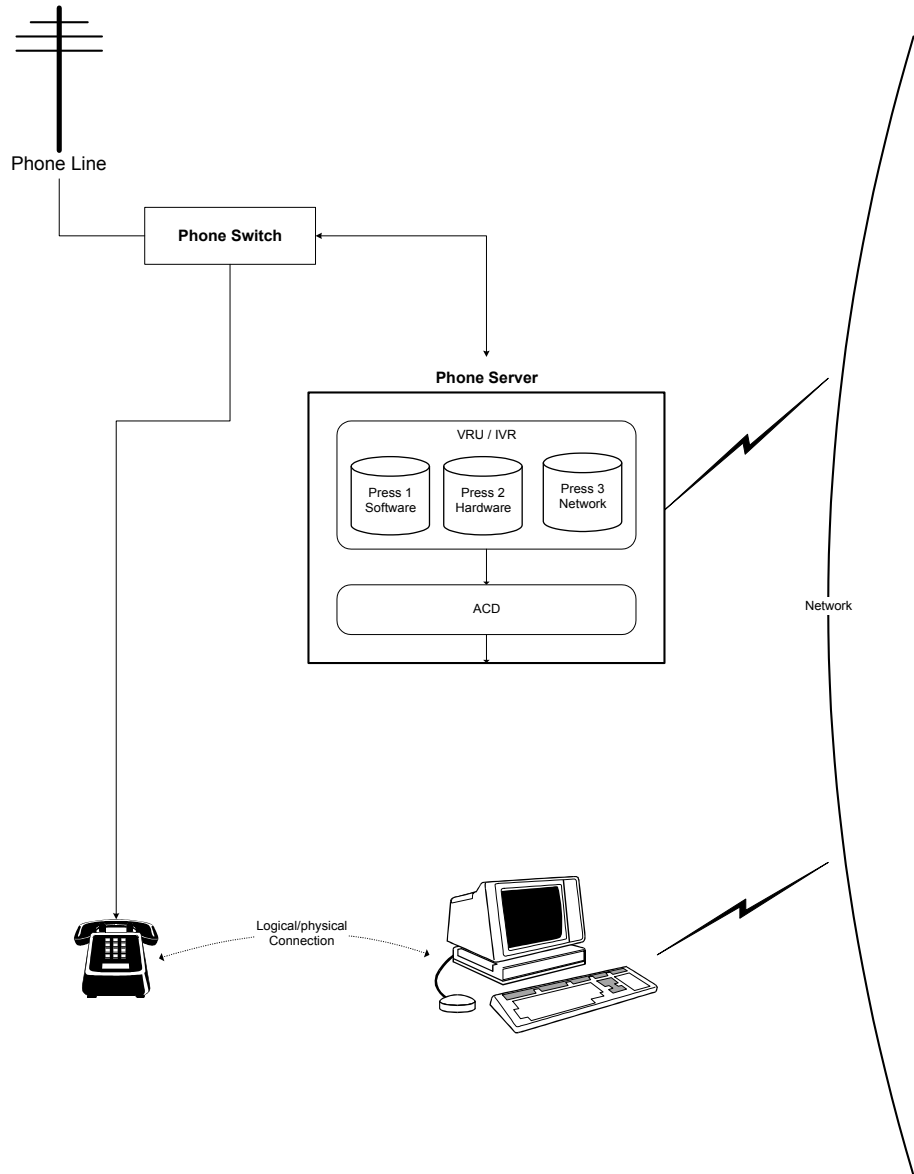


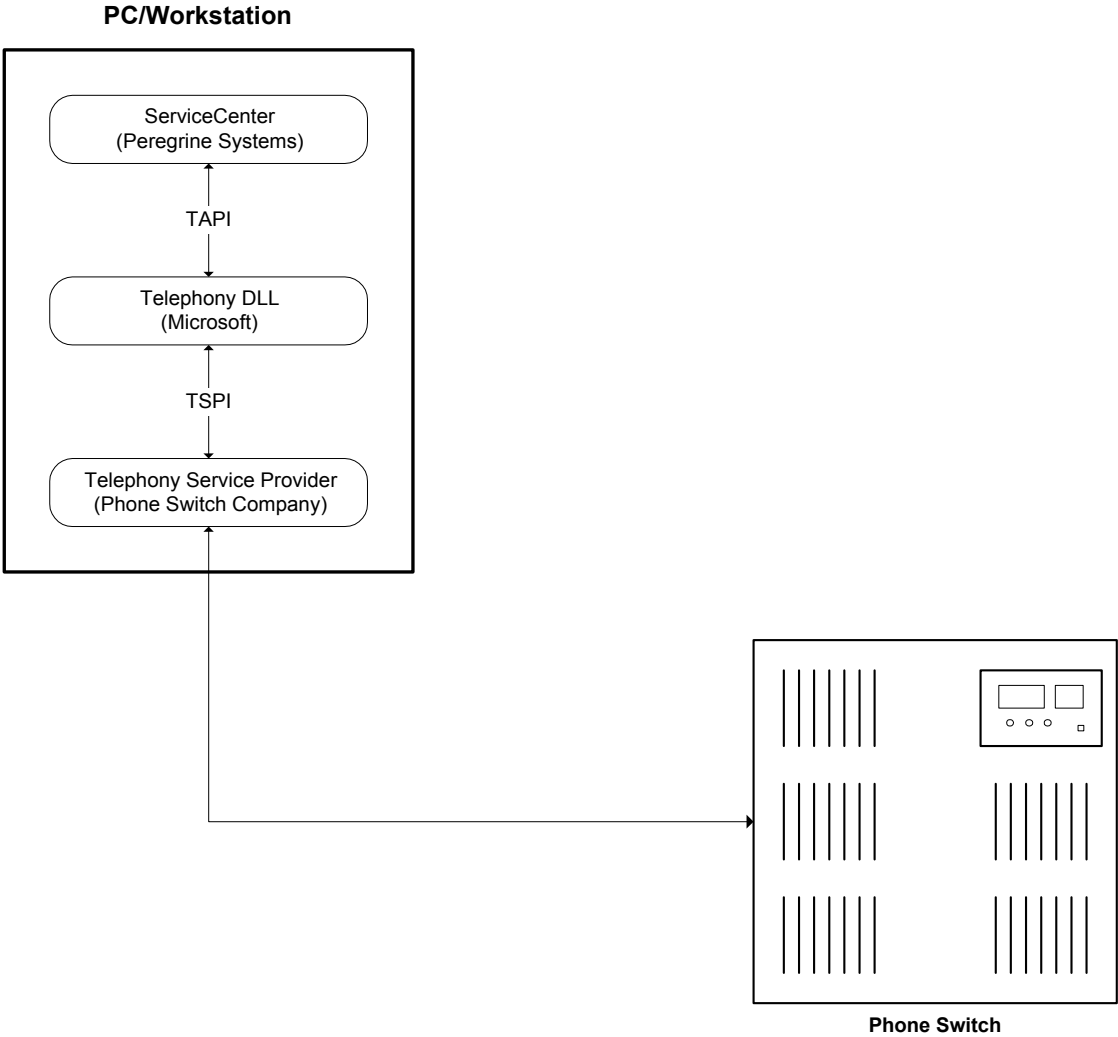
Figure 7-1: TAPI configurations

Call flow

The flow a call takes depends on how your company has set up its phone system. Most companies use a combination of a VRU and an ACD to route the call to an individual. Once that person receives a call, ServiceCenter displays a dialog box showing the caller's name and number (if available).



Architecture



There are two communication protocols used to enable an end-user application (ServiceCenter) to communicate with the phone switch:

- Telephony Application Programming Interface (TAPI)—used by ServiceCenter to communicate with the telephony DLL that Microsoft provides with its operating systems.
- Telephony Services Programming Interface (TSPI)—used by the phone switch to communicate with the telephony DLL. This is accomplished through the Telephony Service Provider from your phone switch which is installed on each workstation. This is similar to a device driver, in that it translates proprietary function calls from the switch into function calls that the telephony DLL can understand and then to TAPI so that any end-user application can use them.

When a call comes in from the phone switch to a person's workstation (no matter which physical model is used) the Telephony Service Provider picks up the call. It uses TSPI function calls to communicate the call information to the telephony DLL. In turn, that uses TAPI function calls to communicate with ServiceCenter.

ServiceCenter communicates with the telephony DLL using only the TAPI function call protocol. Whenever a ServiceCenter user makes a call, answers a call, puts a call on hold, or uses the ServiceCenter shortcut window for any purpose, a TAPI call to the telephony DLL occurs. The telephony DLL uses corresponding TSPI calls to communicate with the Service Provider, which then translates those into specific calls to communicate with the phone and/or phone switch.

When ServiceCenter receives a phone call, a dialog box is displayed that identifies the caller and displays the caller's telephone number and the status of the call. (SCT has three statuses: Incoming, Connected, and Holding.) When the ServiceCenter user answers the call, ServiceCenter passes all that information to the ServiceCenter application layer using System Events (see *System Events in ServiceCenter System Tailoring, Volume 2*).

Note: Neither TAPI nor TSPI protocols should be confused with TSAPI. TAPI and TSPI are from Microsoft. TSAPI is Novell's telephony programming interface. This is a different standard which is not supported by Peregrine Systems.

Connectivity

Before you can use SCT, your PC must be TAPI enabled in the following manner:

- Your PC must have *service provider* software installed. This software *listens* for incoming calls and is provided by your TAPI switch vendor. Check the operating platform requirements to ensure that the service provider software will run on your system.
- Your PC must be connected to your telephone as described by your telephone switch vendor. The connection can be through your company telephone switch or through a direct connection, such as a serial cable between the PC and the telephone (Figure 7-1 on page 197).
- Support is provided for any TAPI compliant switches. The following is a list of tested switches:
 - Nortel Symposium
 - Nortel Meridian 1
 - Lucent PassageWay
 - Siemens Hi-Com 150E
 - Siemens Hi-Com 300E

Enabling TAPI for ServiceCenter

Use either of the two following methods for enabling ServiceCenter Telephony (TAPI):

- Add the Telephony parameter to your ServiceCenter client initialization (sc.ini) file.
- Add the Telephony parameter to the target path of your ServiceCenter client shortcut.

Editing the initialization file

To add the Telephony parameter to the initialization file:

- 1 Log off from your ServiceCenter client session.
- 2 Access your client sc.ini file using a text editor.
- 3 Add the following parameter to the bottom of your file (Figure 7-2 on page 202):

telephony

```

sc.ini - Notepad
File Edit Format Help
#
# ServiceCenter Initialization file
#
# Copyright (c) 1997-2001 Peregrine Systems, Inc.
# All Rights Reserved
#
#####
#
# NOTES ON USAGE:
#
# Lines that begin with the pound sign are comments
#
# The system parameter below may be changed to a TCP/IP service name.
# TCP/IP service names are maintained in the services file.
# As specified here, Servicecenter is using a service port number.
#
# system:12670
#
# Type the authorization code supplied by your Peregrine account representative
# immediately following the "auth:" keyword below.
# Only numbers and the letters A thru Z may appear in the authorization code.
#
#auth expires xxx, 2001
#
#####
#
# Installed Configuration
#
path:c:\Program Files\ServiceCenter\DATA
shared_memory:24000000
ntservice:ServiceCenter
log:c:\Program Files\ServiceCenter\sc.log
bitmap_path:c:\Program Files\ServiceCenter\BITMAPS
system:12670
auth:JEOU9LL4X JF6E11230IS8X MB476HDF
scauto:12690
telephony
  
```

Figure 7-2: ServiceCenter initialization file

- 4 From the File menu, select Save.

Important: Save the file as a *text file*, not in the native format of the editor if you are using a word processor, such as MS Word or Write.

- 5 Restart your ServiceCenter client session, as this saves the settings.

Editing the Shortcut

To add the Telephony parameter to the ServiceCenter shortcut:

- 1 Right-click the client shortcut, and choose **Properties** from the menu.
- 2 Select the Shortcut tab in the Properties window (Figure 7-3 on page 203).
- 3 Add the following parameter to the Target path:
 - telephony

For example: "C:\Program Files\ServiceCenter\Run\scguiw32.exe"
-system:falcon.12670 -telephony

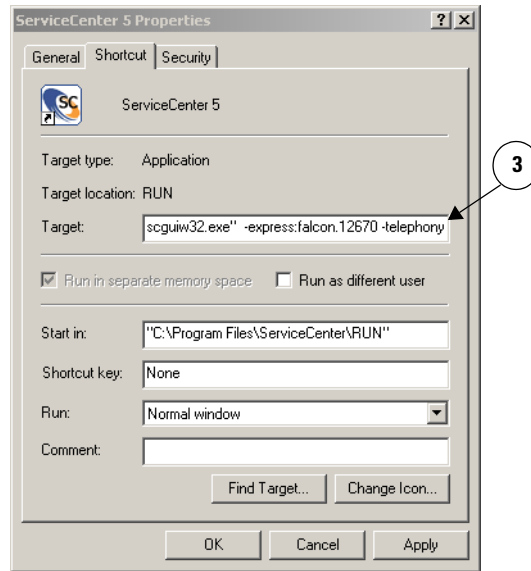


Figure 7-3: Telephony shortcut properties

- 4 Click OK.

Receiving Calls

Using ServiceCenter telephony (SCT) is as easy as clicking your mouse. Answer incoming calls from a dialog box on your screen. ServiceCenter automatically launches Service Management and displays a new call report, allowing you to enter data while you are still talking to the caller. Call reports are discussed in the Service Management chapter of the *ServiceCenter Application Administration Guide*.

Other applications and utilities can be launched with SCT by setting up different system event records. For more information on system events, see the System Tools section in *ServiceCenter System Tailoring*.

To receive a call:

- 1 When a call comes in, a dialog box is displayed on your screen, displaying the name of the caller, the caller's telephone number, and the status of the call. SCT has three statuses: **Incoming**, **Connected** and **Holding**.

- 2 Click **Answer**.

You now can talk to the caller (Jonathon Irwin) via your speaker phone.

The message in the dialog box indicates that the caller is connected and two new buttons are displayed: **Hang Up** and **Hold**.

ServiceCenter automatically accesses Service Management and displays a new call report with the name of the caller already entered. If the caller is in your ServiceCenter database, related information is automatically entered in the record.

- 3 If a second call comes in while you are still on the phone, that caller's name, number, and a status of the call are displayed in the dialog box. The red arrow indicates that the second call is ringing, and the buttons become active for that call.

- 4 Click **Answer**.

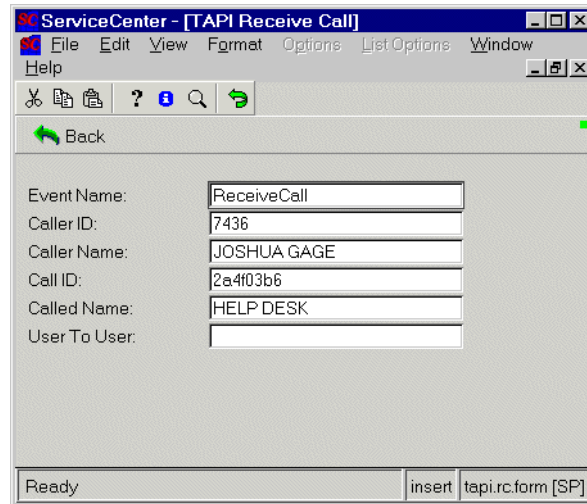
You now can talk to the second caller.

- The first caller is placed on *hold*. That caller remains listed with a status of **Holding**.
- The call report for the previous call is saved and a new call report is opened containing the information for the new caller.
- The status for the new caller changes from **Incoming** to **Connected**.
- The **Answer** button is replaced by the **Hang Up** and **Hold** buttons.
- If another call comes in, the process is repeated. You can have more than two calls listed.
- Click the caller's name to display the buttons for that call.

Note: If you return to a previous call, Service Management does not return to the previous call report. You must access the previous report through the application.

Background Processes

Once the user answers the call, all information about the call is sent to the System Event Handler (SEH) which invokes the corresponding RAD application. You do not have to be on a particular screen for this to happen; the application is started in a new RAD thread.



When you answer a call, ServiceCenter telephony sends a Receive Call event to the SEH. When the RAD application is invoked, it can use the `event.name()` and `event.value()` functions to extract the information about the call. Following is a table of the parameters ServiceCenter telephony sends with this event. When extracting this information, be sure to spell (capitalization and spaces) the names as you see them here. See the RAD application `tapi.receive.call` for an example.

Parameter	Description
Caller ID	Phone number of the person calling
Caller Name	Name of the person calling
Call ID	Unique number that identifies the call. This is reserved for future use.
Called ID	Name of the person being called (you). Some switches use this for additional information.
User To User	Used to receive special information, if any.

Making Calls

Use your RAD application to send a system event to the SEH, which will pass it along to SCT, which then makes the call. The event name is **MakeCall**. The following table contains the parameters SCT expects. See the RAD application **tapi.make.call** for an example. For more information on system events, refer to *System Tools* in *System Tailoring Volume. 2*.

Parameter	Description
Number	Phone number of the person to call
Name	Name of the person to call. This is for cosmetic reasons only. If SCT receives the name of the person from the phone switch it will replace the user supplied name with the switch supplied name in the call box.

Debugging

Occasionally you may encounter a problem with your phone switch. We have provided debugging tools to help diagnose these problems.

ServiceCenter

Two debugging parameters have been created to help diagnose any problems you might be having. The first, **Telephony**, is used to log messages about what is going on with the phone calls. These messages are written to the client log file (**sc.log**) defined in your **sc.ini** file.

To invoke this debugging, put **SCdebug:Telephony:#** in your client **sc.ini** file. The following table explains the different levels of messages that the **#** sign represents.

In the **sc.ini** file:

#	Description
1	Record error messages to the log file
2	Record all success messages to the log file
3	Record all TAPI events and messages to the log file
4	Record the TAPI system capabilities to the log file

The second parameter for debugging is for the Call Box. Using the same format as above, put `SCdebug:TeleDialog:#` in your `sc.ini` file as well. These parameters can be combined on one line in your `sc.ini` file:

```
SCdebug:Telephony:#:TeleDialog:#
```

#	Description
1	Record call and call box creation and destruction in the log file
2	Record call box movement and growth in the log file

Microsoft TAPI Debugger

Microsoft also has a debugging tool called the TAPI Browser. It provides a raw list of all the TAPI function calls and lets you invoke them one at a time to test out your phone switch. There are executables for each version of TAPI 1.3, 1.4, 2.0, they are called `tb13.exe`, `tb14.exe`, and `tb20.exe` and can be found on the Microsoft web site.

To use the MS TAPI Browser, double-click the application to start it.

To receive a call:

- 1 Click **LAp+** at the top of the screen.

This returns:

```
lineInitialize returned SUCCESS
num line devs = 1
```

Note: The example messages won't be exactly what you receive. Some of the hex values will be different, but you should get the same general messages.

- 2 Select the **Parms** check box at the top of the screen.
- 3 Click **Line+** at the top of the screen.
- 4 In the resulting dialog box, click `dwPrivileges` in the **Parameters** section.
- 5 In the **Bit Flags** section make sure only the **Owner** option is highlighted.
- 6 Next, select `dwMediaModes` in the **Parameters** section.
- 7 In the **Bit Flags** section make sure only the **InteractiveVoice** option is highlighted.

- 8 Click OK. This returns:

```
lineOpen returned SUCCESS
```

- 9 Clear the **Parms** check box at the top of the screen.
- 10 Make your call to the test phone.

You should see the following shortcut on the right pane of the TAPI Browser:

```
received LINE_CALLSTATE : hDev=x2f8f1958, cbInst=x0
```

```
p1=x2, OFFERING
```

```
p2=x0,
```

```
p3=x4, OWNER
```

```
lineGetCallInfo returned x0, hLine=x2f8f16c8
```

```
received LINE_CALLINFO : hDev=x2f8f1958, cbInst=x0
```

```
p1=x18000, CALLERID CALLEDID
```

```
p2=x0,
```

```
p3=x0,
```

- 11 Double-click **lineAnswer** in the left pane of the TAPI Browser window.

You should see:

```
lineAnswer returned x1
```

```
received LINE_CALLSTATE : hDev=x21f71958, cbInst=x0
```

```
p1=x100, CONNECTED
```

```
p2=x0,
```

```
p3=x0,
```

```
received LINE_REPLY : hDev=x0, cbInst=x0
```

```
p1=x1,
```

```
p2=x0,
```

```
p3=x1,
```

You now should have successfully received and answered a call.

If this has worked, yet you cannot receive and answer calls using ServiceCenter, there is a problem with ServiceCenter or the way it has been set up. Contact Peregrine Systems Customer Support and provide them with this information as well as the ServiceCenter log containing the Telephony debug information.

If you were unable to receive and answer a call using this TAPI Browser, there is a problem with your TAPI setup or your phone switch. Contact your phone service company with this information.

- 12 To hang up the call and close down the TAPI Browser, click **Call-**, **Line-** and **LAp-**.

TeleCenter Manages Your Telecommunications System

Peregrine Systems' TeleCenter application is a 32-bit software application designed to effectively manage your telecommunications system by incorporating industry-standard call costing capabilities with comprehensive billing and management functions. To learn more about the TeleCenter application and its modules, see the TeleCenter user's guides. The following provides a brief explanation of some of the modules within TeleCenter.

- **Cost Allocation:** This module offers several maintenance options that pertain to costing. For more information about this module, see the *TeleCenter Cost Allocation User's Guide*.
- **Costing Setup:** This module provides an interface to your costing tables. For more information about this module, see the *Costing Setup User's Guide*.
- **IDC Recording:** This module records, loads, and translates your call records. For more information about this module, see the *TeleCenter IDC Recording User's Guide*.

8 LDAP Server Interface

CHAPTER

Overview

Lightweight Directory Access Protocol (LDAP) directories provide a centralized source for information about the people within an organization. Information, such as e-mail addresses, telephone numbers, fax numbers, user IDs, and passwords can be defined in the LDAP directory and referenced by various applications, eliminating the need to maintain user data in more than one location.

ServiceCenter has been configured to interface with LDAP directories to simplify the creation of operator records for large groups of users. This is particularly useful for other groups with similar system access requirements.

Note: LDAP functionality is unavailable on OS/390.

How LDAP Works

ServiceCenter stores information about its users in the **operator** file (for information about the *operator* file, see the *Getting Started* chapter of *System Administrator's Guide, Volume 1*). In addition to general contact information, an operator record defines specific ServiceCenter user attributes, such as capabilities, startup menu, hourly rate, and Mandanten (system database protection) rights. Creating individual operator records for large numbers of users can be impractical.

Using an LDAP directory allows users without operator records to log on to a ServiceCenter server by providing minimal login information, such as login ID and password. Certain fields in the ServiceCenter operator record are *mapped* to attributes already set in the LDAP directory, so the information stored may only include the login ID and password or contain other general contact information taken from the operator file.

Note: When a first-time user logs on to a ServiceCenter server configured to use an LDAP directory, an operator record is created based on a new user template if it is defined in ServiceCenter. If the template does not exist but the system default record does, ServiceCenter will use the system default record to create the operator record. If neither a new user template nor the system default record exists, the appropriate operator record will not be created.

Important: When you set up LDAP as a primary data source for operators to log on to ServiceCenter, make sure you set Allow Access Without Operator Record as *true* in the Incident Management environment. If this is not selected (unchecked), users are logged on with no profile. For more information on configuring the Incident Management environment, see the *ServiceCenter Application Administration Guide*.

Template

ServiceCenter systems that are available to large numbers of users profit from the use of templates by creating classes of users. A template provides a set of default characteristics for logins authenticated by an LDAP server. When a user logs on to ServiceCenter using an operator mapped to an LDAP directory, the LDAP data is merged with a template defined in the ServiceCenter System Wide Company Record, and a new operator record is created. The user is automatically granted the capabilities defined in the template. You can modify capabilities for any group using a template simply by changing the template.

System default records

If you choose not to define a template for users logging on to ServiceCenter with LDAP authentication, you may define a *system default record* (*SYSDEFAULTS) for the operator file. ServiceCenter uses the system default record much like a template and copies the LDAP attribute data to the record to produce a permanent operator record in the ServiceCenter database. Unlike the template, however, you cannot perform mass updates by editing the system default record.

Updating mapped fields

Edits to fields in an operator record that are mapped to LDAP can only be made by users with LDAP administrative privileges and are automatically updated in the LDAP directory. For example, if you change the telephone number for an LDAP user in the ServiceCenter operator record, the number is updated in the LDAP directory.

If new operator records are created from within ServiceCenter, new entries are created within the LDAP directory for all currently mapped fields. Deleting operator records from within ServiceCenter, however, has no effect on LDAP data.

Note: An LDAP server is a shared resource whose data may be used by applications other than ServiceCenter. Deletions of LDAP data should be performed on the LDAP server only by an administrator familiar with the applications that access the LDAP directory.

Mapping other files to LDAP

You may map fields from other files (such as the **contacts** or **device** files) to an LDAP directory to take advantage of centralized data management. You can map a ServiceCenter file to only one LDAP server.

Under certain circumstances, you may wish to map a ServiceCenter file to multiple LDAP base directories. Generally, all LDAP data is extracted from the LDAP base directory specified when the file is mapped; however, you can change this on a per record basis by adding a field to the ServiceCenter file called **sldapbase**. When this field is populated with the appropriate base directory, ServiceCenter constructs the distinguished name (DN) needed to retrieve the data from LDAP by using this value rather than the base directory named during the mapping.

Performing Data Queries

In order to successfully integrate LDAP data with ServiceCenter data, the following process occurs during the execution of a query.

If LDAP is not the primary data source

If LDAP is not the primary data source, the query is issued against the ServiceCenter local data whether it be P4, Oracle, SQL server, and so on. After the matching records are retrieved from the local data, ServiceCenter sends a query request to LDAP using the mapped key field information from the local data as selection criteria. The LDAP returned data is then merged with its appropriate local record, and this data is displayed as the query result in ServiceCenter.

Note: If the data in a mapped field differs between LDAP and ServiceCenter, ServiceCenter will overwrite the LDAP data with local data since ServiceCenter is set to be used as the primary data source.

If LDAP is the primary data source

If LDAP is the primary data source, ServiceCenter sends a query request to LDAP using the mapped fields in the query as selection criteria. ServiceCenter then performs a query on its local data using the mapped key field information as criteria. These records are then merged with the LDAP data and displayed as the query result.

If LDAP is set as the primary data source and the user executes a query that contains no LDAP mapped fields, ServiceCenter will issue a query to LDAP for all of its records. After merging these records with its local data, ServiceCenter will perform the actual selection criteria to retrieve the appropriate data. If either the LDAP server or ServiceCenter does not have its maximum record and time limits set to 0, inaccurate results, including LDAP size or time limit errors, can occur.

Note: If the data in a mapped field differs between LDAP and ServiceCenter, ServiceCenter will overwrite the local data with the LDAP data since LDAP is set to be used as the primary data source.

Setup

You set up your LDAP directory as follows:

- Define the default LDAP server that ServiceCenter uses. This is the default server for all files mapped to LDAP.
- Map individual fields from target files to your LDAP directory and define file-specific server information. You may map to different LDAP servers for different files. The default LDAP server is used if you do not provide the name of a file-specific server.

Defining the Default LDAP Server

Define the default LDAP server information for your system in the `sldapconfig` file. The host name, port number, and base node used in search criteria entered here are used by ServiceCenter as the default values for attaching to and accessing data from an LDAP server.

To configure the default LDAP server:

- 1 In the System Administrator's home menu, click the Utilities tab.

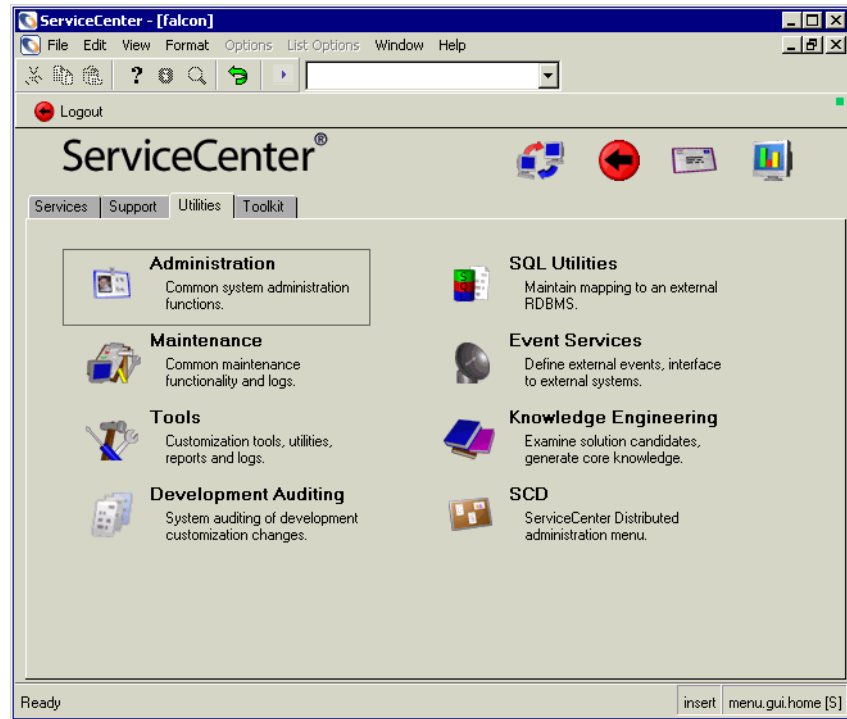


Figure 8-1: Utilities tab in the system administrator's home menu

- 2 Click Tools.

The Tools menu is displayed.

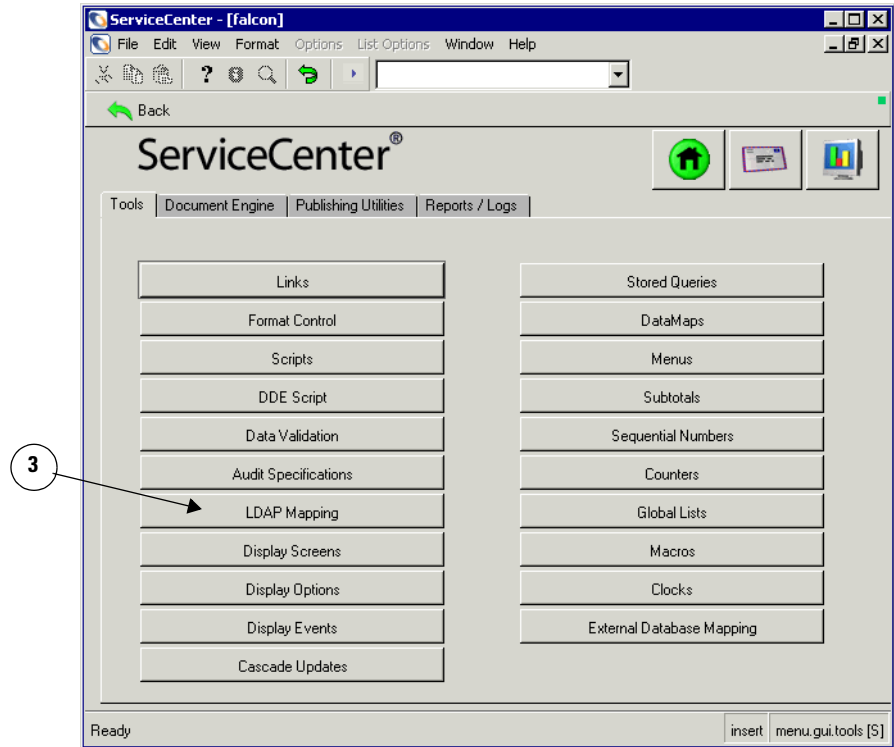


Figure 8-2: Tools menu

3 Click LDAP Mapping.

The LDAP configuration record is displayed.

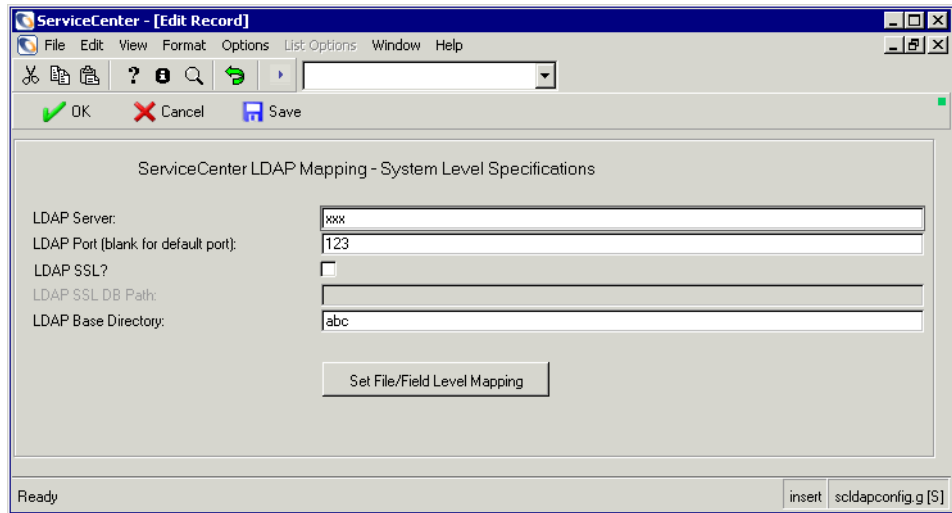


Figure 8-3: LDAP Mapping System Level Specifications record

- 4 Enter the appropriate default information into the following LDAP fields:

Field	Description
LDAP Server	Host address for the LDAP server with which the ServiceCenter client must communicate
LDAP Port (blank for default port)	Port address for the LDAP server. This is the port address where the server is listening for a connection.
LDAP SSL?	Select this box to use SSL authentication. Clear this box if you do not want to use SSL authentication.
LDAP SSL DB Path	If the LDAP SSL? box is checked, type the local path (relative to the ServiceCenter server) to your cert7.db compatible file. This file must contain a trusted root for the certificate your LDAP server will be sending to the ServiceCenter server for authentication.
LDAP Base Directory	Entry level in the LDAP directory where searches are initiated. Because directories are designed to have a global scope, it is often wise to specify searches that start at a low level in the directory tree. ServiceCenter will search the entire level below the starting point indicated, unless overridden by an individual file or field map setting.

- 5 Click **Save**.

The following message is displayed in the status bar: *Record updated in the sldapconfig file.*

Mapping individual Files and Fields to the LDAP Directory

Use the File/Field Level Mapping form to establish LDAP mapping for any file or field in ServiceCenter. If you map more than one file to an LDAP directory, you may define the server connection specifics for each one. ServiceCenter looks in this record for an LDAP server before using the default you established when setting up the LDAP base directory.

The following steps document one of the ways in which you can use the LDAP feature.

To map files and fields to an LDAP directory:

- 1 Click **Set File/Field Level Mapping** in the System Level Specifications record (Figure 8-3 on page 218).

The Field Level Mapping search form is displayed.

- 2 Enter operator in the **Name** field.
- 3 Click **Search** or press **Enter**.

The LDAP mapping record for the operator file is displayed (Figure 8-4).

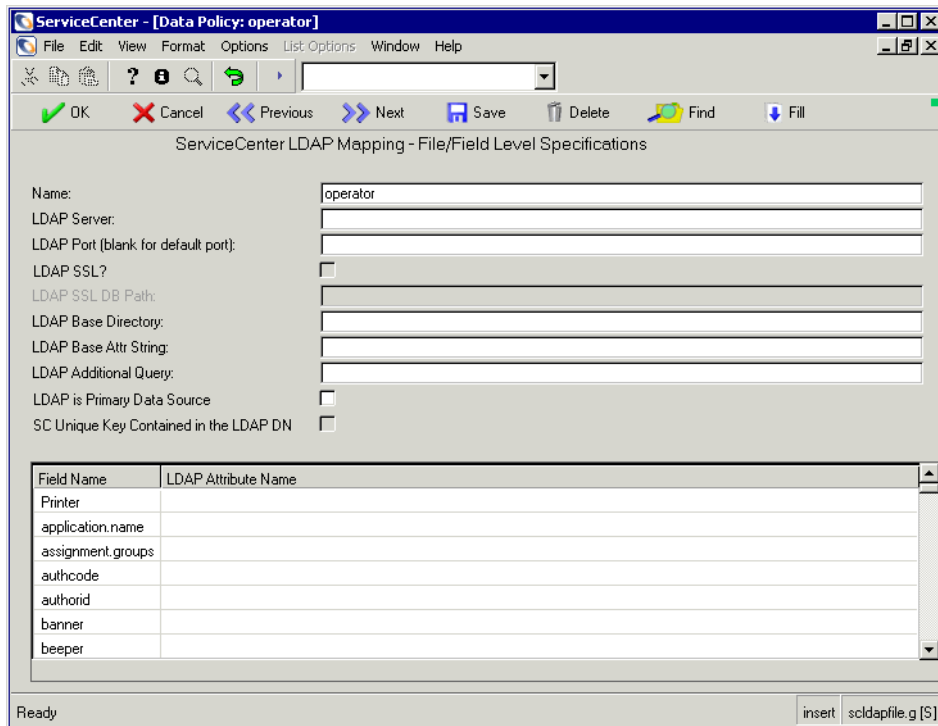


Figure 8-4: LDAP Mapping record for the operator file

4 Complete the following fields:

Field	Description
Name	Name of the ServiceCenter file whose fields you want to map to the LDAP directory. (Complete this field for an Add. It is unnecessary to complete the field for an Update, as is documented in this example.)
LDAP Server	Host address for the LDAP server containing the data for this file, if it is different from the default LDAP server. The value in this field overrides the system-wide server information defined in the LDAP Mapping System Level Specifications record (Figure 8-3 on page 218).

Field	Description
LDAP Port (blank for default port)	Port address for the LDAP server maintaining the data for this file, if different from the default LDAP server port. The value in this field overrides the system-wide server information defined in the LDAP Mapping System Level Specifications record (Figure 8-3 on page 218).
LDAP SSL?	Select this box to use SSL authentication. Clear this box if you do not want to use SSL authentication.
LDAP SSL DB Path	If the LDAP SSL? box is checked, type the local path (relative to the ServiceCenter server) to your cert7.db compatible file. This file must contain a trusted root for the certificate your LDAP server will be sending to the ServiceCenter server for authentication.
LDAP Base Directory	Base entry in the directory where searches are initiated, if different from the default base directory. The value in this field overrides the system-wide directory information defined in the LDAP Mapping System Level Specifications record (Figure 8-3 on page 218).
LDAP Base Attr String	<p>Default values for attributes that have not been mapped to ServiceCenter fields. If there are attributes required by LDAP that cannot be mapped to an existing ServiceCenter field, default values for these attributes must be entered in this field for LDAP to accept records inserted from within ServiceCenter. An example of such an attribute string might be:</p> <p style="padding-left: 40px;">Objectclass=top, Objectclass=person, Objectclass=organizationalPerson, Objectclass=inetOrgPerson</p> <p>Default attribute values will differ for each ServiceCenter file that is mapped to an LDAP directory.</p>
LDAP Additional Query	Additional query to be used, if LDAP is the target for all queries against this file.

Field	Description
LDAP is Primary Data Source	<p>Logical field that determines whether or not LDAP is the primary source for this file. If the check box is selected (<i>true</i>), LDAP is the target for all queries against this file. If the check box is not selected (<i>false</i>), the normal ServiceCenter database is the target for all queries against this file, and the corresponding LDAP entry is read to fill in the fields mapped to LDAP.</p> <p>Note: If you have specified LDAP as your primary data source and have mapped ServiceCenter operators to an employee file in LDAP, queries against the operator file will display <i>all</i> the names in the LDAP directory, including individuals who may not be able to access ServiceCenter.</p>
SC Unique Key Contained in the LDAP DN	Select to use the unique key from the distinguished name (DN) used in the LDAP authentication process that is built using the LDAP base directory name (specified in the mapping).
Field Name	Map the fields in the ServiceCenter file to the LDAP directory, using the attribute names defined for your LDAP server. If a field named sldapbase exists in this ServiceCenter file, the value from this field in each record will be used as the record's LDAP base directory, which will override the system-wide directory information defined in the LDAP Mapping System Level Specifications record. This will allow each record to have a different LDAP base directory value, based upon how the fields in the files are mapped. If an sldapbase field does not have a value for the record, the base directory assigned to the file or the default directory will be used instead.

Your LDAP server contains default attributes for certain generic fields. For example, the **name** field in the operator record may have an LDAP attribute of **uid**, while the **password** field might have an LDAP attribute of **userpassword**.

- 5 Click **Save** to create a new record.

The following message is displayed in the status bar: *Data Policy record updated.*

- 6 Click **OK** to exit the record.

Using LDAP with a Secure Socket Layer (SSL) Connection

Secure Socket Layer (SSL) connections enable you to protect sensitive and/or proprietary data across computer networks and systems. ServiceCenter allows for secure socket layer connections to LDAP servers as a means of keeping vital information secure.

Note: In order to establish an SSL connection between ServiceCenter and the LDAP server, a number of system steps are required. It is important that this process be handled by an administrator familiar with LDAP and its SSL procedures.

Connecting to the LDAP Server with an SSL Connection

- 1 Using your LDAP server documentation, set up the LDAP server so that it will allow SSL connections.

Note: Make sure to record the SSL port number for later use in ServiceCenter.

- 2 Complete the LDAP mapping record as follows:
 - a In the **LDAP Port (blank for default port)** field, set the LDAP port to the same port number used by the LDAP server for SSL Connections (the default is 636).
 - b Check (select) the **ldap SSL?** check box.
 - c In the **LDAP SSL DB Path** field, type the local path (relative to the ServiceCenter server) to your cert7.db compatible file.

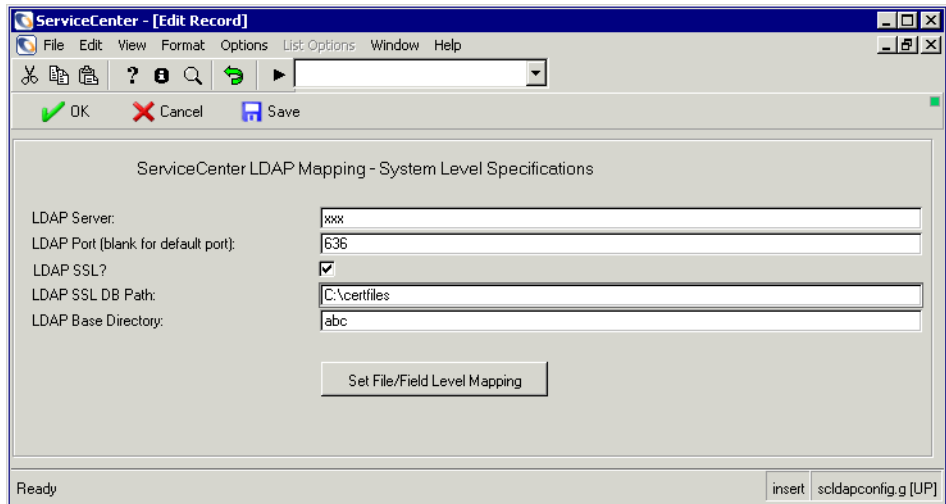


Figure 8-5: LDAP SSL Configuration

- 3 Shut down ServiceCenter, and then restart it.
- 4 Log in to ServiceCenter using your login information.
Users should now be allowed into ServiceCenter, and the use of SSL should be completely transparent to them.
- 5 Optionally, set the LDAP SSL? and LDAP SSL DB Path values to use SSL only on certain table connections.
The values on the mapping format also override any values entered using the LDAP Setup Format.

Control an LDAP Query with sc.ini File Parameters

There are two initialization (`sc.ini`) file parameters to control LDAP when performing a query. These parameters are specified in the server side `sc.ini` file.

Parameter	Description
<code>ldapmaxrecords</code>	<p>Defines the maximum number of records that should be returned by an LDAP query. If the parameter is not provided, or if a value of 0 is supplied, there will be no record limit.</p> <p>Note: The maximum number of records returned may be 1 less than the number specified by <code>ldapmaxrecords</code>. When a query is done against LDAP, some LDAP servers return a record that equates to the base directory that is being searched. This record is stripped by ServiceCenter, but does count as far as LDAP is concerned.</p>
<code>ldaptimelimit</code>	<p>Defines the maximum number of seconds that LDAP should spend processing a query. If the parameter is not provided or the value is set at 0 (zero), then there will be no time limit.</p>

Note: The maximum number of records or maximum time limit returned can also be restricted by the ldap server if it has a maximum set for either records or time. For example, if the `ldapmaxrecords` parameter is set to 2000 and the ldap server sets its maximum records limit to 1000, only 1000 records will be returned when processing an LDAP query. Likewise, if the LDAP server has a time limit of 120 seconds and ServiceCenter has its limit at 180, the query will time out if it exceeds 120 seconds.

Create ServiceCenter Records from LDAP Data

Once you have configured an LDAP server and have mapped individual fields from a file to an LDAP directory, you must decide how your ServiceCenter system should create new operator records from LDAP data. Typically, the only user data mapped to an LDAP directory is general contact information such as user ID, password, telephone number, or e-mail address. This is company-wide data, used by other applications that access the LDAP server. ServiceCenter-specific fields, such as those that define capabilities, menus, or security groups, are not likely to be mapped.

As soon as any field within the operator file is mapped to LDAP, ServiceCenter will start using LDAP to authenticate passwords. The distinguished name (DN) used in the LDAP authentication process is built using the LDAP base directory name (specified in the mapping) combined with the UID attribute (or attribute name which is mapped to the **name** field). There are instances when the DN used to bind to the LDAP directory should be different from the DN used to retrieve data. You can accomplish this by providing a different base directory name in the **ldapauthbase** (LDAP Base Name) field in the operator record.

When a new user, authenticated through an LDAP directory, attempts to log on to ServiceCenter, the system must somehow create a new operator record to allow that user appropriate access to the system. The authenticated user must be granted some predefined capability and a menu on the spot or login is not permitted. A complete operator record for a user authenticated through an LDAP server is created automatically by combining the general LDAP data with predefined ServiceCenter data using either of the following methods:

- Login template (for information, refer to *Login Template* on page 227)
- Default system record (*SYSDEFAULTS) (for information, refer to *System default records* on page 213 and *Defining a Default System Record* on page 232)

Parameters for Binding to the LDAP Server

Use the following parameters in the `sc.ini` file to bind to the LDAP server.

Parameter	Description
<code>ldapbinddn</code>	Name used to BIND to LDAP server
<code>ldapbindpass</code>	Password used to BIND to LDAP server

Login Template

A login template enables ServiceCenter to create an operator record automatically for first-time users with similar capabilities whose logins are authenticated by LDAP. When a new user attempts to log on, general LDAP data is merged with the ServiceCenter-specific data included in your template to create an operator record for that session. This record is added to the ServiceCenter database and remains there until manually deleted.

You may create a new template from scratch or copy an existing record and modify it accordingly. You can design a template for any type of operator, from a system administrator to a technician. A template allows you to assign a default set of capabilities and attributes to operators without having to define individual records. To change the capabilities of a group of operators, edit the template, and each record created from that template is updated.

Note: Only one login template may be defined at a time.

Creating a template

In this example, we use an operator record for user BOB.HELPDESK with limited capabilities.

Note: Templates created entirely within ServiceCenter are not visible when LDAP is selected as your primary data source. To view your template or the operator records of users not appearing in the LDAP directory, you must clear the **LDAP is Primary Data Source** check box in the File/Field Level Specifications form (Figure 8-4 on page 220). Reselect LDAP as your primary data source to view operator records of those users appearing in the LDAP directory. It is not necessary to log out when toggling between data sources.

To create an operator record template by copying an existing record:

- 1 In the system administrator's home menu, select the Utilities tab (Figure 8-1 on page 216).

- 2 Click **Administration**.

The administration menu is displayed.

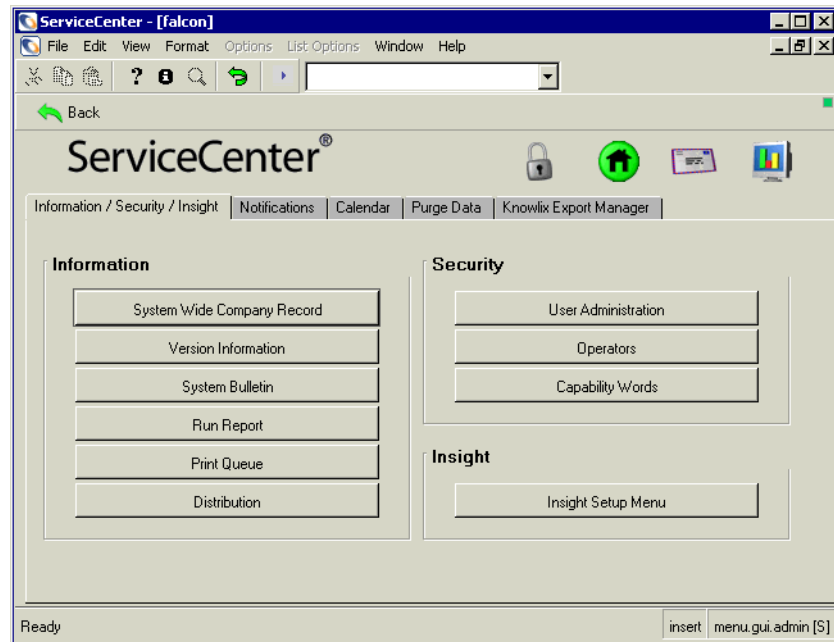


Figure 8-6: Administration menu

- 3 Click **Operators**.

A blank operator record form is displayed.

- 4 Enter the name of an existing operator in the **Login Name** field whose capabilities most closely match those you want to copy.

For this example, enter BOB.HELPDESK. This user is standard in the ServiceCenter system.

- 5 Click **Search** or press **Enter**.

The requested record is displayed.

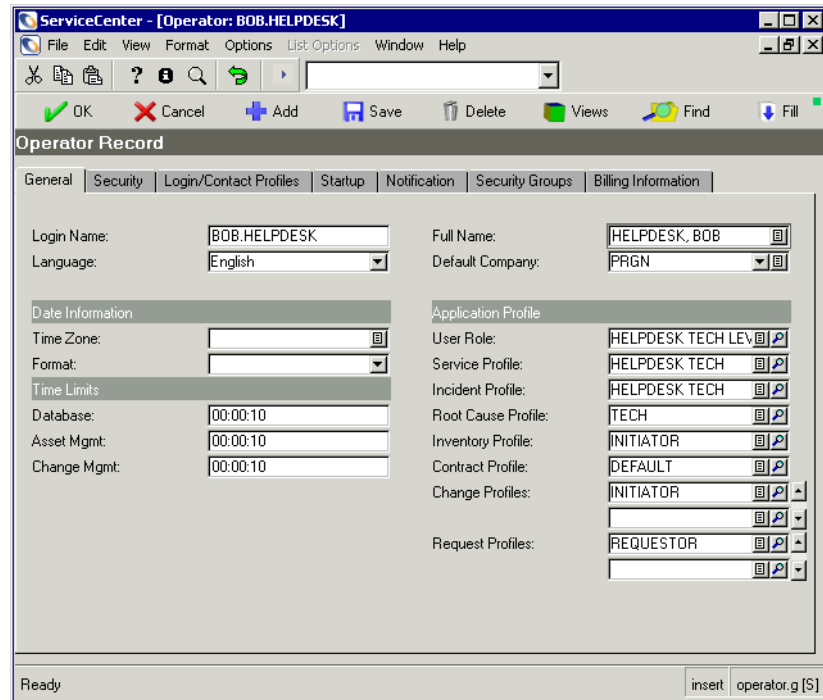


Figure 8-7: Operator record—displaying a user

- 6 Replace the user name (BOB.HELPDESK) with the name of your template.

In this example, we have called our template **NEWUSER** (Figure 8-8 on page 230).

Figure 8-8: Operator record template

- 7 Edit any other ServiceCenter specific fields accordingly.
- 8 Click **Add** to add your template to the operator file.

The following message is displayed in the status bar: *Operator record added.*

- 9 Click **OK** to exit the operator record.

Applying a Login Template

Once you have created your template, apply it to the System Wide Company Record in ServiceCenter. You can define only one template at a time for new operators. ServiceCenter merges login data from the LDAP directory with the template and creates an operator record for that user within ServiceCenter. This record is added to the ServiceCenter database and remains there until manually deleted.

To specify the use of a template in the System Wide Company Record:

- 1 In the system administrator's home menu, click the Utilities tab (Figure 8-1 on page 216).
- 2 Click **Administration**.
The administration menu is displayed (Figure 8-6 on page 228).
- 3 Click **System Wide Company Record**.
The System Wide Company Record for your company is displayed.

The screenshot shows the 'ServiceCenter - [Information]' application window. The title bar includes standard window controls and a menu bar with 'File', 'Edit', 'View', 'Format', 'Options', 'List Options', 'Window', and 'Help'. Below the menu bar is a toolbar with icons for 'OK', 'Cancel', 'Save', 'Find', and 'Fill'. The main content area is titled 'System Information Definition' and is divided into several tabs: 'Logon Info', 'Active Integrations', 'Menu Information', 'Password Standards', 'Password Composition', and 'Password Lifetime'. The 'Logon Info' tab is selected and contains two sub-sections: 'General Login Information' and 'Use User Lockout'. The 'General Login Information' section has fields for 'Max Logins Per User' (value: 2) and 'Attempts Per Login Session' (value: 5). The 'Use User Lockout' section has a checkbox for 'Use User Lockout' (unchecked), a field for 'Attempts Until Lockout' (value: 10), a radio button for 'Complete Lockout' (selected), and a field for 'Lock Out Time Period' (value: 00:10:00). To the right of these fields is an 'Account Expiration' section with fields for 'Inactivation Interval' and 'Expiration interval'. The 'Company Information' section at the top of the form has fields for 'Company Name' (Peregrine Systems Inc.), 'Address' (3611 Valley Centre Drive), 'City' (San Diego), 'State' (CA), and 'Zip' (92130). The status bar at the bottom shows 'Ready' and 'insert info.company.g [S]'.

Figure 8-9: System Wide Company Record—defining an LDAP template

- 4 Enter the name of your template in the **Operator Template** field.
In this example, we will enter **NEWUSER**.
- 5 Click **Save**.
The following message is displayed: *Information record updated.*

Defining a Default System Record

A default system record (*SYSDEFAULTS) functions much like a template, in that it provides ServiceCenter-specific information for new users authenticated by an LDAP server. When the new user attempts to log on to a system in which a *SYSDEFAULTS record has been defined, a permanent operator record is created using LDAP data and values from the *SYSDEFAULTS record.

Note: Once added to the ServiceCenter database, operator records created in this fashion are independent of *SYSDEFAULTS and cannot be altered by updates to *SYSDEFAULTS. However, updates to any field mapped to LDAP in an operator record created from a *SYSDEFAULTS record causes the data in the LDAP directory to be updated.

For procedures on administrative access to the LDAP directory, refer to the documentation for your LDAP server.

When a new user logs in to ServiceCenter, the system looks for a template definition in the System Wide Company Record. If no template is defined in the System Wide Company Record, the system uses the *SYSDEFAULTS record from the *operator* file. If both a template and a *SYSDEFAULTS record are defined, the system selects the template from which to create an operator record. If no template or *SYSDEFAULTS record is defined, login is denied.

Note: A *SYSDEFAULTS record can be used for any ServiceCenter file mapped to an LDAP server.

To create a *SYSDEFAULTS record in the *operator* file:

- 1 In the system administrator's home menu, click the Utilities tab (Figure 8-1 on page 216).
- 2 Click **Administration** (Figure 8-6 on page 228).
The Administration menu is displayed.
- 3 Click **Operators**.
A blank operator record search form is displayed (Figure 8-7 on page 229).
- 4 Click **Search** or press **Enter**.
- 5 Select an operator from the record list whose capabilities most closely match those you want to apply to your default system record.

- 6 Replace the name in the Login Name field with *SYSDEFAULTS.

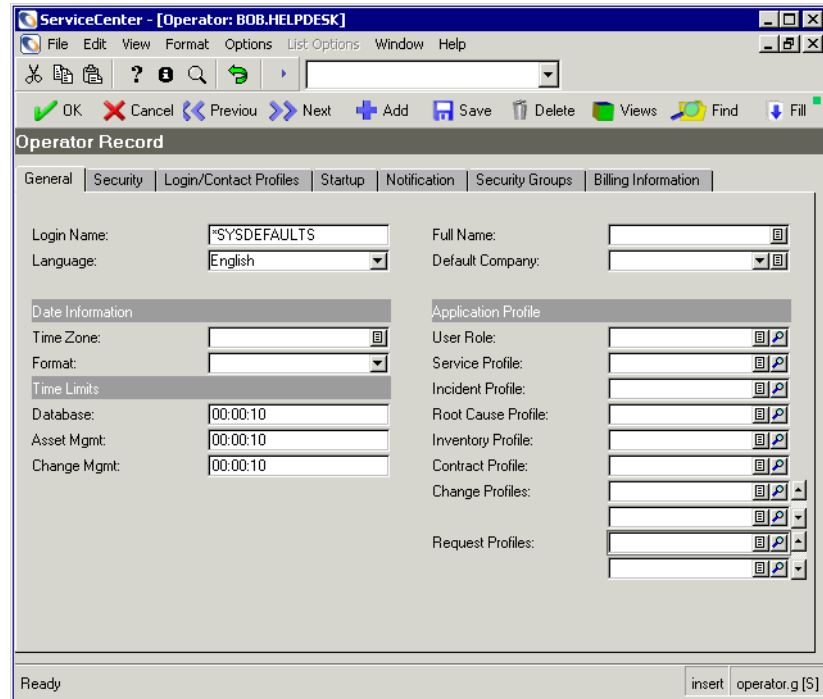


Figure 8-10: SYSDEFAULTS record in the operator file

- 7 Remove any information from fields that have been mapped to attributes in LDAP.
- 8 Edit ServiceCenter-specific fields if necessary.
- 9 Click **Add** to add your default system record to the database (Figure 8-10 on page 233).
- 10 Repeat this procedure for any other ServiceCenter files you have mapped to an LDAP directory.

For example, create a *SYSDEFAULTS record for the contacts file and generate permanent contact records from LDAP login information.

Note: *SYSDEFAULTS records are not visible when LDAP is selected as your primary data source. To view this record or the operator records of users not appearing in the LDAP directory, you must clear the **LDAP is Primary Data Source** check box in the File/Field Level Specifications form (Figure 8-4 on page 220). Reselect LDAP as your primary data source to view operator records of those users appearing in the LDAP directory. It is not necessary to log out when toggling between data sources.

LDAP Administrator Capabilities

The LDAP server must have an administrator to maintain the data contained in the directory. Records must be added, deleted, or edited as the information about the company's users changes. Administrative functions in the LDAP directory can be accessed in two ways:

- Tools provided with your LDAP server
- Capabilities defined in the ServiceCenter operator record

LDAP Tools

Refer to the documentation for your LDAP server for procedures for administrative access to the LDAP directory.

Note: Peregrine Systems recommends using the tools provided with your LDAP server for maintaining the LDAP directory rather than using the ServiceCenter operator record for access.

Operator Record

To grant viewing privileges of an operator record, modify the `ou` command in the base directory path code of the `ldapauthbase` field in the operator record (see Figure 8-11 on page 235).

For example, you might want to make an administrator operator record available for viewing along with the other operator records. To do this, change the base directory path of an administrator's base directory code in the `ldapauthbase` field of the operator record from `ou=People,o=peregrine.com` to `ou=Administrators,o=peregrine.com`.

The administrator's operator record is then available for viewing along with the other operator records. Likewise, if you no longer want an operator record to be available for viewing to those without LDAP administrative privileges, change the ou command from `ou=Administrators,o=peregrine.com` to `ou=People,o=peregrine.com`.

The value in the `ldapauthbase` field of the operator record overrides the default base directory information provided in the `sldapconfig` file. When the field is changed, the new setting defines a different branch of the LDAP directory structure, allowing you to access the newly-defined administrator operator record along with other operator records.

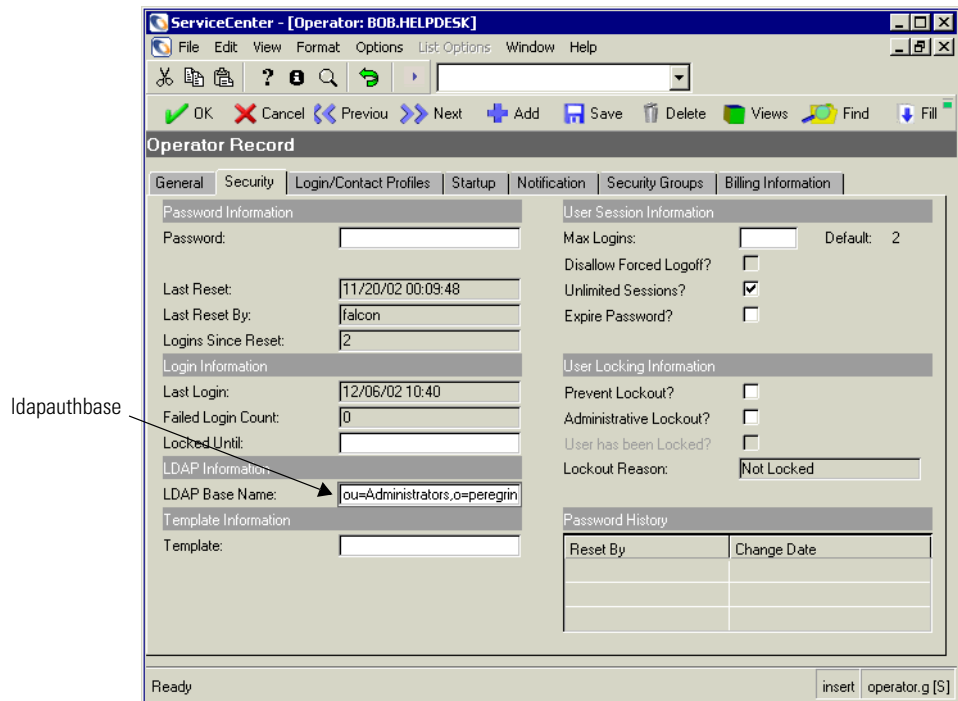


Figure 8-11: Operator record with a base directory defined

Manage LDAP Connection with sc.ini File Parameters

The LDAP interface employs two initialization (*sc.ini*) file parameters to manage the connection to your LDAP servers.

Parameter	Description
<code>ldapservers</code>	Defines alternate servers in case the primary server is inaccessible.
<code>ldapdisable:1</code>	Disables the LDAP login authentication procedure.

Defining Alternate Servers

If you have selected LDAP for user authentication by mapping the `operator` file, Peregrine Systems recommends that you provide a backup LDAP server. Should your primary LDAP server be unavailable for any reason (server failure or incorrect host information), you will be prevented from logging on to ServiceCenter. To avoid being locked out of your system, you can define alternate servers in your ServiceCenter initialization (*sc.ini*) file.

Note: If an LDAP server is not available or if the LDAP server is available but the user is not defined in LDAP, then ServiceCenter will automatically revert to using the data stored in the ServiceCenter `operator` file. If passwords have been stored in this file, then logins will still be allowed.

An initialization parameter called `ldapservers` can be added to the *sc.ini* file to define any number of alternate LDAP servers. If ServiceCenter cannot access the primary server defined in the `sldapconfig` file, the system will attempt to connect to each one of the alternate servers in sequence until a connection is made.

The syntax for this parameter is as follows:

```
ldapservers:host1:port,host2:port,host3:port...
```

The port number is optional and, if not provided, will default to the LDAP default value of 389 for unsecure connections and 636 for secure connections.

Example

```
ldapservers:titan:389,europa:389,phobos:389
```


To add the `ldapservice` parameter to the `sc.ini` file:

- 1 Open the RUN directory of your ServiceCenter 3 or later release installation.
- 2 Double-click the `sc.ini` file.
- 3 If prompted to select an application in which to open the file, select **Notepad**.
- 4 Type `ldapservice:<host1:port,host2:port...>` on the last line of the `sc.ini` file.

Example:

```
path:C:\Program Files\ServiceCenter\DATA
shared_memory:48000000
ntservice:ServiceCenter
log:C:\Program Files\ServiceCenter\sc.log
bitmap_path:C:\Program Files\ServiceCenter\BITMAPS
system:12670
auth:W6478H90 RT8692L0 34586RHY 3UG76D6K
scauto:12690
ldapservice:titan:389,europa:389,phobos:389
```

- 5 From the **File** menu, choose **Save**.
- 6 Close the record.

Disabling LDAP

If LDAP is being used for authentication, and no LDAP server is available, logins to ServiceCenter might not be possible. The `ldapdisable:1` initialization parameter bypasses the LDAP server and forces the system to use login data from the ServiceCenter database instead. When `ldapdisable:1` is added to the `sc.ini` file, any user with an operator record in ServiceCenter whose password is valid will be able to log on.

Note: If an LDAP server is not available or if the LDAP server is available but the user is not defined in LDAP, then ServiceCenter will automatically revert to using the data stored in the ServiceCenter **operator** file. If passwords have been stored in this file, then logins will still be allowed.

To disable LDAP:

- 1 Open the RUN directory of your ServiceCenter 3 or later release installation.
- 2 Double-click the *sc.ini* file.
- 3 If prompted to select an application in which to open the file, select Notepad.
- 4 Add `ldapdisable:1` to the bottom of the *sc.ini* file.

Example:

```
path:C:\Program Files\ServiceCenter\DATA
shared_memory:48000000
ntservice:ServiceCenter
log:C:\Program Files\ServiceCenter\sc.log
bitmap_path:C:\Program Files\ServiceCenter\BITMAPS
system:12670
auth:W6478H90 RT8692L0 34586RHY 3UG76D6K
scauto:12690
ldapdisable:1
```

- 5 From the **File** menu, choose **Save**.
- 6 Close the record.

9 Load Manager

CHAPTER

Overview

The ServiceCenter Load Manager information in this chapter has been designed to aid ServiceCenter Administrators who are responsible for setting up multiple servers and redirecting servers that have reached capacity. With the Load Manager setup, servers can be added to a complex ServiceCenter system to meet the demand for CPU and memory usage. These servers can run on multiple platforms, such as, Unix, Windows NT/2000/XP, Linux, and OS/390. The Load Manager manages the resources used among the servers, such as monitoring users, licensing, and managing resource locking.

Topics in this chapter include:

- *What is Load Manager?* on page 241.
- *Why Use Load Manager?* on page 242.
- *Installing Load Manager* on page 242.
- *Configuring Your System* on page 243.
- *Using Load Manager* on page 246.
- *ServiceCenter Error Messages* on page 247.

Requirements

Load Manager requires that the ServiceCenter database be completely mapped to a relational database management system (RDBMS). When ServiceCenter data is mapped to an RDBMS, each field in each file becomes a column in an RDBMS table. Each ServiceCenter file is mapped to one or more tables in the RDBMS, and each record in the ServiceCenter file is converted to one or more rows in the corresponding RDBMS tables. For information on data mapping, see the *ServiceCenter Database Management and Administration Guide*.

Prerequisites

Load Manager executes as an *scenter* executable option in the runtime environment on the normal ServiceCenter system. The runtime environment is the engine of ServiceCenter that compiles and runs RAD applications on your server or client platform and interacts with your server or client's operating system. To learn more about the runtime environment (RTE), see the *ServiceCenter Upgrade Utility Implementation Guide*.

To utilize the Load Manager, ServiceCenter needs to be installed. Follow the instructions in the installation guide that pertains to your platform, such as Unix, Windows NT/2000/XP, Linux, and OS/390 systems.

Related Documentation

Related documentation includes:

- *System Tailoring*
- *System Administrator's Guide*
- *Database Management and Administration Guide*
- *RAD User's Guide*
- *Technical Reference*
- A series of installation guides specific to system platforms

What is Load Manager?

ServiceCenter's Load Manager allows multiple ServiceCenter server installations on multiple machines to function as one logical installation. ServiceCenter utilizes a system resource called shared memory. As its name implies, this is a common piece of memory that is shared among all ServiceCenter processes. ServiceCenter uses shared memory to hold common data objects (application, format, and database dictionary records) and system status information (such as, user names, login times, and idle time), and to control resource locking.

The use of shared memory requires that all ServiceCenter processes execute on the same machine. In a Load Manager environment, a designated ServiceCenter process acts as a conduit between shared memory and processes executing on another machine. The Load Manager and remote ServiceCenter processes communicate via TCP/IP. In Figure 9-1 on page 241, the Load Manager is executing on Machine A. In addition, Machines B, C, and D are hosting ServiceCenter processes. The processes on Machines B, C, and D communicate their shared memory requests to the Load Manager via TCP/IP requests. All the ServiceCenter processes on all the machines are running on the same application files mapped to an RDBMS.

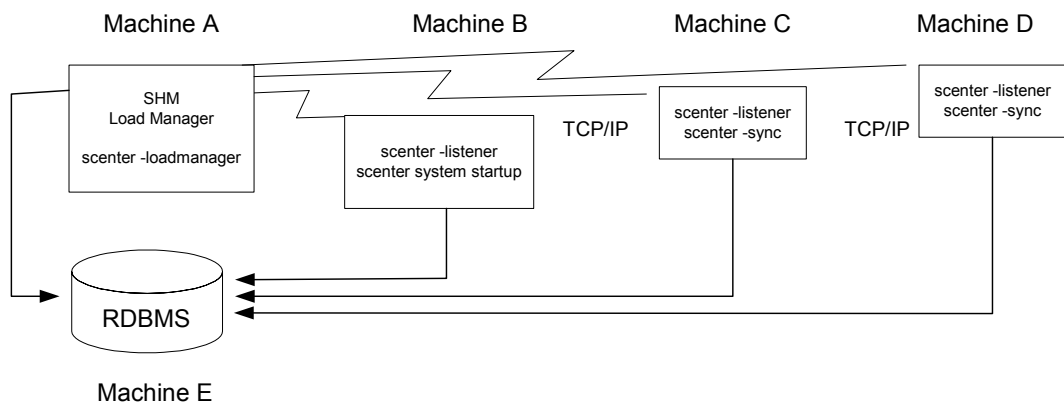


Figure 9-1: Load Manager managing resources amongst the client servers

Why Use Load Manager?

Load Manager allows System Administrators to improve system performance without adding new hardware resources. Once the hardware has reached its physical limits within a ServiceCenter environment and the performance of the system cannot be improved without additional hardware, Load Manager takes over by helping to manage the system resources among the servers.

In a typical ServiceCenter installation, the number of user connections and the performance of those connections are directly related to the speed of the processor and amount of RAM installed on the hardware. Once the hardware has reached its physical limits, the performance of the system cannot be improved without additional hardware resources.

Installing Load Manager

Several techniques are available to perform this setup; this document examines the most direct method.

Load Manager supports ServiceCenter servers running on Unix, Windows NT/2000/XP, Linux, and OS/390 systems. Load Manager does not require the servers to be the same platform type, but all platforms tied together have to support the same RDBMS.

Note: Peregrine Systems Professional Services should be contacted to assist with installation and configuration.

Once you have successfully converted your data to the RDBMS, you might want to run LFMAP on all P4 files to gain space in all of your P4 files. For information on the LFMAP procedure, see the *ServiceCenter Database Management and Administration Guide*.

Important: Every ServiceCenter system in the logical installation needs to have an identical set of P4 files, so once the conversion is done, you need to copy the complete set of P4 files to all of the other installations. These files will not be updated, but since they contain viable information that establishes the initial connection to the RDBMS, you need an identical copy of all the P4 files from the primary system (in Figure 9-1 on page 241, this would be Machine A) copied onto the data directory of each of the secondary systems (in Figure 9-1 on page 241, this would be Machines B, C, and D).

Configuring Your System

Configuring your system consists of the following tasks in order to successfully add servers to meet the demand for CPU space and memory usage.

- *Configuring the Primary System* on page 243.
- *Configuring Secondary Servers* on page 244.
- *Configuring Clients* on page 245.

Configuring the Primary System

- 1 Identify the primary ServiceCenter machine to be used as the Load Manager. Looking at Figure 9-1 on page 241, this would be Machine A. This machine must have a TCP host name that can be pinged. This name would be the name of the Windows NT/2000/XP.
- 2 Allocate an available listening TCP port number. Or, you can re-use the port you were already using as your regular TCP listener. In this example, assume that port number 12685 is available (not allocated by another process on the machine). For specific instructions on allocating a TCP port number for OS/390, refer to the *ServiceCenter Installation Guide for OS/390*.

- 3 Modify the configuration file to start the ServiceCenter Load Manager process for that component of the system.
 - For OS/390 servers, modify the *SCCFG* file.
 - For all other server types, modify the *sc.cfg* files.

Remove all commands in this file and replace them with:

```
scenter -listener:12685 -loadmanager
```

The Load Manager will accept connections on port 12685.

Note: Do not start any other process/listener in this *sc.cfg*, as Load Manager needs to run alone in this dedicated ServiceCenter system.

- 4 Since the Load Manager is responsible for maintaining the license information, its *sc.ini* file should contain the *auth:* parameter and, optionally, the *namedusersfile:* parameter.

Note: None of the other ServiceCenter installations need an *auth:* or *namedusersfile:* parameter.

Revise the *sc.ini* file on the Load Manager server to verify the authorization code and named user file for the Load Manager.

Configuring Secondary Servers

Set parameters as follows, so that the ServiceCenter processes can be run on other machines, allowing them to access the Load Manager.

- For OS/390 servers, modify the *PARMS* file.
- For all other machine types, modify the *sc.ini* files on the machines that will be accessing the Load Manager.

- 1 Go to the ServiceCenter Programs file directory.
- 2 Edit the *sc.ini* file with any text editor.

For example: On Windows NT, go to **Start>Programs>ServiceCenter>Init file** to open the *sc.ini* file with Notepad.

The *sc.ini* file is opened in Notepad.

- 3 Add the following parameter in the *sc.ini* file:
 `loadmanagerserver:<host>.12685`
 This will allow the ServiceCenter processes on other machines to contact the Load Manager.
- 4 Verify SQL parameters and remove IR Expert parameters.
- 5 Edit the *sc.cfg* file by removing `system.startup` in all secondary systems but one (in our sample, Figure 9-1 on page 241, either Machines B, C, or D). Add `scenter -sync` to all others.
- 6 From the File menu, choose **Save**. Exit Notepad.
- 7 Stop and restart the ServiceCenter servers.

Configuring Clients

- ▶ To set up the clients, change their `express:` or `system:` parameter (for example, `express:<host>.12685`).

The Load Manager will accept client connections and automatically re-routes them to the next available listener.

Note: If you have assigned Load Manager the old TCP listener's port number, you do not have to change any clients' parameters.

Using Load Manager

Load Manager allows you to establish multiple application servers with a single application database. An SQL (Structured Query Language) database, which is a standard interactive and programming language for getting data to and from an RDBMS, is required if multiple systems are going to be tied together through the ServiceCenter Load Manager. When a client connects to another server via the ServiceCenter Load Manager, Load Manager connects that client to a system that has the most spare capacity at time of connection. Once the client is connected to a system, the client remains with that system for the entire session. Figure 9-2 on page 246 illustrates how ServiceCenter Load Manager allows establishing multiple application servers with a single application database.

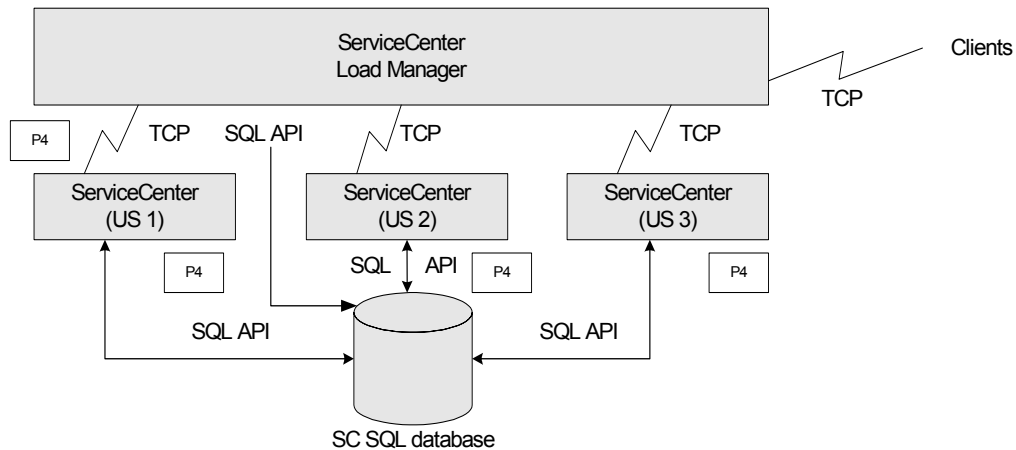


Figure 9-2: Load Manager establishing multiple application servers with a single application database

The SQL controls all locking across the systems and keeps track of the availability and capacity of each system.

ServiceCenter Error Messages

Error messages within the *sc.log* indicate that the process could not establish a connection to the Load Manager. Either the parameter does not specify the correct host name and port of the Load Manager process, or the Load Manager process has not been started. The following error messages can be found in the *sc.log*.

Error messages found in the *sc.log*

Error Message(s)	Description and Resolution
Error connecting to server, request timed out, errno (#####). -or- Could not connect to the Load Manager specified on the <code>loadmanagerserver:</code> parameter. Verify the parameter is correct and that the Load Manager process has been started.	The process could not establish a connection to the Load Manager. Either the parameter does not specify the correct hostname and port of the Load Manager process, or the Load Manager process has not been started. The process will continually attempt to connect to the Load Manager until a successful connection has been established. If you start the Load Manager process, the hanging processes will eventually connect.

10 Report Writer

CHAPTER

Overview

This chapter introduces Report Writer.

The following topics are discussed.

- *How to Access Report Writer* on page 250.
- *How to Create a Simple Report Listing* on page 252.
- *How to Query for Reports* on page 266.
- *How to Use Query Formats with Reports* on page 268.
- *How to Use Initial Formats with Reports* on page 275.
- *How to Use Page Breaks and Report Totals* on page 278.
- *How to Use Format Control with Reports* on page 292.
- *How to Run Reports in a Foreground* on page 311.
- *How to Schedule Reports for Background Processing* on page 312.
- *Online Viewing of Reports* on page 320.
- *How to View Report Schedule Records* on page 325.
- *How to Copy/Rename Reports* on page 326.
- *How to Delete Reports* on page 327.
- *How to Use Stacked Queries for Sorting* on page 328.

Understanding Report Writer

Report Writer allows you to define selection, totaling, and report format criteria to create listings from one or more database files, using one database file as the primary file. The reports may be as simple or complex as you wish, and can be modified at any time to add complexity or clarify details. The Report Writer allows for complete user-definition.

How to Access Report Writer

- 1 Enter `rw` on any menu command line and press **Enter**.

The *report.prompt.rw* form is displayed.

- Search for an existing report or primary file name.

Press **Enter** or click **Search** to perform a *true* query without entering any values in the blank form and bring up a QBE list of existing report records. From the QBE list select a report to edit, delete, or run.

-or-

Enter an existing report or primary file name to be queried and press **Enter** or click **Search**. From the QBE list select a report to edit, delete, or run.

- Create a new report.

Press **F1** or click **New** to begin the process of creating a new report. The *report.maint* form is displayed. For detailed information on creating a new report, see *How to Create a Simple Report Listing* on page 252 for detailed information.

- 2 Click **Back** or press **F3** to exit Report Writer.



Command Keys

Listed below are the command keys available from the *report.maint* form.

Command Key	Description
F3 = OK	Ends the <i>edit</i> mode, saves the newly-created report or any modifications made to an existing report, and returns to the <i>format report</i> .
F4 = Cancel	Cancels out of creating a new report and does not save the information. Returns back to the <i>report.prompt.rw</i> format.
F5 = Run	Begins the process of running a report in the foreground or background. A newly-created report is saved and modifications made to an existing report are also saved.

After selecting **Edit** or when creating a new report, the following options are displayed:

Option	Description
Forms Designer	Allows you to access Forms Designer to alter existing or create new forms.
Set Query	Begins the process of defining Primary File Queries. Refer to <i>How to Use Query Formats with Reports</i> on page 268 for more information.
Totals	Opens a window at the bottom of the current screen, which allows for the definition of Total Parameters. Refer to <i>How to Use Page Breaks and Report Totals</i> on page 278 for more information.
Copy/Rename	Begins the process of copying or renaming a report. Refer to <i>How to Copy/Rename Reports</i> on page 326 for more information.

How to Create a Simple Report Listing

This section demonstrates how to create a simple report listing:

- 1 From the System Administrator's menu, enter `rw` on any menu command line and press **Enter**.

The `report.prompt.rw` form is displayed.



- 2 Enter the name of the new report in the **Report** field of the `report.prompt.rw` form, and then press **F1** or click the **Create New Report** button.

For this example, enter `problem.category` as the name of the report in the **Report** field.

Note: Choose a unique name for the report and enter it here. Any naming scheme can be used. It is recommended that you do not use special characters other than “.” or blank spaces. Usually the first node up to the first “.” is the same as the primary file of the report.

For this example, if you enter `problem category` without . (period), the primary file is not created. If you enter `problem.category`, the primary file is created.

The system checks to see if the report format already exists.

- If the report format already exists, the following message is displayed in the status bar: *The report name <report name> is already used as a format. Please re-enter.*

You can either make changes to the existing report, or try entering a different name to create a new report.

- If the report format does not exist, a `report.maint` form is displayed (Figure 10-1).

The **Header** and **Details** fields in the Formats structure of the form are pre-filled with the name of the report and an added extension of *.header* and *.detail*, respectively.

The screenshot shows the 'Report Maintenance' dialog box in ServiceCenter. The 'Basic' section has the following values: Name: problem.category.report; Title: OPEN PROBLEM ANALYSIS REPORT - CATEG; Application: (empty); Initial Form: (empty); Primary File: problem; Query Form: (empty); Author: (empty); Editor: falcon; Date: 04/28/03 12:21:25. The 'Formats' section has: Header: problem.category.header; Subhdr: (empty); Detail: problem.category.detail; Print master Header: Yes; Print Subheader: (empty); Subheader Page Break: (empty). The 'Select' section has: Primary Query: status="closed" and flag=true; Stacked Queries: (empty); Sort Sequence: category; Lines Per Page: 66; Width (80/132): 80. The status bar at the bottom indicates 'Ready' and 'Response 0.111 draw 0.100 | insert | report.maint.g [S]'.

Figure 10-1: Creating a new report format

- 3 Using the Tab key to move the cursor, fill in the fields. For information on field descriptions, see *Field Descriptions for the report.maint Form* on page 255 and *Field Descriptions of the report.totals Form* on page 280.

Note: Be sure to fill in the *required* fields.

- 4 When you are finished adding information to the necessary fields, click OK.

You will receive a message in the status bar that states: *Record added to the report file* (see Figure 10-2 on page 254).

The screenshot shows the 'Report Maintenance' dialog box in ServiceCenter. The 'Basic' section contains the following fields: Name (problem.category.report), Title (OPEN PROBLEM ANALYSIS REPORT - CATEG), Application, Initial Form, Primary File (problem), and Query Form. The 'Author' field is empty, 'Editor' is 'falcon', and the date is '04/29/03 07:31:17'. The 'Save Report' checkbox is checked. The 'Formats' section includes 'Header' (problem.category.header), 'Subhdr', and 'Detail' (problem.category.detail). The 'Print master Header' is set to 'Yes', 'Print Subheader' is set to 'No', and 'Subheader Page Break' is empty. The 'Select' section includes 'Primary Query' (status~="closed" and flag=true), 'Stacked Queries', 'Sort Sequence' (category), 'Lines Per Page' (66), and 'Width (80/132)' (80). The status bar at the bottom shows 'Ready' and 'Response 0.160 draw 0.100 insert report.maint.g [S]'.

Figure 10-2: Report Maintenance *report form*

Note: The maximum width of a report is 132 characters. If the formats that are defined for a report exceed this limit, they are generated with that width to the spool file. However, when passed to JES (OS/390) or the printer (Unix), they are truncated to 132 characters (or the maximum width as defined in the config file for the printer that is used to print the report).

Note: To set the width of a report, complete the **Page Width** field of the Client Printing structure in the user's Operator record. For more details on completing the Operator record, see the *Client Printing* section in *System Administrator's Guide, Volume 1*.

After the necessary fields have been defined, you must create the *header* and *detail* formats that are used by the report (if they have not already been created). For more information, see *Creating and Editing Report Formats* on page 264.

Field Descriptions for the *report.maint* Form

Field	Description
<i>Basic Structure</i>	
Name	Choose a unique name for the report and enter it here. Any naming scheme can be used; however, it is recommended that you do not use special characters other than “.” or blank spaces. Usually the first node up to the first “.” is the same as the primary file of the report. This field is required.
Title	Enter the report title that will print at the top of every page as part of the <i>master header</i> format.
Application	ServiceCenter application being used for this report.
Initial Form	Defines the name of the format to be used to obtain data from the operator at run time, which can then be used during report generation. Refer to <i>How to Use Initial Formats with Reports</i> on page 275 for more information.
Primary File	<p>Name of the primary ServiceCenter file that is used to select records, such as the probsummary file, to satisfy the query requirements of the report.</p> <p>This file is the logical file that uses the queries that are defined in the fields: Primary File, Query and/or Stacked Queries.</p> <p>If this field is left blank, it will default to being the same as the first node (up to the first “.”) of the Report Name after F1 is pressed to create a new report. If the report name does not contain a “.” then it defaults to blank when F1 is pressed to create a new report. The next displayed format (<i>report.maint</i>) must then be defined. This field is required.</p> <p>For this example, this field is defaulted to problem, because a primary file was not specified on the <i>report.prompt.rw</i> format. The report name does have at least one “.” (such as, <i>problem.category</i>).</p>

Field	Description
Query Form	<p>Defines the format to be used that will allow the operator to define via QBE searching techniques the primary file query to use to select records for the report. The QBE is translated to an SQL format and is stored by operator ID and report name.</p> <p>Any number of queries can be stored per report and they can then be accessed and used for a particular execution of a report. Refer to <i>How to Use Query Formats with Reports</i> on page 268 for more information.</p>
Author	This field is populated by ServiceCenter, which uses the operator's login.
Editor	This field is populated by ServiceCenter, using the name of the last person (operator) who updated the report.
Save Report	The default is checked (<i>true</i>) to automatically save the new report.
Summary Detail	<p>Select one of the following valid entries:</p> <p>Summary = summary only prints subtotals, summary totals, and grand totals.</p> <p>Detail = details are selected from the Primary File, Query, or Stacked Queries and are printed using the format defined in the Detail field in the Format structure. Any other total/subheader formats are also printed.</p> <p>This field defaults to Detail.</p>
Auto Manual	<p>Select one of the following valid entries:</p> <p>Auto = If a Report Type was not defined on the <i>report.prompt.rw</i> format in the previous step, it is defaulted to auto.</p> <p>If a report is defined as auto, the standard functions of the Report Writer application (Page Breaks, Totals, Format Control) can be used during report generation.</p> <p>Manual = If a report is defined as manual, user-defined RAD panels are required to perform all functions of the report (such as, report selection, calculations, and page breaks).</p>

Formats structure

Field	Description
Header	<p>The <i>header</i> format name is a user-defined format that is printed at every top-of-page break. This format will be printed in addition to the master header, and usually contains column headings that explain the fields in the detail section of the report.</p> <p>Note: If a <i>header</i> format is not defined for a report but Yes is specified for the Print master Header field and Detail is selected to print in the Summary/Detail field, both the <i>header</i> format and the <i>detail</i> format will print.</p> <p>Note: The maximum width of a report is 132 characters. If the header formats that are defined for a report exceed this limit, they are generated with that width to the spool file. However, when passed to JES (OS/390) or the printer (Unix), they are truncated to 132 characters (or the maximum width as defined in the config file for the printer that is used to print the report).</p> <p>Note: To set the width of a report, complete the Page Width field of the Client Printing structure in the user's Operator record. For details on completing the Operator record, see the <i>Client Printing</i> section in <i>System Administrator's Guide, Volume 1</i>.</p>
Subhdr	<p>Name of the user-defined format to print subheaders. The <i>subheader</i> format is used when a report is designed to print information from a file that immediately precedes detail information extracted from a secondary file. If subheaders are used, totals may be calculated only for the subheader records.</p> <p>Subheader reports require a Format Control record to be defined for the <i>detail</i> format of the report, which identifies the secondary file that is used to extract information for the <i>detail</i> format.</p>

Field	Description
Detail	<p>Name of a user-defined format, which is used for printing the detail information of each record extracted from the primary file or, if subheaders are used, it will print the detail information extracted from the secondary file.</p> <p>The field names referenced on this format must be field names from the primary or secondary file's database dictionary and/or variables that are defined through Format Control or Totals Parameters definitions.</p> <p>The naming standard for this field is the report name (problem.category) with the character string .detail appended (for example, problem.category.detail). However, this naming convention does not have to be followed. As a default, the detail format of the report takes on the name of the newly created report with an added extension of <i>.detail</i>.</p> <p>The same detail format can be used on multiple reports, if desired. This field is required.</p>

Field	Description
Print master Header	<p>Valid entries are:</p> <p>Yes = Yes, which is the default. If Yes is selected, one of two formats (<i>report.master.hdr132</i> or <i>report.master.hdr80</i>, depending on the value specified in the Width field) is printed every time an auto report takes a top-of-page exit.</p> <p>Note: The formats <i>report.master.hdr132</i> and <i>reportmaster.hdr80</i> print the following information:</p> <ul style="list-style-type: none"> ■ Company title as defined in the System Wide Company Record in the info file. ■ Date/time the report started processing. ■ Report title. ■ Page number. ■ Operator ID of operator who generated report. ■ Selection criteria (if query is <i>true</i>, then it is reported as <i>all records</i>, otherwise the actual query is printed). ■ Sort fields. <p>No = when only the format specified in the field header format name is printed during top-of-page exits. If No is specified and the field header format name is blank, only the format specified in the field detail format name is printed. The fields normally printed on a master header may be incorporated into the user-defined <i>header</i> format.</p> <p>Note: To set the width of a report, complete the Page Width field of the Client Printing structure in the user's Operator record. For more details on completing the Operator record, see the <i>Client Printing</i> section in <i>System Administrator's Guide, Volume 1</i>.</p>
Print Subheader	<p>Yes = when a subheader is defined and Yes is selected, the subheader is printed on the report. The default used is Yes (see Figure 10-1 on page 253).</p> <p>No = when a subheader is defined and No is selected, the subheader is not printed on the report.</p>
Subheader Page Break	<p>Valid entries are:</p> <p>Yes = when a subheader is defined and Yes is selected, it is printed at top-of-page breaks. The default used is Yes.</p> <p>No = when a subheader is defined and No is selected, the subheader format is printed once for every detail record.</p> <p>Note: A subheader format has to be defined in order for this selection to have any significance on the report. For this example, a subheader is not used.</p>

Field	Description
<i>Select structure</i>	
Primary Query	<p>Defines either the query that is used to extract records from the primary file during processing of the report or if a query is used to extract records from a sortwork file (if one or more <i>stacked queries</i> are defined).</p> <p>At user discretion, the query can only be defined for a particular report, or can be used in conjunction with stacked queries. This field is required.</p> <p>Note: For performance considerations, you must have a good understanding of the relationship difference between the primary file query and stacked queries. Refer to <i>How to Use Stacked Queries for Sorting</i> on page 328 for more information.</p>
Stacked Queries	<p>An array (list) field that can optionally contain multiple queries that are used to extract records from the primary file. Refer to <i>How to Use Stacked Queries for Sorting</i> on page 328 for more information.</p>
Sort Sequence	<p>An array (list) field used to specify the sort sequence of the detail records extracted, using the primary file query or stacked queries.</p> <p>If a primary file query is defined and no stacked queries are defined, the sort fields <i>must</i> be defined as <i>keys</i> in the primary file's dbdict.</p> <p>If at least one stacked query is defined, the sort fields do not have to be defined as keys, but they <i>must</i> be defined as <i>fields</i> in the primary file's dbdict. Refer to <i>How to Use Stacked Queries for Sorting</i> on page 328 for more information.</p> <p>For this example, the report does not use stacked queries, therefore the <i>sort sequence</i> must be defined as a key in the primary filename (probsummary) dbdict. The sort sequence used for the demonstration report is <i>assignment</i>.</p>
Lines Per Page	<p>Defines how many lines per page the report allows, including the lines of the <i>master header</i> format, <i>header</i> format, <i>subheader</i> format, and <i>detail</i> format. The default is 66 lines per page.</p> <p>Users who direct their listings to laser printers may want to consider changing this value to 60, because of the 1/2 inch margin requirement for the bottom and top of a page. Otherwise, the page may overflow. This field is required.</p>

Field	Description
Width (80/132)	<p>Control which master header format is printed during top-of-page breaks.</p> <p>If the value is 80 or less, the master header format <i>report.master.hdr80</i> is used.</p> <p>If the value is greater than 80, the master header format <i>report.master.hdr132</i> is used. The default used is 80.</p> <p>This field does not perform any type of truncation on any formats used in the report. The value entered is not validated. This field is required.</p> <p>Note: To set the width of a report, complete the Page Width field of the Client Printing structure in the user's Operator record. For more details on completing the Operator record, see the <i>Client Printing</i> section in <i>System Administrator's Guide, Volume 1</i>.</p> <p>For this example, report width of 80 is used.</p>
<i>Subtotals structure</i>	
Format Name	<p>The name of the format that is used to print accumulated statistics at a user-defined control/condition break. More than one statistic can be accumulated and printed for each control break.</p> <p>To add totals to the Subtotals structure, select Options > Totals.</p>
Field/Variable Names to Total	<p>A field name that is defined in the primary file's dbdict, or a variable that is defined in an earlier subtotals definition within the same report, or a variable that is defined in a Format Control definition for the <i>detail</i> format of the report.</p> <p>Report Writer will use the value of this field to perform the indicated calculation each time a <i>detail</i> record is processed.</p>
Calc	<p>One of four choices that indicates which type of calculation is performed using the value of the field or variable defined in field/variable names to total. The indicated calculation is performed for every detail record processed by the report regardless of whether or not the detail format is printed.</p>

Field	Description
SUM	<p>Accumulates the values of a specified field for each detail record extracted by the Primary File Query or Stacked Queries.</p> <p>If a variable is specified for the Field/Variable Name To Total field, then the value contained in that variable at the time each detail record is printed (or would have been printed if d was specified for summary or detail on the <i>report.maint</i> format) is accumulated in the variable specified in the Variable Names On Format field.</p> <p>The accumulation continues until some control/condition break is reached, at which time the accumulated total is stored in the variable defined in the Variable Names On Format field, the <i>subtotal format</i> is printed, and the variable used to accumulate the totals is reset to zero.</p>
AVG	<p>Accumulates the values of a specified field for each detail record extracted by the Primary File Query or Stacked Queries, and the total number of detail records read that produced the accumulated total.</p> <p>If a variable is specified for the Field/Variable Name To Total field, then the value contained in that variable at the time each detail record is printed (or would have been printed if d was specified for summary or detail on the <i>report.maint</i> format) is accumulated in the variable specified in the Variable Names On Format field.</p> <p>When a control/condition break is detected, the accumulated total is divided by the total number of records. This result is stored in the variable defined in the Variable Names On Format field.</p> <p>The control/condition break also causes the <i>subtotals format</i> to print, causing the variable used to accumulate the totals and the total number of records to be reset to zero</p>

Field	Description
MIN	<p>Examines the value of a specified field in each detail records and retains the lowest value encountered.</p> <p>If a variable is specified for the Field/Variable Name To Total field, then the value contained in that variable at the time each detail record is printed (or would have been printed if d was specified for summary or detail on the <i>report.maint</i> format) is accumulated in the variable specified in the Variable Names On Format field.</p> <p>When a control/condition break is detected, the value is stored in the variable defined in the Variable Names On Format field.</p> <p>The control/condition break also causes the <i>subtotals format</i> to print, causing the variable used to accumulate the totals and the total number of records to be reset to zero.</p>
MAX	<p>Examines the value of a specified field in each detail record and retains the highest value encountered.</p> <p>When a control/condition break is detected, the value is stored in the variable defined in the Variable Names On Format field.</p> <p>The control/condition break also causes the <i>subtotals format</i> to print, causing the variable used to accumulate the totals to be reset to zero.</p>
Variable Names on Format	<p>Defines the variable that contains the results of the specified calculation. This variable is normally printed via the <i>subtotals format</i>. However, this is not a requirement. The variable can also be used in subsequent calculations.</p>

Creating and Editing Report Formats

The following steps will help you to create and edit report formats directly from Report Writer.

- 1 From the System Administrator's menu, enter `rw` on any menu command line and press **Enter** or click **Search**.

The *report.prompt.rw* form is displayed.

- 2 Enter the name of the existing report in the **Report** field of the *report.prompt.rw* form, and then press **Enter** or click **Search**.

For this example, enter `problem.category` (the name of the newly created report discussed in *How to Create a Simple Report Listing* on page 252).

The *report* form is displayed (Figure 10-2 on page 254).

- 3 Click **Edit** to update the report and add any necessary information. The *report.maint* form is displayed (Figure 10-3). See the field descriptions discussed in the *Field Descriptions for the report.maint Form* on page 255.

ServiceCenter - [enter data and select function]

File Edit View Format Options List Options Window Help

F3 - OK F4 - Cancel F5 - Run

Report Maintenance

Basic

Name: problem.category.report Author: [] []

Title: OPEN PROBLEM ANALYSIS REPORT - CATEG Editor: falcon 04/29/03 07:31:17

Application: []

Initial Form: []

Primary File: problem

Query Form: []

Save Report

Summary Auto

Detail Manual

Formats

Header: problem.category.header Print master Header Yes

Subhdr: [] Print Subheader No

Detail: problem.category.detail Subheader Page Break []

Select

Primary Query: status~="closed" and flag=true

Stacked Queries: []

Sort Sequence: category Lines Per Page: 66

Width (80/132): 80

Ready Response 0.160 draw 0.100 insert report.maint.g [S]

Figure 10-3: Editing a report in the *report.maint* form

- 4 If the report format name has not been previously entered, select **Options>Totals>Forms Designer**. At the **enter format name** prompt, enter the *detail format* name.

For this example, enter `problem.category.detail` and press **Enter**.

Fill in the detail format, following normal Forms Designer procedures. Select **Options > Forms Designer**. For information on Forms Designer, see *ServiceCenter System Tailoring*.

Note: When filling in a *detail* format for a report, only input fields are defined. However, this is not a restriction. Label fields can also be defined, if needed.

5 Press F1 to add.

Note: Edit the field names to reflect their dbdict field names. To ensure that the input field names are correct, the format may be associated with the primary file of the report. As each input field is defined, check the field name to verify that the field does exist in the primary file.

6 Press F3 to close the edit field window.

7 Press F3 until you return to the *report.maint* format and save the newly-created format.

Note: Update the *header* format, if prompted to do so.

When you are ready to run this report, refer to *How to Run Reports in a Foreground* on page 311 and *How to Schedule Reports for Background Processing* on page 312 for complete instructions.

How to Query for Reports

1 From the System Administrator's menu, enter *rw* on any menu command line and press **Enter**.

The *report.prompt.rw* form is displayed.

2 Press **Enter** or click **Search** without entering any values to perform a *true* query and pull up a QBE list of existing report records.

-or-

Enter an existing report name to be queried and press **Enter** or click **Search**.

3 For this example, query for any report that has a primary file of *problem* or *probsummary*. Enter *probsummary* in the **Primary File** field of the *report.prompt.rw* form. Press **Enter**.

If the query entered selects more than one record, a QBE list of reports is displayed (Figure 10-4).

Note: Beginning with ServiceCenter 5.0, the primary file to query for problem records is *probsummary*. Use the *problem* file when the **Use Paging** field is checked in the **Incident Management Environment** record.

If the query entered produces only one record, that report record is displayed on the *report* screen.

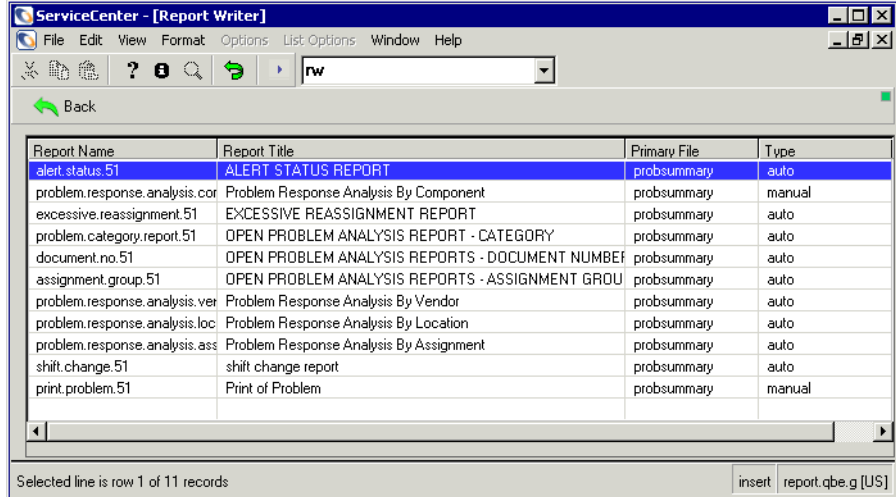


Figure 10-4: A QBE list of reports displays in response to the query

- 4 Scroll through the list. Place the cursor on any record within the QBE list and double-click on that record to select the record. The record is displayed on the *report* form (see Figure 10-2 on page 254).

The report can be edited, deleted, or scheduled to run. When you are ready to run this report, see *How to Run Reports in a Foreground* on page 311 and *How to Schedule Reports for Background Processing* on page 312.

- 5 Click **Back** or press **F3** to return to the main Report Writer menu (*report.prompt.rw* form).

How to Use Query Formats with Reports

Report Writer provides the option to define a set of pre-defined queries for a report, which can then be used as the *primary file query* when the report is initially scheduled to run. These queries are stored by report name and user ID, and can be accessed for viewing and updating.

Creating Primary File Queries

When creating primary file queries, you have two options:

- Enter the name of the form to be used when creating queries in the **Query Form** field.
- or-
- Leave the **Query Form** field blank, which will cause Report Writer to use the value in the **Primary File** field for the name of the format to be used in creating queries.

Note: The query form name must be valid in order to be used.

To access a report for viewing and updating when a record is selected:

- 1 Click **Edit** or press F2.

The *report.maint* form displays (Figure 10-3 on page 265), which allows you to edit the current report.

Note: In the OS/390 environment, use < > instead of ~ (tilde) for queries.

- 2 Click **OK** to save your changes.

A message displays in the status bar that states: *Record updated in the report file.*

-or-

Click **Run** to begin running the report.

A message is displayed in the status bar that states: *Record updated in the report file.*

The Run Report Exerciser (*report.prompt.exer*) form is opened and you can fill in specifics on this form to run the report.

To learn more about running reports, see [How to Run Reports in a Foreground](#) on page 311 and [How to Schedule Reports for Background Processing](#) on page 312.

To establish pre-defined queries for a report:

- 1 Access an already existing report. From the System Administrator's menu, enter `rw` on any menu command line and press **Enter**.

The *report.prompt.rw* form is displayed.

- 2 Press **Enter** or click **Search** to display a QBE list of existing report records.

-or-

Enter an existing report name to be queried.

For this example, enter `problem.category.report`. The *report* form is displayed (Figure 10-2 on page 254).

- 3 Click **Edit** or press **F2** (edit report) to access the edit functions of Report Writer.

Note: In the OS/390 environment, use `< >` instead of `~` (tilde) for queries.

- 4 Select **Options > Set Query** to access the query format and begin the process of setting queries for the report (Figure 10-5 on page 269). When working with this format, use the command keys described in the following section, *Command Keys* on page 270.

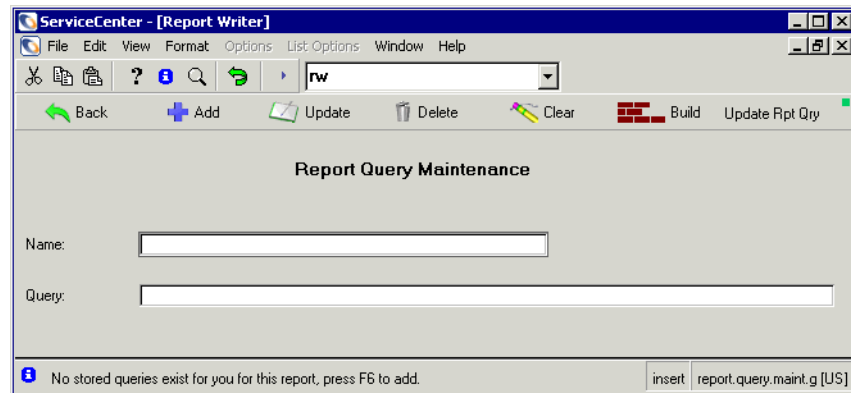


Figure 10-5: Report Query Maintenance

Command Keys

Listed below are the Command Keys available from the *report.query.maint* form.

Command Key	Description
F1 = add	Adds the currently displayed query and name to the query database. The name field must be unique per user and report.
F2 = update	Applies changes made to the query and/or name.
F3 = back	Returns to the previously displayed format.
F4 = delete	Deletes the currently displayed query record.
F5 = clear	Clears all displayed fields.
F6 = build qry	Displays the format defined in the Query Form . If no format is specified, the format that is the same name as the <i>primary filename</i> of the report (if any) is displayed.
F8 = upd rpt qry	Copies the currently displayed query and overlays the <i>primary file query</i> of the report currently being edited.

For this example, select the *problem.category.report* record.

The Report Query Maintenance (*report.query.maint*) form displays.

- If report queries were already defined for this report, a QBE list of those queries is displayed (Figure 10-6). From this list, you can select a query for maintenance.

- If there are no report queries defined for this report, a message is displayed in the status bar that states: *No stored queries exist for you for this report. Press F6 to add.*

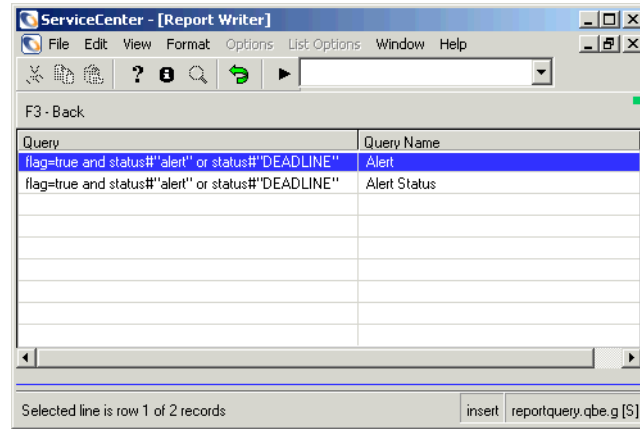


Figure 10-6: Report Writer queries - *reportquery* form

- 5 Select a record from this list and click **Edit** or press **F2** to make changes.

When you finish editing the record, the following message is displayed in the status bar: *Report query record has been updated.*

-or-

Enter the name of the query in the **Name** field and the query into the **Query** field of the *report.query.maint* form.

For this example, enter the following information into the *report.query.maint* form.

- a Enter **Alert Status** in the **Name** field.
- b Enter the following in the **Query** field:
status#"alert" or status#"DEADLINE"

Note: In the OS/390 environment, use **< >** instead of **~** (tilde).



- 6 Click **Add** to add the stored query.

A message is displayed in the status bar that states: *Report query record has been added.*



- 7 To begin the process of entering a report query, press F6 or click **Build**.

The *problem* form is displayed (Figure 10-7). The format that you specify here when you enter QBE arguments will display in the *Query* field of the *report.query.maint* form (Figure 10-5 on page 269).

Figure 10-7: Entering a report query

- 8 Enter data in the fields that will satisfy your QBE search. The more fields you fill in, the narrower your QBE search will be. The fields are described at the end of these steps in *Field Descriptions for the problem Form* on page 274. For this example, enter the field value `=systems` in the *Category* field.
- 9 Press F6 (query window) to open the query window at the bottom of the screen. This window will display the SQL translation of the query entered on the screen, which can be modified.

-or-

Press **Enter** to translate the query into SQL and the *report.query.maint* format is redisplayed with that SQL expression in the **Query** field.

For this example, press F6.

Note: This routine does not automatically insert the *flag=true* portion of queries, which are normally associated with queries used against Incident Management and Change Management.

The user is responsible for making the necessary modification to the translated SQL to accommodate these or any other special query needs.

For this example, press **Enter**. The following *report.query.maint* form is opened (Figure 10-8).

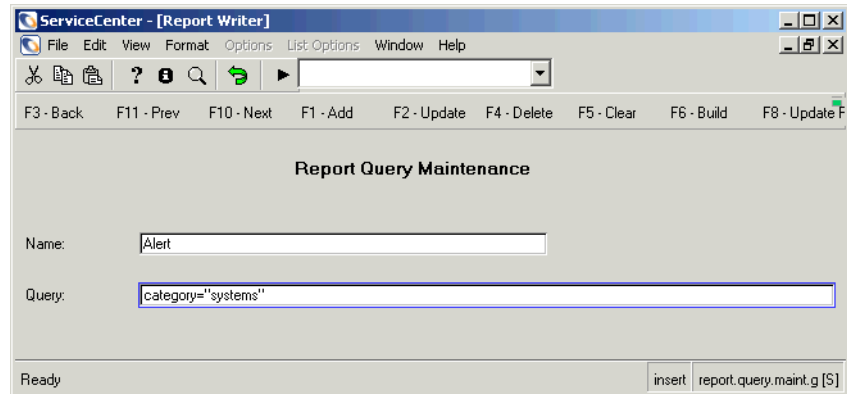


Figure 10-8: Completed Report Query Maintenance build

- 10 If the query is correct, press **Enter**.

-or-

If the query shown in the *report.query.maint* form is not the desired query, make changes as needed and click **Update**.

For this example, the desired query is shown, which will select all records from the *probsummary* file that have a *category* value of *systems*.

The translated SQL is displayed in the **Query** field.

- 11 Define a unique query name for the query.

For this example, enter *systems*.

- 12 Optionally, you can modify the SQL expression shown in the **Query** field.

For this example, add **flag=true** to the existing expression in the **Query** field.

- 13 Press **F1** or click **Add** to add the new query record. You will receive a message displayed at the bottom of the screen, stating: *Report query has been added*.

- 14 Click **Back** twice to get back to the *report.maint* form.



- 15 Click **Run** to run the newly created report query.

A QBE list of one or more queries is displayed.

- 16 Double-click the query name of one of the query records listed.

- 17 The Run Report Exerciser (*report.prompt.exer*) form is now displayed, from which you can schedule the report to be run in the background or foreground.

For more information on processing reports, see *How to Schedule Reports for Background Processing* on page 312 and *How to Run Reports in a Foreground* on page 311.

While in the *report.prompt.exer* form, the query that was defined in the previous steps now is displayed in the **Primary File Query** field. However, the original primary file query defined in the report record itself is *not* affected.

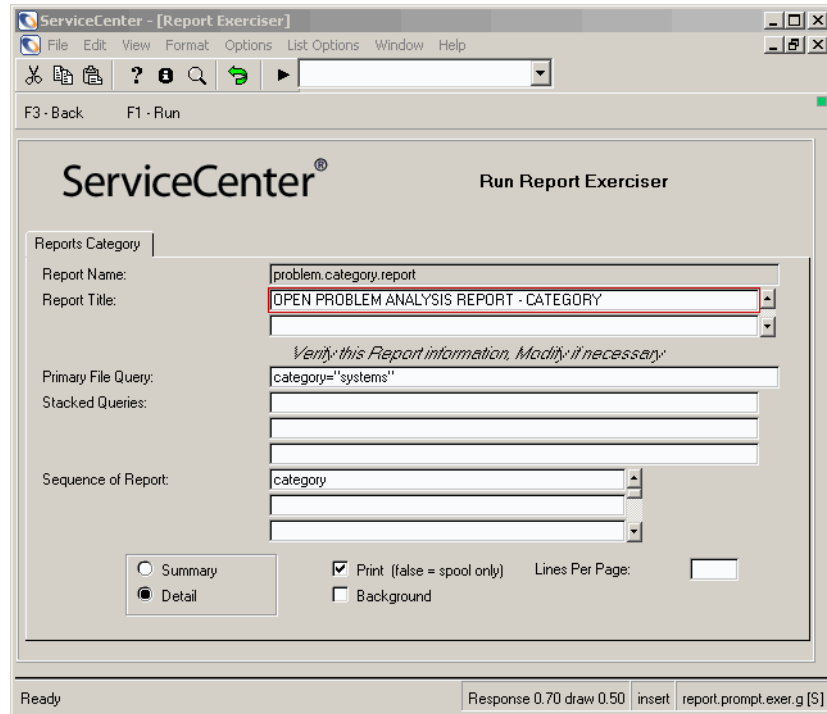


Figure 10-9: Processing reports using the Run Report Exerciser

Field Descriptions for the *problem* Form

The field descriptions in the *problem* form describe how you will set up your data to satisfy your QBE search. The Header Information structure delineates how each record is printed within the query.

Field	Description
<i>Header Information structure</i>	
Incident No	Incident number of queried records.
Status	Status of the call reports to be used for this query.

Field	Description
Category	Category by which an incident has been classified.
Assigned To	Qualified person to which the incident has been assigned.
Priority	Priority (1 through 4) assigned to the call report.
Change No	Unique change record number.
Page	Page number of the printed report.
Reassign Count	Query incident tickets that have been reassigned a specified number of times.
Opened	Date the incident ticket was opened.
Alert	If an alert has been generated for this record, the alert status that was generated.
Updated	Date the record was last updated.
Closed	Date the record was closed.
Logical Name	Logical file name for this record.
Location	Query a particular company location to be selected, or leave blank to include all company locations.
Contact Name	Query for a particular contact name.
Service Contact	Query for a particular service contact -- a contact outside of business hours.

How to Use Initial Formats with Reports

An *initial format* is used to allow the user to enter data, which will assist in developing the query for the report.

For example: If the report requires a date/time range, it is simpler to provide the user with a format to enter this range rather than entering a complete SQL type query in the **primary file query** field.

If an *initial format* will be required to retrieve data from the user at runtime, the format name must be entered in the **initial format** field in the *report.maint* form.

Accessing Forms Designer

To start Forms Designer:

For detailed information on using Forms Designer, see *ServiceCenter System Tailoring*.

- 1 When in the *report.maint* or *report.totals* form, go to **Options > Forms Designer**.

-or-

Click the Toolkit tab located on the system administrator's home menu.

The selections available in the Toolkit are displayed. Click the **Forms Designer** button.

The initial Forms Designer screen is displayed (see Figure 10-10 on page 276).

A Drawing Canvas and Tools Palette are displayed. When you begin to create the new report form, the Properties Window box also displays.

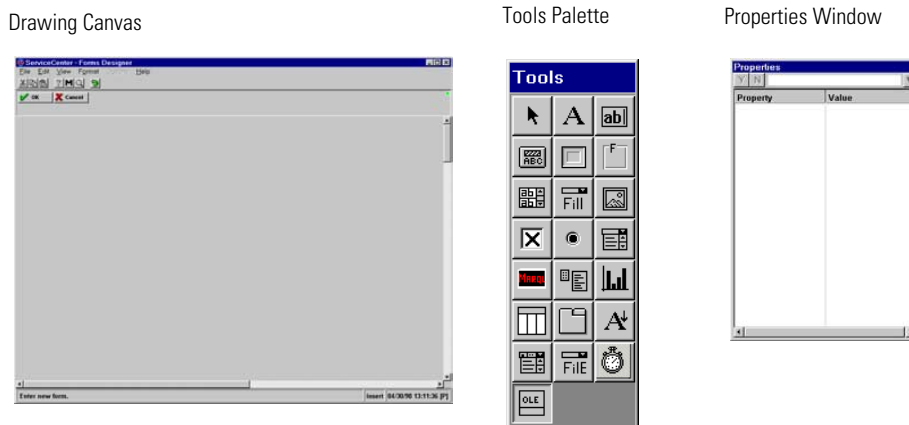


Figure 10-10: Drawing canvas, tools palette, and properties window in Forms

With the Drawing Canvas, Tools Palette, and Properties Window you can begin to design and create a new report.

For detailed information on creating a new report in Forms Designer, see *ServiceCenter System Tailoring* for the usual Forms Designer procedures.

For this example, we will access the *problem.category* report.

- 2 From the System Administrator's menu, enter `rw` on any menu command line and press **Enter** or click **Search**.

- 3 Enter `problem.category` in the **Report** field of the `report.prompt.rw` form.
The `report` form is displayed (Figure 10-2 on page 254).
- 4 Click **Edit** or press F2 (edit report) to access the edit functions of Report Writer.
Note: In the OS/390 environment, use `< >` instead of `~` (tilde).

The `report.maint` form is displayed (Figure 10-3 on page 265). See the *Field Descriptions for the report.maint Form* on page 255.
- 5 Go to **Options > Forms Designer** to create the format, using the usual Forms Designer procedures. For more information on Forms Designer, see *ServiceCenter System Tailoring*.
Note: `$start` and `$end` are the input field names for the fields shown when creating your new form in Forms Designer.
- 6 To include the information that will be entered in the initial format, change the primary file query statement to:
`number>$start and number<$end`
- 7 Press F5 to run the report.
Note: Reports using the *initial format* option cannot be run in the background. Therefore, the **Run report in background** field is defaulted to *f (false)*. See *How to Schedule Reports for Background Processing* on page 312 for more information.
- 8 Press F1 to complete the request.
See *How to Run Reports in a Foreground* on page 311 and *How to Schedule Reports for Background Processing* on page 312 for more information about running reports.

How to Use Page Breaks and Report Totals

In addition to producing simple report listings, Report Writer also allows for accumulation of various statistics, based on user-specified totals and parameters. These statistics can be printed, according to user-defined formats at various phases of report generation based on user-defined control/condition breaks. Variations of Totals Parameters can be used with or without Format Control definitions, which provide the ability to create a wide variety of report statistics.

Accessing the Totals Sections of a Report

This section demonstrates accessing the Totals section of a report and defines the various Totals Parameters, including how to reference them and how to create the formats that are used to print accumulated statistics.

To access the Totals section of a report:

For this example, we will access the `problem.category` report.

- 1 From the System Administrator's menu, enter `rw` on any menu command line.

The `report.prompt.rw` form is displayed.

- 2 Enter `problem.category` in the **Report** field.

The `report` form is displayed.

- 3 Click **Edit** or press F2 (edit report) to access the edit functions of Report Writer.

Note: In the OS/390 environment, use `< >` instead of `~` (tilde).

The `report.maint` form is displayed (Figure 10-3 on page 265). See the [Field Descriptions for the report.maint Form](#) on page 255.

- 4 Select **Options > Totals** to access the `report.totals` form.

This is a lengthy form, so use the scroll bars to move to the left and to the right, as well as up and down (see the form pictured in two sections, Figure 10-11 on page 279 and Figure 10-12 on page 279). See the [Field Descriptions of the report.totals Form](#) on page 280.

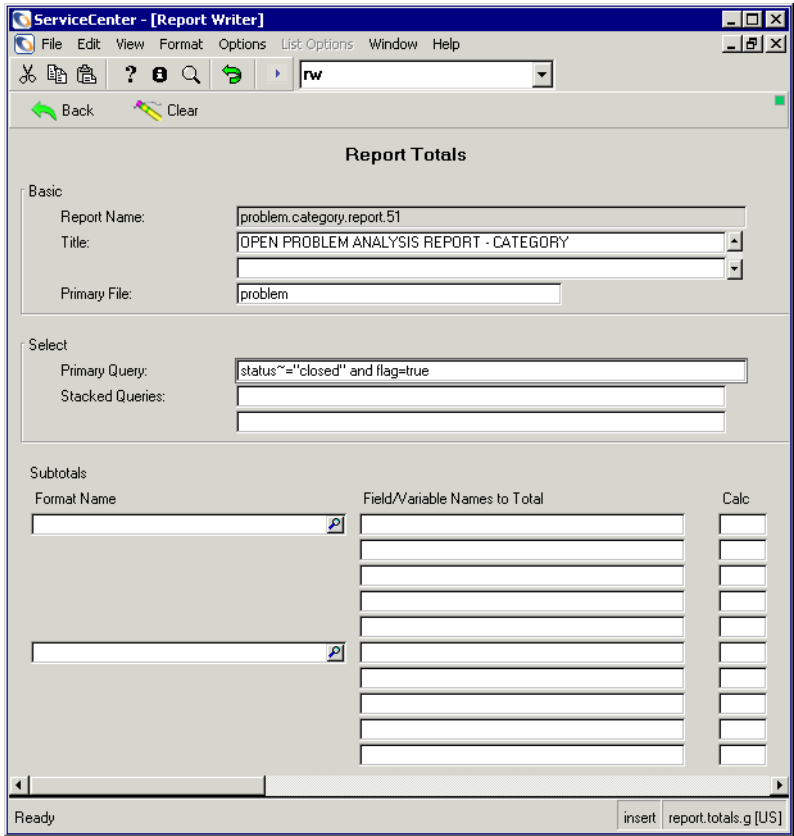


Figure 10-11: First section of the report.totals format

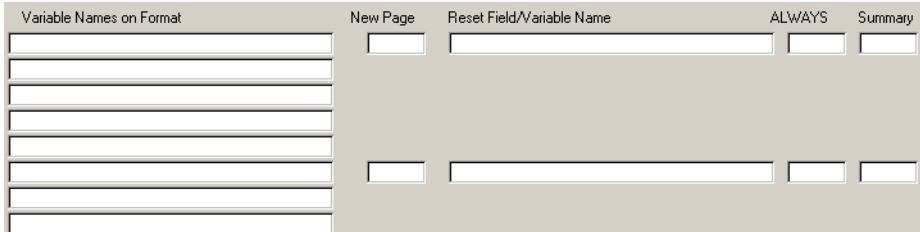


Figure 10-12: Second section of the report.totals format

Field Descriptions of the *report.totals* Form

Field	Description
<i>Totals Structure</i>	
Format Name	Name of the format used to print accumulated statistics at a user-defined control/condition break. More than one statistic can be accumulated and printed for each control break. To add totals to the Subtotals structure, select Options > Totals .
Field/Variable Names To Total	Either defined in the primary file's dbdict, defined as a variable in an earlier subtotals' definition within the same report, or defined as a variable in a Format Control definition for the <i>detail format</i> of the report.
Calc	One of four choices that indicates which type of calculation is performed, using the value of the field or variable defined in field/variable names to total . Note: The indicated calculation is performed for every detail record processed by the report, regardless of whether or not the detail format is printed. Valid Calc (calculation) types are: <p>SUM Accumulates the values of a specified field for each detail record extracted by the Primary File Query or Stacked Queries. If a variable is specified for the Field/Variable Name To Total field, then the value contained in that variable at the time each detail record is printed (or would have been printed if d was specified for summary or detail on the <i>report.maint</i> format) is accumulated in the variable specified in the Variable Names On Format field.</p> <p>The accumulation continues until some control/condition break is reached, at which time the accumulated total is stored in the variable defined in the Variable Names On Format field, the <i>subtotal format</i> is printed, and the variable used to accumulate the totals is reset to zero.</p>

Field	Description
AVG	<p>Accumulates the values of a specified field for each detail record extracted by the Primary File Query or Stacked Queries, and the total number of detail records read that produced the accumulated total.</p> <p>If a variable is specified for the Field/Variable Name To Total field, then the value contained in that variable at the time each detail record is printed (or would have been printed if d was specified for summary or detail on the <i>report.maint</i> format) is accumulated in the variable specified in the Variable Names On Format field.</p> <p>When a control/condition break is detected, the accumulated total is divided by the total number of records. This result is stored in the variable defined in the Variable Names On Format field.</p> <p>The control/condition break also causes the <i>subtotals format</i> to print, causing the variable used to accumulate the totals and the total number of records to be reset to zero</p>
MIN	<p>Examines the value of a specified field in each detail records and retains the lowest value encountered.</p> <p>If a variable is specified for the Field/Variable Name To Total field, then the value contained in that variable at the time each detail record is printed (or would have been printed if d was specified for summary or detail on the <i>report.maint</i> format) is accumulated in the variable specified in the Variable Names On Format field.</p> <p>When a control/condition break is detected, the value is stored in the variable defined in the Variable Names On Format field.</p> <p>The control/condition break also causes the <i>subtotals format</i> to print, causing the variable used to accumulate the totals and the total number of records to be reset to zero</p>
MAX	<p>Examines the value of a specified field in each detail record and retains the highest value encountered.</p> <p>When a control/condition break is detected, the value is stored in the variable defined in the Variable Names On Format field.</p> <p>The control/condition break also causes the <i>subtotals format</i> to print, causing the variable used to accumulate the totals to be reset to zero.</p>
Variable Names on Format	<p>Defines the variable that contains the results of the specified calculation. This variable is normally printed via the <i>subtotals format</i>. However, this is not a requirement. The variable can also be used in subsequent calculations.</p>

Field	Description
New Page	Indicates whether or not Report Writer should print the next detail record at the top of a new page <i>after</i> the indicated <i>subtotals format</i> has been printed. Valid entries are: <i>true</i> (yes), <i>false</i> (no), or <i>blank</i> (<i>false</i> and blank are synonymous). The default is blank. This can also be used to define a page break condition with no subtotals being printed.
Reset Field/Variable Name	The field name or variable name that is used to test for control/condition breaks. Every time the value of the field/variable changes, the indicated <i>subtotals format</i> is printed and all variables defined in Variable Names On Format field are reset to zero and the next detail record is printed after the report takes a top-of-page break. If no <i>subtotals format</i> is defined, only a top-of-page break is taken on the condition break.
ALWAYS	If <i>true</i> is specified, the <i>subtotals format</i> is printed every time a control break is detected (even if only one detail record has been subtotaled). If <i>false</i> (or blank) is specified, the <i>subtotal format</i> is <i>not</i> printed if only one detail record has been processed when a control/condition break is detected. The default is blank.
Summary	If <i>true</i> is specified, the totals defined will print in the summary format. If <i>false</i> (or blank) is specified, the <i>summary format</i> will not print. Enter true for all entries defining Summary Totals. Leave blank for all other types of totals.
Subtotal Expression	An expression that is evaluated against every detail record processed to further qualify when the indicated calculations will be performed. If the expression evaluated to <i>true</i> against the current detail record, the indicated calculations are performed. Note: The results of subtotals expressions are printed at the end of the report after all subtotals and before any summary totals.

Using Simple Subtotals

The following report demonstrates how to define the necessary totals parameters to cause Report Writer to print a *subtotals format*.

For this example, we will use the demonstration report, *problem.category.report*, created in *How to Create a Simple Report Listing* on page 252.

This report generates subtotals from the Primary file (probsummary), showing the total number of records for each Assignment Group.

To access the totals section within this report, see the instructions discussed in *Accessing the Totals Sections of a Report* on page 278. See the steps described below to create a totals format.

To create a Totals format in Forms Designer:

- 1 Enter the name of the format being used to print the subtotal in the **Format Name** field of the Totals structure (see Figure 10-11 on page 279).

For this example, enter `probsummary.demo.totals` and press **Enter**.

The system checks for a format called *probsummary.demo.totals* in the format file.

If the format already exists, it is displayed.

If the format does not exist, go to **Options > Forms Designer** to create a format using normal Forms Designer procedures. This format can be designed as you wish. For more information on Forms Designer procedures, see *ServiceCenter System Tailoring*.

- 2 Create each field, using the input properties described in *Input Properties Field Descriptions for the subtotals Form* on page 284.
- 3 Click **OK** to save your new form before exiting the Forms Designer to save the changes you made to the format.
- 4 Follow any other details described in the Forms Designer procedures discussed in *ServiceCenter System Tailoring*.
- 5 Scroll to the left to enter parameters in other fields on this format.

Note: Any time you scroll left or right while in this format, place the cursor in the home position before scrolling. This ensures that the labels and input fields remain properly aligned within the structured array.

- 6 To define the field by which you will be subtotaling, enter its name in the **reset/field variable name** field.

This causes the *subtotals format* (in this example, *problem.category.totals*) to print every time the value of the **assignment** field changes.

For example: If the **assignment** field value of the current detail record is **network control** and the next detail record has an **assignment** field value of **software support**, the subtotals are printed before the next detail record.

- 7 Scroll to the left again to enter parameters in the third section of this format.
8 Enter **true** in the **Always** field.

This forces the subtotals to print even if only one detail record is printed.

- 9 Press F3 (end) to return to the *report.maint* form.

Note: If you would like to erase all values entered in the *report.totals format*, press F4 (clear fields).

A confirmation prompt will display, allowing you to either press F1 (clear fields) to continue clearing the fields or press F3 (abort) to cancel deleting the *totals* information.

- 10 When you are ready to run this report, refer to *How to Run Reports in a Foreground* on page 311 and *How to Schedule Reports for Background Processing* on page 312 for complete instructions.

Input Properties Field Descriptions for the *subtotals* Form

The following input properties field descriptions are pre-defined by Report Writer. They are available for use when creating the *subtotals format* for any report you create in Forms Designer. For detailed information about Forms Designer procedures, see *ServiceCenter System Tailoring*.

Field	Description
report.count	Contains the total number of detail records processed by the report and is normally printed on a <i>grand totals format</i> , but may be printed on any totals format to show a running total.
report.subcount	An array that contains the total number of records processed for each calculation of each subtotal. If it is necessary to reference a particular array element on a <i>subtotals format</i> , the correct field name would be report.subcount,x where x is an array element number relative to the position of a calculation within the totals definition. This field is normally used in conjunction with the report.sumflds field for summary totals.

Field	Description
<code>report.sumflds</code>	An array that contains the value of each field that caused a control break. This field is normally used in conjunction with <code>report.subcount</code> for summary totals.
<code>report.subfld</code>	Contains the value of the field entered in the reset field/variable name at the time a control/condition break is detected.
<code>\$rw.dummy</code>	A variable defined by Report Writer that contains <i>NULL</i> characters.
<code>\$rw.months</code>	An array that contains full month names. <i>January</i> is the first element with <i>December</i> being the twelfth element.
<code>\$rw.mon</code>	An array that contains abbreviated (three-character) month names. <i>Jan</i> is the first array element with <i>Dec</i> being the twelfth element.
<code>\$rpt.date</code>	A variable defined by Report Writer that contains the date and time when the report was executed.
<code>\$page</code>	A variable defined by Report Writer that contains the page of the output of the report.

Using Subtotals, Summary Totals, and Grand Totals

This report demonstrates how to define the necessary Totals Parameters to cause Report Writer to print subtotals, summary totals, and grand totals.

Summary Totals

Display the same statistics that were printed in each subtotal, but in a summarized format. Changing the **summary/detail** field on the *report. maint format* to *s* will cause only the totals portions (subtotals, summary totals, and grand totals) of the report to print. The *detail format* will not print.

Grand Totals

Show the total number of detail records that were selected during report generation. This report will use the *enduser inventory* file as the *primary file*.

Create the report through Report Writer. See *How to Create a Simple Report Listing* on page 252 for detailed instructions on creating a report, including creating the outlined header and detail formats procedures.

- 1 Use the following parameters to create this report.
 - a Enter a unique name in the **Report Name** field.
For this example, enter **enduser.count**.
 - b Enter any user-defined title in the **Report Title** field.
For this example, enter **Monthly Inventory Count of Enduser Device by Model**.
 - c Select **Auto** in the **Auto/Manual** field.
 - d Select **Detail** in the **Summary/Detail** field.
 - e Enter **enduser** in the **Primary File name** field.
 - f Use the default value of **enduser.count.header** for the **Header Format Name** field.
 - g Use the default value of **enduser.count.detail** in the **Detail Format Name** field.
 - h Enter **true** in the **Primary File Query** field.
 - i Enter **model** in the **Sequence of Report** field.
 - j Use the following fields for the *detail format*:
 - model
 - logical.name
 - serial.no.
 - location
- 2 When you are finished adding the report, press F2 (edit rpt) to access the edit functions.
The report is displayed.
- 3 Press F4 (totals) to access the *report.totals format*.
- 4 Enter the name of the format being used to print the subtotal in the **Totals Format Name** field.
The *subtotals format* can be created through the normal Forms Designer procedures either before entering the Totals Parameters or during the Totals Parameters definition process. For more information on Forms Designer procedures, see *ServiceCenter System Tailoring*.

For this example, create the subtotal format **enduser.count.subtot** from the *report.totals* form.

- 5 Position the cursor on the **Totals Format Name** field and go to **Options > Forms Designer**.

The system checks for a format called **enduser.count.subtot** in the format file. If that format already exists, it is displayed.

If the format does not exist, a blank screen will display, allowing you to fill in the format using the normal Forms Designer procedures.

For information on Forms Designer procedures, see *ServiceCenter System Tailoring*.

The format may be designed as needed.

Note: The value in parentheses () (**\$rw.dummy**) is not intended to be a label on the format.

- 6 Edit each field to update the format. Be sure to update the format when prompted.
- 7 Click **OK** to return to the *report.totals* form.
- 8 To force a page break after each subtotal format prints, enter **true** in the **new page** field.
- 9 To access the second portion of the *report.totals* form, press **Home** to move the cursor from the array, and scroll to the left until you see the **label reset/field variable name**.

To define the field by which you will be subtotalling, enter its name in the **reset/field variable name** field. This causes the subtotals format (enduser.count.subtot) to print every time the value of the field **model** changes.

For this example, enter **model**.

- 10 Scroll to the left again to enter parameters in the third section of this format.

Using Calculations in a Report

Create the report through Report Writer. For detailed instructions on creating a report, including creating the outlined header and detail formats procedures, see *How to Create a Simple Report Listing* on page 252.

- 1 Use the following parameters to create this report.
 - a Enter a unique name in the **Report Name** field.
For this example, enter `problem.category2`.
 - b Enter any user-defined title in the **Report Title** field.
For this example, enter **Demo of How to Use Totals Calculations**.
 - c Enter `auto` in the **Report Type** field.
 - d Select **Detail** in the **Summary/Detail** field.
 - e Enter `probsummary` in the **Primary File name** field.
 - f Use the default value of `problem.category2.header` for the **Header Format Name** field.
 - g Use the default value of `problem.category2.detail` in the **Detail Format Name** field.
 - h Enter the following in the **Stacked Query** field: `flag=true`.

Note: OS/390 users can also enter `status<>="closed"`. Unix users can also enter `status~="closed"`.

 - i Enter **assignment** in the **Sequence of Report** field.
The system checks for a format called `enduser.count.total` in the format file.
If the format already exists, it will display.
If the format does not exist, a blank screen is displayed, allowing you to fill in the format using normal Forms Designer procedures.
For detailed Forms Designer procedures, see *ServiceCenter System Tailoring*. The format can be designed as needed.

j Use the following fields for the *detail format*:

number
 category
 assignment
 status
 open.time
 \$demo.elapsed

k When you are finished adding the report, press F2 (edit rpt) to access the edit functions. The report is displayed.

Note: In the OS/390 environment, use < > instead of ~ (tilde).

2 Press F4 (totals) to access the report.totals format.

3 Enter the name of the format being used to print the subtotal in the **totals format name** field.

For this example, enter `problem.category2.subtotal`.

If the subtotals format needs to be created, go to **Options > Forms Designer**. For additional information on accessing Forms Designer, refer to *Accessing Forms Designer* on page 276. For detailed procedures on working within the Forms Designer utility, see *ServiceCenter System Tailoring*.

For this example, create the `problem.category2.subtotal`.

If the format does not exist, a blank screen is presented (see Figure 10-3 on page 265). Fill in the detail format, following normal Forms Designer procedures. Select **Options > Forms Designer**. For information on Forms Designer, see *ServiceCenter System Tailoring*.

Note: When filling in a *detail* format for a report, only input fields are defined. However, this is not a restriction. Label fields can also be defined, if needed.

4 To calculate the *average* and *sum* of the variable `$demo.elapsed` for each grouping of problem records by category, enter the values shown for the **field/variable names to total** and **calc** fields.

5 Scroll left to view the second portion of the *report.totals* format.

The values entered in the **variable names on format** array identify where the result of the calculation will be printed within the subtotal format.

Enter the variable names used on the `problem.category2.subtotal` format.

- 6 Enter the value category in the **reset field/variable name** field to calculate the totals separately for each category grouping.
- 7 When you are finished entering the appropriate data, press F3 (end) to return to the *report.maint* screen.

Using Format Control to Evaluate the Value to be Used for the \$demo.elapsed Variable

Format Control records can be attached to any form or file within ServiceCenter and do not require special programming skills to implement. Routines defined in Format Control can be user interactive or transparent, and are performed when a record is displayed or when a user adds, updates, or deletes a record from the database. Format Control is easily applied and changed.

To learn more about Format Control, refer to *ServiceCenter System Tailoring*.

The calculations process within Format Control can be used whenever you want to perform a calculation on currently available fields or variables. To evaluate the value to be used for the \$demo.elapsed variable used in the *problem.category2.detail* format, you must attach Format Control to the detail format.

Accessing Format Control

The best method for accessing Format Control is through Forms Designer. This method reduces the chances of accessing the incorrect record, because you are accessing the record directly from the form with the same name as the Format Control record.

To access Format Control:

- 1 When in the *report.maint* or *report.totals* form, go to **Options > Forms Designer**.

The initial Forms Designer screen is displayed (see Figure 10-10 on page 276).

- 2 Enter the name of the form you wish to see in the **Form** field of the Forms Designer dialog box.

-or-

Click **Search** to find the form.

3 Go to Options > Format Control.

The Format Control record is displayed for the selected form.

Important: Be sure that the name of this Format Control record is the same name as your *detail format*.

Command Keys to be Used While in Format Control

Command Key	Description
F1 = add	To add the Format Control record for the <i>detail</i> format.
F2 = update	To update a Format Control record.
F3 = end	To exit the Format Control record when you are finished entering the calculation expressions. Continue to press F3 to exit and return to the <i>report.maint</i> form.
F5 = calcs	Accesses the calculations section of the Format Control record. Example of a calculation expression: if (flag=false) then (\$demo.elapsed=close.time in \$file - open.time in \$file) else (\$demo.elapsed=tod() - open.time in \$file)
F6 = more	Accesses other sections of the Format Control record.
F12 = end (no add)	Cancels out of creating a new report and does not save the information. Returns back to the <i>report.prompt.rw</i> format.

When you're ready to run the report, refer to *How to Run Reports in a Foreground* on page 311 and *How to Schedule Reports for Background Processing* on page 312.

How to Use Format Control with Reports

When a report is created using Report Writer, these reports use several forms to generate the report. There are the *Initial*, *Header*, *Detail*, *Subheader*, and *Totals* forms. Each of these forms can have Format Control records set up to be used by Report Writer when reports are executed at runtime of the report.

For detailed information on using Format Control, refer to *ServiceCenter System Tailoring*.

Format Control and Report Formats

Report Format	Description
Initial Formats	Format Control on the <i>initial</i> format can be set up to initialize data entered on the <i>initial</i> format, using the initialization expressions section to validate data entered on the <i>initial</i> format. Use the F1(add) option of any section. The Format Control record would look like Format Control on any database format.
Header Formats	Format Control on the <i>header</i> format can be set up to initialize variables that can be used by the other formats and are defined in the initialization expressions section of the Format Control record. If a variable is to be initialized only once while the report is being executed, it should be done in the <i>header</i> format.
Detail Formats	Format Control on the detail format can be set up to initialize or calculate data, select records from secondary files, and execute P4 subroutines, using the display option of the Format Control record. Validation expressions will have no effect in this version of the Report Writer. If a variable is set in the <i>detail</i> format, it will be reset each time a record is selected.

Using Format Control for Subheaders

The *subheader* format is used when a report is designed to print information from a file that immediately precedes detail information extracted from a secondary file.

Subheader reports require a Format Control record to be defined for the *subheader* format of the report, because the *subheader* format identifies the secondary file that is used to extract information for the *detail* format.

The *subheader* format will contain fields from the primary file defined in the report. The *detail* format will contain fields from the secondary file, which is defined using the Secondary File Queries section of the Format Control record of the *subheader* format.

The primary and secondary files are connected by a common field value.

For detailed information on using Format Control, refer to *ServiceCenter System Tailoring*.

In this example, we will match the value in the **location** field from the **location** file (primary file) with the value in the **location** field from the **enduser** file (secondary file). Each time the report selects a record based on the primary file query from the primary file (**location** file), Report Writer uses Format Control to find the matching records in the secondary file (**enduser** file) using the **location** field.

To create a report using Format Control:

- 1 Create a new report in Report Writer (see Figure 10-13 on page 295). For details on creating a new report, refer to *How to Create a Simple Report Listing* on page 252.

Use the following values when creating the report in the *report.maint* form.

Field	Value
Name	location.summary (or any unique name)
Title	A user-defined title
Auto or Manual	Auto.
Summary or Detail	Detail
Primary File	location
Header Format	location.summary.header (default)
Subhdr Format	location.summary.subhdr
Detail Format	location.summary.detail (default)
Primary Query	true
Sort Sequence	location

Note: location must be a key in the primary file's dbdict before it can be used as a sort field, since stacked queries are not used.

ServiceCenter - [enter data and select function]

File Edit View Format Options List Options Window Help

OK Cancel Run

Report Maintenance

Basic

Name: Author:
 Title: Editor:
 Application:
 Initial Form:
 Primary File:
 Query Form:

Save Report

Summary Auto
 Detail Manual

Formats

Header: Print master Header:
 Subhdr: Print Subheader:
 Detail: Subheader Page Break:

Select

Primary Query:
 Stacked Queries:
 Sort Sequence: Lines Per Page:
 Width (80/132):

Subtotals

Format Name	Field/Variable Names to Total	Calc	Variable Names on Format
<input type="text"/>	<input type="text"/>		
<input type="text"/>	<input type="text"/>		

Ready insert report.maint.g [S]

Figure 10-13: Creating a new report

- 2 Click **Options > Forms Designer** to create the format in Forms Designer. A blank form is displayed.

A Drawing Canvas and Tools Palette are displayed (see Figure 10-10 on page 276). When you begin to create the new report form, the Properties Window box also displays.

- 3 With the Drawing Canvas, Tools Palette, and Properties Window you can begin to design and create a new form for the report.

For detailed information on creating a new form for your report in Forms Designer, see *ServiceCenter System Tailoring* for the usual Forms Designer procedures.



- 4 When you are finished working within the Forms Designer utility, click **OK**. The new form is displayed in the location.summary (*forms.designerformat*) form. (If the format already exists, it is displayed.)
- 5 If you need to make further design changes, click **Design**.
- 6 When you are finished with your design changes, click **Options > Format Control** to create a new Format Control record.
- 7 You are prompted to save your changes. Click **Yes**.
The *formatctrl.maint.initial* form is displayed (Figure 10-14 on page 296).

Important: Make sure the name of your Format Control record is the same name as your *subheader* format.

In this example, there are no entries required in the **Initialization Expressions** field.

ServiceCenter - [Format Control]

File Edit View Format Options List Options Window Help

Back New Search

Format Control Maintenance - Main Information

Name:

File Name:

System:

Query Format:

Default QBE Fmt:

Save Copy

Stored Form Name

Run Script

Use Default Sort

Default Sort sequence for queries

Initialization Expressions

* No records found. insert | formatctrl.maint.initial.g(fc.search) | [S]

Figure 10-14: Format Control Maintenance form

8 Click New.

The following message is displayed in the status bar: *Format Control record added.*

9 Click Options > Queries.

10 Click Yes.

The *formatctrl.maint.queries.b* form is displayed (Figure 10-15 on page 297).

ServiceCenter - [Format Control location.summary]

File Edit View Format Options List Options Window Help

OK Back Add Save Delete

Views Queries Calculations Validations Subroutines Add Options Privileges

Format Control Maintenance - File Queries

Name: location.summary View: short

Add	Upd Query	Del	Dis	Req'd	Init	Filename	Field Name to Check
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>

Ready insert formatctrl.maint.queries.b.g(fc.view) [S]

Figure 10-15: Format Control Maintenance - File Queries form

- 11 Make the file queries entries as needed.

Field Value Entries

Field	Description
Dis	Enter true in the Dis (display) field.
Filename	Enter enduser . This identifies the enduser file as the secondary file for this report. The location file and enduser file will be connected when the values in their location fields match.
Query	Enter <code>location=location in \$file</code> .
Req'd	Enter false in the Req'd (required) field. This signifies that the match is not required for the subheader format to print.

- 12 Click **Save** to save the record.
- 13 Click **OK** to exit the forms and return to the *report.maint* form.
To learn more about creating Format Control records for your reports, refer to *ServiceCenter System Tailoring*.

Detail Formats

To create detail formats:

For information on creating a report in Report Writer, refer to *How to Create a Simple Report Listing* on page 252.

For this example, we will use the *location.listing* report.

- 1 From the *report.maint* form, put your cursor in the **Detail** field of the Formats structure of the *report.maint* form (on the *location.summary.detail* value) and go to **Options > Forms Designer**.

The system checks for a format called *location.summary.detail* in the format file. If the format already exists, it is displayed (Figure 10-16). If it does not exist, a blank screen is presented, which allows you to paint the format using normal Forms Designer procedures.

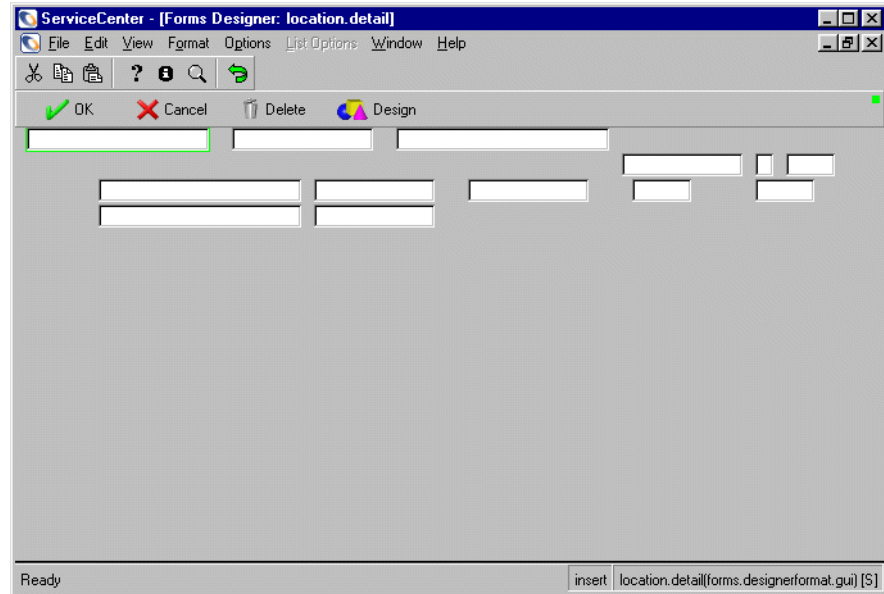


Figure 10-16: *location.detail (forms.designerformat) form*

For information on accessing Forms Designer, see *Accessing Forms Designer* on page 276. For detailed information on using Forms Designer, see *ServiceCenter System Tailoring*.

- 2 Click **Design** to create the format or change the existing format.

The *location.detail* design form is displayed (Figure 10-17).

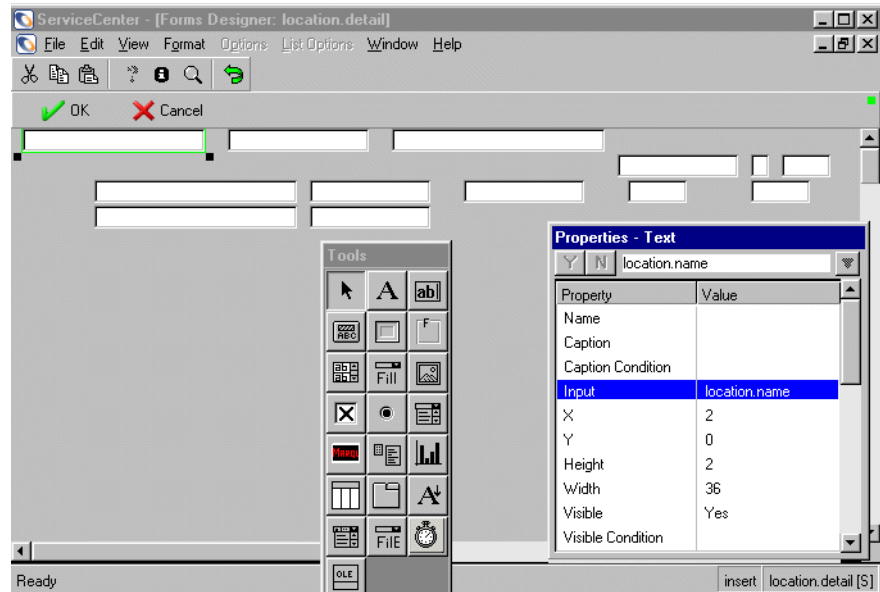


Figure 10-17: Input fields within the Properties - Text dialog box

The input fields for each field within the Forms Designer palette to be noted in the Properties - Text dialog box are as follows.

Input Fields

- | | |
|-------------------|-----------------------|
| ■ location.name | ■ phone |
| ■ location | ■ crt.phone |
| ■ address | ■ loc.open.time |
| ■ city | ■ loc.close.time |
| ■ state | ■ other.contact |
| ■ zip | ■ other.contact.phone |
| ■ primary.contact | |

Note: The value in the **Input** field is the name of that **Input** field and is *not* intended to be a label on the format. You *must* enter a name into the **Caption** field of the Properties - Text dialog box in order to have the name of the field displayed on the report.

- 3 Click **Add** or **OK** to return to the *report.maint* form.

Using Format Control in a Matrix Report

The following sample report gives summary count information on severity codes and the status of Incident tickets. The severity code labels print in the column and the status (such as, open and update) labels print in a row.

The report consists of a *header* and two *totals* formats. It is a summary report, but it does not use the summary field (true/false) on the totals screen. There is a Format Control record associated with both the *header* and *detail* formats. They contain initialization and calculation expressions for processing record counts.

To create a matrix report using Format Control:

- 1 For details on creating a new report, refer to *How to Create a Simple Report Listing* on page 252.

Use the following values when creating the report in the *report.maint* form.

Field	Value
Name	problem.severity.summ (or any unique name)
Title	A user-defined title
Auto or Manual	Auto.
Summary or Detail	Summary
Primary File	probsummary
Header Format	problem.severity.summ.header (default)
Detail Format	problem.severity.summ.detail (default)

Field	Value
Primary Query	flag=true Note: OS/390 users can also enter status<>="closed". Unix users can also enter status~="closed".
Sort Sequence	severity Note: In order to sort and break on the severity field for a summary type report, the same field must also be set up as a <i>key</i> in the probsummary dbdict, or the primary query could be defined as a stacked query. For more information on the Database Dictionary (dbdict), refer to <i>ServiceCenter System Tailoring</i> .

- When you have all the field values entered (Figure 10-18 on page 302), place the cursor in the Header field (on the problem.severity.summ.header value) and click **Options > Forms Designer**.

Figure 10-18: Creating a matrix report

The system checks for a format called *problem.severity.summ.header* in the format file. If the format already exists, it is displayed. If it does not exist, a blank screen is presented (Figure 10-19 on page 303), which allows the user to paint the format using normal Forms Designer procedures.

For detailed information on the Forms Designer procedures, refer to *ServiceCenter System Tailoring*.

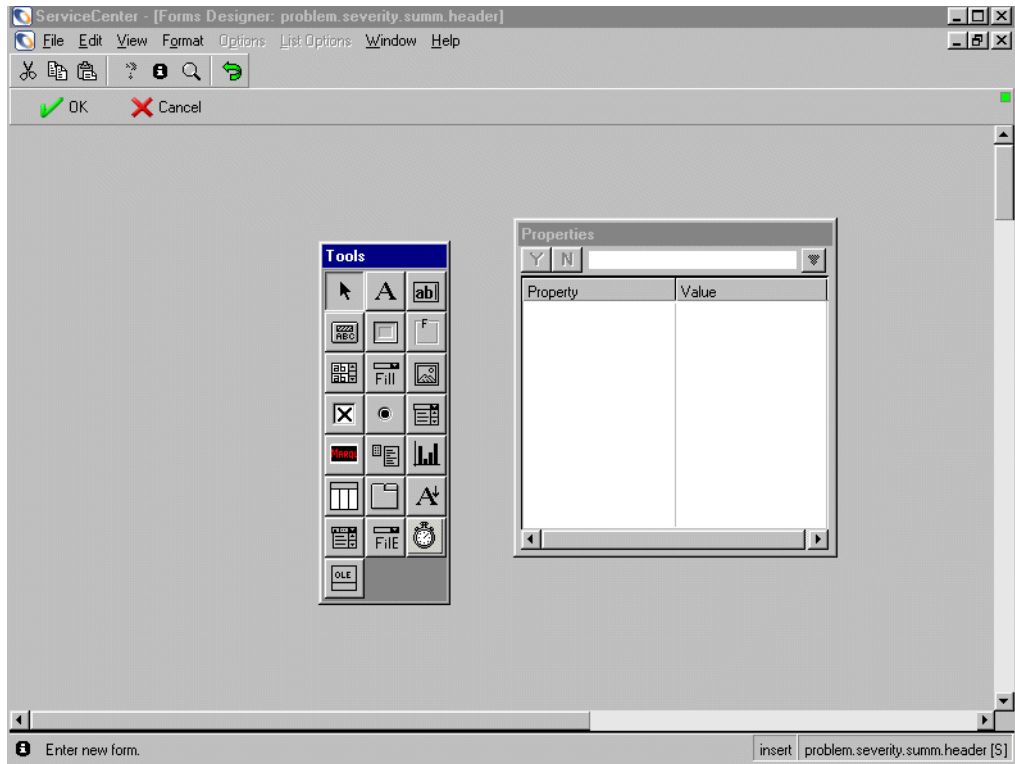


Figure 10-19: Blank *problem.severity.summ.header* form

Note: In order to sort and break on the severity field for a summary type report, the same field must also be set up as a key in the probsummary dbdict, or the primary query could be defined as a stacked query.

- 3 Click OK to add the new format and return to the *report.maint* form.
- 4 To create a *detail* form, place your cursor in the **Detail** field (on the *problem.severity.summ.detail* value) and click **Options > Forms Designer**.

The system checks for a format called *problem.severity.summ.detail* in the format file. If the format already exists, it is displayed. If it does not exist, a blank screen is presented, which allows the user to paint the format using normal Forms Designer procedures.

Note: Since this report is a summary type report, the *detail* format will not print when the report is generated. However, because a Format Control record is required for calculations to occur on each detail record, we must create a *detail* format to attach to the Format Control record.

Subtotal Format

To create the subtotal format:

- 1 Position the cursor on the first Subtotals **Format Name** field and click **Options > Forms Designer**.

The initial Forms Designer (*format.prompt.initial*) form is opened.

- 2 Enter *problem.severity.summ.sub* and click **New**.
- 3 A question is displayed that asks if you want to use the Forms Designer wizard. Click **No**.

A blank Forms Designer palette (*problem.severity.summ.sub* form) is opened.

- 4 Follow normal Forms Designer procedures to create a new form. For information on Forms Designer procedures, refer to *ServiceCenter System Tailoring*.

Note: The value in parentheses in the Properties - Text dialog box is the name of the **Input** field and is not intended to be a label on the format. You must enter a caption in the **Caption** field in order to have the field name displayed on the report.

- 5 When you are finished, click **OK**.
- 6 From the **Options** menu, select **Totals**.

- 7 Fill in the required fields in the *report.totals* format as follows:

Subtotals Format Name	Field/Variable Names to Total
problem.severity.summ.sub	\$opn
	\$upd
	\$al1
	\$al2
	\$al3

- 8 Access the second portion of the *report.totals* format and scroll to the right until you see the **Reset Field/Variable Name** field.
Enter severity.
- 9 Enter true in the ALWAYS field.
Entering true causes the subtotals to print even if only one detail record is selected before the subtotal is reset.
- 10 Click OK.

Total Format

To create the total format:

- 1 Position the cursor on the second totals **Format Name** field and enter `problem.severity.summ.tot` and click **Options > Forms Designer**.
A blank Forms Designer palette is opened.
- 2 Follow normal Forms Designer procedures to create a new form. For information on Forms Designer procedures, refer to *ServiceCenter System Tailoring*.
Note: The value in parentheses in the Properties - Text dialog box is the name of the **Input** field and is not intended to be a label on the format. You must enter a caption in the **Caption** field in order to have the field name displayed on the report.
- 3 When you are finished with editing, click OK.
- 4 From the **Options** menu, select **Totals**.

- 5 Fill in the required fields in the *report.totals* format as follows:

Totals Format Name	Field/Variable Names to Total
problem.severity.summ.tot	\$opn
	\$upd
	\$al1
	\$al2
	\$al3

- 6 Scroll down and fill in the Variable Names on the Format field as follows.

Totals Variable Names on Format

\$open

\$updt

\$alrt1

\$alrt2

\$alrt3

\$opn.tot

\$upd.tot

\$alrt1.tot

\$alrt2.tot

\$alrt3.tot

- 7 Scroll over until you see the **ALWAYS** field.
Enter true.
- 8 When you are finished, click **OK**.

Format Control Record for the Header format

To create a Format Control record for the header format:

- 1 Position the cursor on the Format **Header** field and click **Options > Forms Designer**.
- 2 Create the necessary fields using the normal Forms Designer procedures.
For detailed information on the Forms Designer procedures, see *ServiceCenter System Tailoring*.
- 3 Click **OK**.
- 4 Click **Options > Format Control**.
- 5 When prompted, click **Yes**.

This will take you into the Format Control utility and allow you to add a new Format Control record (see Figure 10-14 on page 296).

Note: Be sure that the name of this Format Control record is the same name as your *header* format.

- 6 Enter the following statements in the **Initializations Expressions** field.

Initializations Expressions Values

```
$open=0 ; $updt=0 ; $alrt1=0 ; $alrt2=0 ; $alrt3=0
```

```
$opn.tot=0 ; $upd.tot=0 ; $alrt1.tot=0 ; $alrt2.tot=0 ; $alrt3.tot=0
```

Note: These initialization expressions will be executed only once during the report generation.

- 7 When you are finished, click **OK**. Continue to click **OK** until you return to the *report.maint* form.

Format Control Record for the Detail format

To create a Format Control record for the detail format:

- 1 Position the cursor on the Format **Detail** field and click **Options > Forms Designer**.
- 2 Create the necessary fields using the normal Forms Designer procedures.
For detailed information on the Forms Designer procedures, see *ServiceCenter System Tailoring*.
- 3 Click **OK**.
- 4 Click **Options > Format Control**.
- 5 When prompted, click **Yes**.

This will take you into the Format Control utility and allow you to add a new Format Control record (see Figure 10-14 on page 296).

Note: Be sure that the name of this Format Control record is the same name as your *detail* format.

- 6 Enter the following statements in the **Initializations Expressions** field.

Initializations Expressions Values

```
$open=0 ; $updt=0 ; $all=0 ; $al2=0 ; $al3=0
```

Note: These initialization expressions will be executed each time a new *detail* record is selected.

- 7 When you are finished, click **OK**.
- 8 Click **Calculations** in the Format Control Maintenance form.

The following Format Control Maintenance - Calculations form is displayed (Figure 10-20 on page 309).

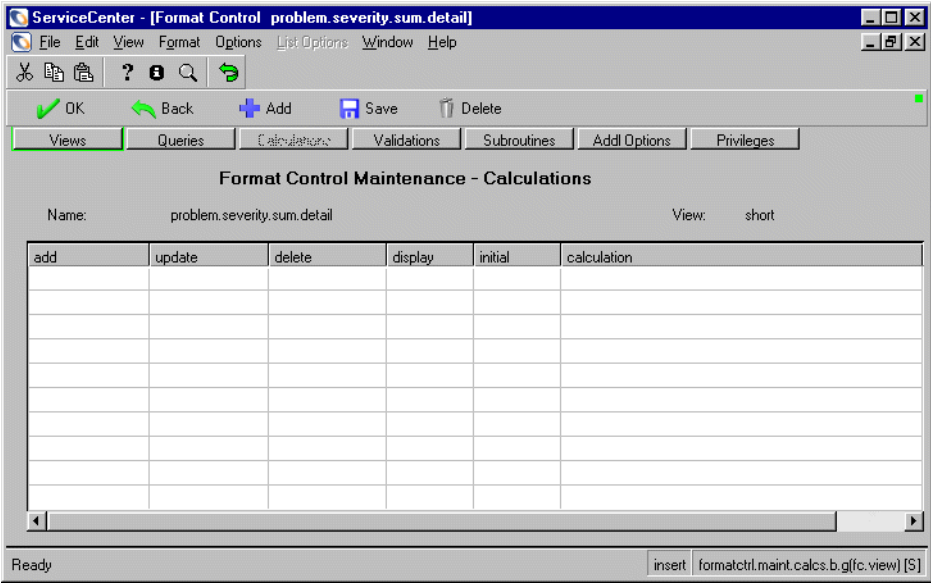


Figure 10-20: Format Control Maintenance - Calculations form

- 9 Enter the values in the fields as described below.

Field	Values to be entered
display	true
calculation (<i>enter all on one line</i>)	If status in \$file#"ope" enter: \$opn=1
	If status in \$file#"upd" enter: \$upd=1
	If (index("1",status in \$file)>0) enter: \$al1=1
	If (index("2",status in \$file)>0) enter: \$al2=1
	If (index("3",status in \$file)>0) enter: \$al3=1
	Note: To increase performance, the above expressions should be spread out on one expression line (separated by semicolons).
	Important: <i>Never</i> place a semicolon at the end of the final expression. This will cause the system to expect another expression.

- 10 When you are finished, click **OK**. Continue to click **OK** until you return to the *report.maint* form.

How to Run Reports in a Foreground

Running a report in the foreground dedicates the user's session and terminal to the task until the report has completed.

To set up reports to run in the foreground:

- 1 Access the report you wish to run. (For details on accessing a report, see *How to Access Report Writer* on page 250.)

The report form is displayed (see Figure 10-2 on page 254).

- 2 When you have access to the *report* form, check the information to make sure it is correct.

Click **Edit** or press **F2** to make any necessary modifications.

- 3 Click **Run** or press **F5** to run the report.

Note: In the OS/390 environment, use `< >` instead of `~` (tilde).

- 4 From the `report.prompt.exer` form, you have the opportunity to modify the **summary or detail** and **primary file query** definitions.

- a The **Print report after spooling** field defaults to `true` (yes). Change this to `false` (no).

- b The **Run report in background** field defaults to `true` (yes). Change this to `false` (no).

If you leave the default set to `true`, the report will run at a set time in the background processor. To learn more about running reports in the background, see *How to Schedule Reports for Background Processing* on page 312.

If you change the default setting to `false`, the report will be scheduled to pass to the printer and print immediately after it has been generated.

After the report is finished spooling, the following message is displayed in the status bar: *Report spooled as no: xxx (report title)*.

- c The *report* form is redisplayed.

- Press **F1** to view the report. The spoolheader format is displayed, which allows you to view the report output online. For more information about viewing reports, see *Online Viewing of Reports* on page 320.

-or-

- Press **F3** to end the task.

Note: In the OS/390 environment, use `< >` instead of `~` (tilde).



How to Schedule Reports for Background Processing

Understanding Background Report Processing

Running a report in the background means that a simulated user session within the ServiceCenter system is used to generate the report. The default background processor is named *report* and is normally made active, along with other background processors every time ServiceCenter is started, and is the processor that is used to process the report. The *info* file contains the definitions for all background processors and also defines which of those processors are automatically started at startup time.

When a report is scheduled to run in the background, information about the report is first added to the *schedule* file. The report background processor scans the *schedule* file every 60 seconds, which is the default scan interval, for any records whose scheduled time has elapsed and whose class is defined as *report*. When a record is found, the background processor performs all functions necessary to generate the report to the *spool* and *spoolheader* files. Reports may be scheduled to run in the future and be given a repeat interval to automatically reschedule. For example, you can schedule a report to run once every few hours, once a day, or once a week.

The *spoolheader* file contains information about the run statistics of the report (such as, the time the report started processing, the time the report finished, the total number of pages printed, and which operator scheduled the report for processing). The *spoolheader* file is also linked to the *spool* file.

The *spool* file contains the actual pages of the report. You can access the *spoolheader* file and view the generated reports online. To learn more about viewing reports online, refer to [Online Viewing of Reports](#) on page 320.

Normally after a report has been generated to the *spoolheader* and *spool* files, it is passed to the OS/390 or Unix systems based on parameters defined by the user at the time the report was scheduled to run.

Scheduling a Report for Background Processing

To schedule a report for background processing, access the report you wish to run in the background processor. To schedule a report for background processing:

- 1 Access a report, following the procedures discussed in *How to Access Report Writer* on page 250.

In this example, the *Problem Response Analysis by Assignment* report is opened in the *report.maint* form.

The screenshot shows the 'Report Maintenance' window in ServiceCenter. The window title is 'ServiceCenter - [enter data and select function]'. The menu bar includes File, Edit, View, Format, Options, List Options, Window, and Help. The toolbar contains icons for Cut, Copy, Paste, Help, Find, Undo, and a dropdown menu showing 'rw'. Below the toolbar are buttons for 'OK', 'Cancel', and 'Run'.

The main area is divided into several sections:

- Basic:**
 - Name: problem.response.analysis.assign.51
 - Title: Problem Response Analysis By Assignment
 - Application: (empty)
 - Initial Form: (empty)
 - Primary File: probsummary
 - Query Form: (empty)
 - Author: (empty)
 - Editor: falcon
 - Date: 02/16/03 20:33:50
 - Save Report
 - Summary, Detail
 - Auto, Manual
- Formats:**
 - Header: enduser.report.hdr.ass
 - Subhdr: (empty)
 - Detail: problem.resp.analysis
 - Print master Header: Yes
 - Print Subheader: (empty)
 - Subheader Page Break: (empty)
- Select:**
 - Primary Query: flag=true
 - Stacked Queries: (empty)
 - Sort Sequence: assignment
 - Lines Per Page: 66
 - Width (80/132): 80
- Subtotals:**
 - Format Name: enduser.report.detail.ass
 - Field/Variable Names to Total: \$pra.elapsed
 - Calc: AVG
 - Variable Names on Format: \$elapsed.time

The status bar at the bottom shows 'Ready' and 'insert report.maint.g [US]'.

Figure 10-21: Accessing a report to schedule background processing

- From the report.maint form, ensure that all the information about the report is correct.

If any information needs to be modified, click **Edit** or press F2 and make the necessary changes.

- When you are finished editing the report, click **Run** or press F5 to schedule the report to be processed in the background.

Note: In the OS/390 environment, use < > instead of ~ (tilde).

The Run Report Exerciser (*report.prompt.exer*) form is opened.

Figure 10-22: Selecting report run options in the Run Report Exerciser

- Click **Summary** to print a summary report.

-or-

Click **Detail** to print a detailed report.

- If necessary, modify the **primary file query** field definitions.

To learn more about query formats within reports, refer to *How to Use Query Formats with Reports* on page 268.

- 6 Select one of the following print options.
 - Select (check) **Print** to print the report when it is finished spooling. The report will print immediately after it has been generated.
Otherwise, leave the field blank (unchecked) to only spool the report. The report will remain in the **spool** file until it is manually deleted or scheduled to print.
-or-
 - Select (check) **Background** to print the report in the background when it is finished spooling. The report is scheduled to run at a particular time (defined in the next screen sequence) in a background processor.

Otherwise, leave the field blank (unchecked) to print the report in the user's session.

To learn more about running reports in a user's session, refer to *How to Run Reports in a Foreground* on page 311.
- 7 Enter the *number* of lines per page in the **Lines Per Page** field that you would like printed on the report. Otherwise, it will default to the original settings entered in the *report* form.
- 8 Click **Run**.

The Schedule a Report (*report.prompt.sch2*) form is opened (Figure 10-23 on page 316).

Figure 10-23: Scheduling a report to print

- 9 Schedule the report to print by entering the values in the fields of the *report.prompt.sch2* form. The fields are described below.

Field Descriptions

Field	Description
Time to Print	The date and time that the report will begin processing in the report background processor. It is defaulted to the current date and time, but can be changed as needed.

Repeat Interval structure

Field	Description
Specific Int	<p>Defines a specific interval to establish a periodic frequency for when the report will be automatically rescheduled for processing.</p> <p>This is a relative date and time field and is added to the Time to Print field value each time the report is generated by the background scheduler.</p> <p>Note: If this field is used, <i>do not</i> select a frequency within the Monthly, Quarterly, Semi-Annually, or Annually options listed below.</p>
Monthly Quarterly Semi-Annually Annually	<p>Indicates the frequency in which the report should be run. Type X to select one of the following options: Monthly, Quarterly, Semi-Annually, or Annually.</p> <p>Note: Select only <i>one</i> option, otherwise you will receive a message stating that you must make only one selection.</p> <p>Note: If this field is used, <i>do not</i> use the Specific Int field.</p>
<i>Printer structure</i>	
Name	<p>Defines the printer to be used when printing the report. This is defaulted to being the same printer value as defined in the operator record for the user login name running the report. This can be changed to any valid printer on the network.</p> <p>To obtain a valid list of all the available printers, put your cursor in the Name field and click Browse. Double-click a printer in the Device Name field to select it from the list.</p>
Class (OS/390 only)	<p>Defines the <i>sysout</i> class that overrides the <i>sysout</i> class defined in the config file. If this field is left blank, the printer class is defaulted to the same class as defined in the config file for the printer selected in the Name field.</p>
<i>Controls structure</i>	
Number of Copies	<p>The total number of copies of the report to be printed. The default number is one copy, but this number can be increased as needed.</p>

Field	Description
Delete Report	<p>Specify whether or not the report will be automatically deleted from the spool and spoolheader files after it has been printed.</p> <p>If set to <i>true</i> (yes), the report will be deleted. If set to <i>false</i> (no), the report will be saved in the spool and spoolheader files for viewing or rescheduling after it has been printed. The default setting is <i>true</i> (yes).</p>
Dist Group	<p>Defaults to <i>blank</i>. You can optionally click Browse to select a Group Name in the QBE list. Only one group can be selected as the distribution group.</p> <p>If a group is specified, all operators defined in the distribution group will receive the number of copies specified in the Number of Copies field. The copies for all operators will print to the designated printer.</p>

- 10 After you have entered all the necessary values to generate the report, click **Run**.

You are returned to the Report Maintenance (*report*) form (Figure 10-24 on page 319) and the following message is displayed in the status bar: *Approx. time of exec: (00:00:00) for report: (name of report).*

ServiceCenter - [Report Exerciser]

File Edit View Format Options List Options Window Help

Back View

Report Maintenance

Basic

Name: Author:

Title: Editor:

Application:

Initial Form:

Primary File:

Query Form:

Save Report

Summary Auto Detail Manual

Formats

Header: Print Master Header:

Subheader: Print Subheader:

Detail: Subheader Page Break:

Select

Primary Query:

Stacked Queries:

Sort Sequence: Lines Per Page:

Width (80/132):

Subtotals

Format Name	Field/Variable Names to Total	Calculation	Variable Names on Format
<input type="text" value="enduser.report.detail.ass"/>	<input type="text" value="\$pra.elapsed"/>	<input type="text" value="AVG"/>	<input type="text" value="\$elapsed.time"/>
<input type="text" value=""/>	<input type="text" value=""/>	<input type="text" value=""/>	<input type="text" value=""/>

Approx. time of exec: 00:00:03 for report: Problem Response Analysis By Assignment

Figure 10-24: Confirmation message received after all necessary values entered to run the report

A record has been added to the schedule file that now contains all the pertinent run information.



11 Press **Enter** to erase the message and return to Ready.

Click **View** to view the report. To learn more about viewing reports, refer to *Online Viewing of Reports* on page 320.

Online Viewing of Reports

When a report is generated in the foreground or in the background processor, it is added to the **spoolheader** and **spool** files to be printed or available for online viewing. The contents of the report cannot be modified, but the **spoolheader** file information can be adjusted, if necessary.

The **spoolheader** and **spool** files are both used to view reports. The **spoolheader** file defines such things as the name of the report, the time the report started spooling and ended spooling, and the operator who scheduled the report to be processed. The **spool** file contains the actual report (each page being a separate record within the **spool** file). The **spoolheader** and **spool** files are then linked to each other.

Accessing the Spoolheader and Spool Files

To view a report online:

- 1 Access a report, following the procedures discussed in *How to Access Report Writer* on page 250.

In this example, the *resolution code listing* report is opened in the *spoolheader* form (Figure 10-25 on page 321).

The fields in the *spoolheader* form are described in *Spoolheader Form Fields* on page 322.

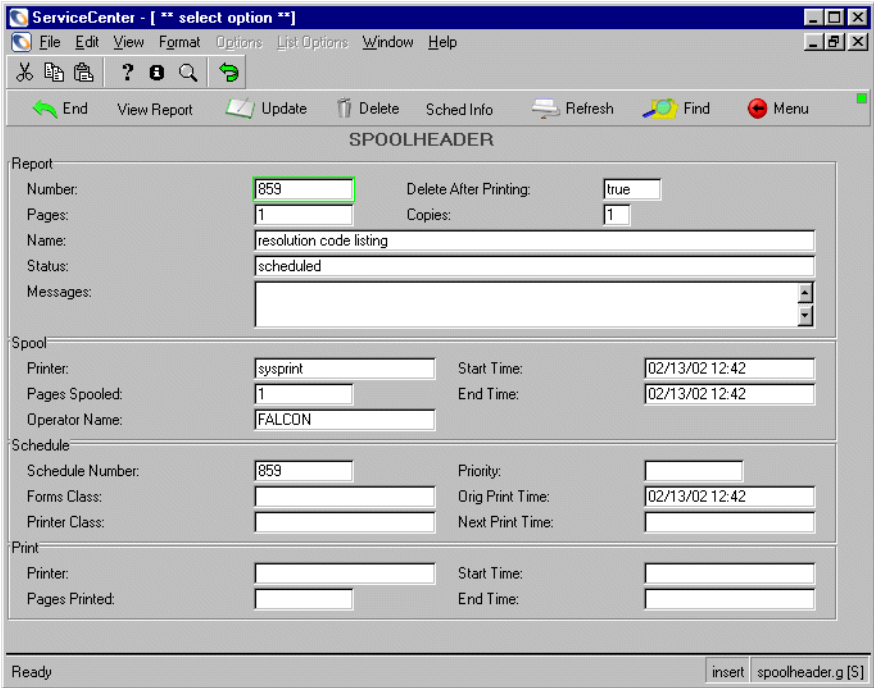


Figure 10-25: Accessing a report in the spoolheader file

Spoolheader Form Fields

Field	Description
<i>Report structure</i>	
Number	The unique report number assigned by the ServiceCenter system.
Pages	The number of pages generated in the report (the total number of times the report issued a top-of-page break).
Copies	The total number of report copies that will be printed (defined when the report was scheduled for background processing).
Name	The report title, as defined in the Report Writer record.
Status	<p>The current status of the report. This value is controlled by the ServiceCenter system. The following is a list of valid report statuses:</p> <ul style="list-style-type: none"> spooling - The report is current being processed in a user's session or in a background processor. spooled - The report is finished spooling and is ready for online viewing. scheduled - The report output has been scheduled to print. printing - The report is currently printing. printed - The report has completed printing and the delete report after printing field setting was <i>false</i> when the report was scheduled to print. (The report was set <i>not</i> to be deleted.) <p>For more information on the values to be set when scheduling a report to print, see How to Schedule Reports for Background Processing on page 312.</p>
<i>Messages</i>	
<i>Spool structure</i>	
Printer	Indicates which internal configuration definition is being used to obtain device information when the report is set to print.
Pages Spooled	Indicates the total number of pages generated by the report. If the report is currently being generated in the background, this number will increment until the report is finished processing.

Field	Description
Operator Name	<p>The operator who initiated the report for background processing or who actually ran the report in the user session.</p> <p>When the spoolheader form is initially displayed, this is defaulted to the operator ID of the current operator. If you leave the system default operator ID and press Enter, the system will search the spoolheader file for all records that have been generated by that operator ID. You can modify this field to search for any valid operator ID, or blank out the field altogether.</p> <p>Note: If you attempt to search the <i>spoolheader</i> records based on the default operator ID and no records are found, the system will return a message that states: <i>no records found</i>. The operator field will be blanked out.</p>
Start Time	The date and time that the report began processing in the foreground or background processor.
End Time	The date and time that the report ended processing in the foreground or background processor.
<i>Schedule structure</i>	
Schedule Number	After a report is scheduled to run, this field is filled in with a unique schedule file number associated with the report.
Forms Class	Any data filled in will override the <i>forms</i> value specified in the <i>config</i> record.
Printer Class	Any data filled in will override the <i>printer class</i> value specified in the <i>config</i> record.
Priority	<i>Not currently being used.</i>
Orig Print Time	The date and time that the report is scheduled to being spooling.
Next Print Time	When the report is run in the foreground and the print time is scheduled to run at a specific date with a repeat interval specified, the Next Print Time field reflects the scheduled date and the repeat interval.
<i>Print structure</i>	
Printer	Indicates the printer class to which the report is being generated.

Field	Description
Pages Printed	Indicates the total number of pages that have been generated up to the current time.
Start Time	Indicates the date and time when the report began to generate.
End Time	Indicates the date and time when the report finished generating.

- You can use any of these fields to search for *spoolheader* records. Place your cursor in the field on which you would like to base your search, and then click **Find** or press **F8**.

A QBE list is displayed.

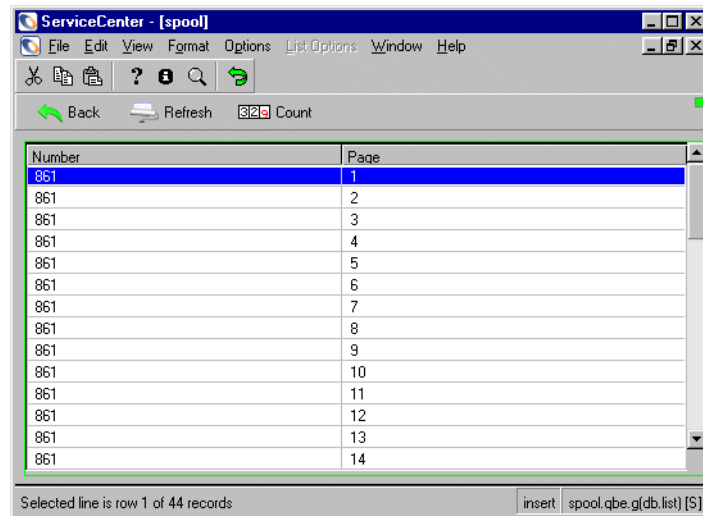


Figure 10-26: Searching spoolheader records

- 3 You have the following options when viewing the spoolheader records.
 - Put your cursor on any of the page number line items and press **Enter** to view that page of the report.
 - Press **Next** or **Previous** to scroll through the previous and next pages of the report.

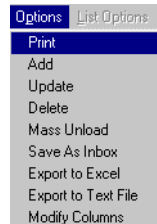


Figure 10-27: Report Writer QBE list Options menu

- Click **Refresh** to refresh your view of the report.
- Press **F4** to delete the report.
- Click **Back** or press **F3** to finish viewing the report.

How to View Report Schedule Records

Whenever a report is scheduled for background processing, it is first added to the schedule file with a scheduled class of *report*. The schedule definition remains in effect until the report is processed by the report background processor, or until the definition is deleted.

When the background processor begins processing the report, the *class* field is changed from *report* to *blank* and the *status* field is updated to *application running*. When both of these conditions exist, the report is actively being processed by the *report* background processor.

If a repeat interval was specified in the schedule record, the same schedule record is used to re-define when the report should be spooled again. Report Exerciser (the application called by the background processor) takes the original expiration time and adds the repeat interval to it, updates the status to re-scheduled, and changes the application name back to its original setting. See the Report Run Exerciser displayed in Figure 10-22 on page 314.

How to Copy/Rename Reports

To copy and rename reports:

- 1 Access a report, following the procedures discussed in *How to Access Report Writer* on page 250.

In this example, the *ocml.category.print* report is opened.

- 2 Click Edit or press F2 to access the edit functions of Report Writer.

Note: In the OS/390 environment, use < > instead of ~ (tilde).

- 3 Select **Options > Copy/Rename**.

The Report Entry (*report.prompt*) form is displayed with the current name of the report in the Name field (Figure 10-28).

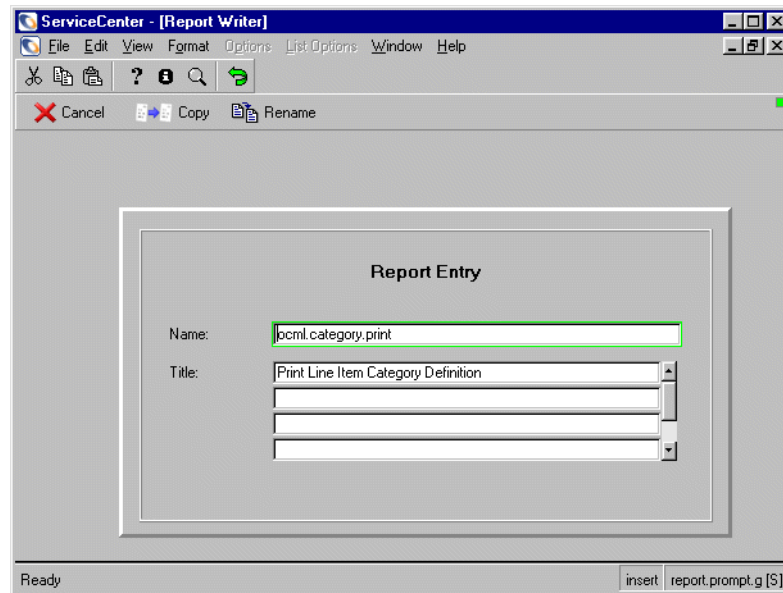


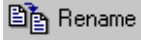
Figure 10-28: Using Report Entry to copy and rename reports

- 4 Enter the *new* name of the report over the existing name in the Name field. This must be a unique name.
- 5 Enter the *new* report title in the Title field.
- 6 Continue with one of the following options.



Copy

- Click **Copy** to create a second copy of the report using a different name. This copies all the Report Writer definitions from the current record to the new report name. If any changes were made to the report title, the edited version is copied to the new report. The existing Report Writer definition will not be changed during the copy process. The formats defined in the original report will not be copied. However, formats can be shared between reports.



Rename

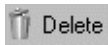
- Click **Rename** to rename the current report. This renames the report, but only the report name and report title are changed. All other definitions for the report remain unchanged.
- Click **Cancel** or press F3 to cancel the procedure and return to the *report.maint* form. Since the procedure has been canceled, the original report remains unchanged.

How to Delete Reports

To delete reports:

- 1 Access a report, following the procedures discussed in *How to Access Report Writer* on page 250.

In this example, the *ocml.category.print* report is opened.



Delete

- 2 Click **Delete** or press F4 to delete the report.
The Report Deletion (*report.prompt.delete*) form is displayed (Figure 10-29 on page 328).
- 3 Continue with one of the following Delete options.
 - Click **Abort Del** to end the delete process and exit to the *report* form.
 - Click **Rpt Record** to delete only the *report* record.
 - Click **Rpt & Fmts** to delete the *report* record and its *report* formats.
 - Click **All** to delete the *report* record, *report* formats, and all associated application panels. This option is only available for *manual reports*.

Note: When you delete report formats, be sure that the formats defined in this report are not required in any other report that is used.

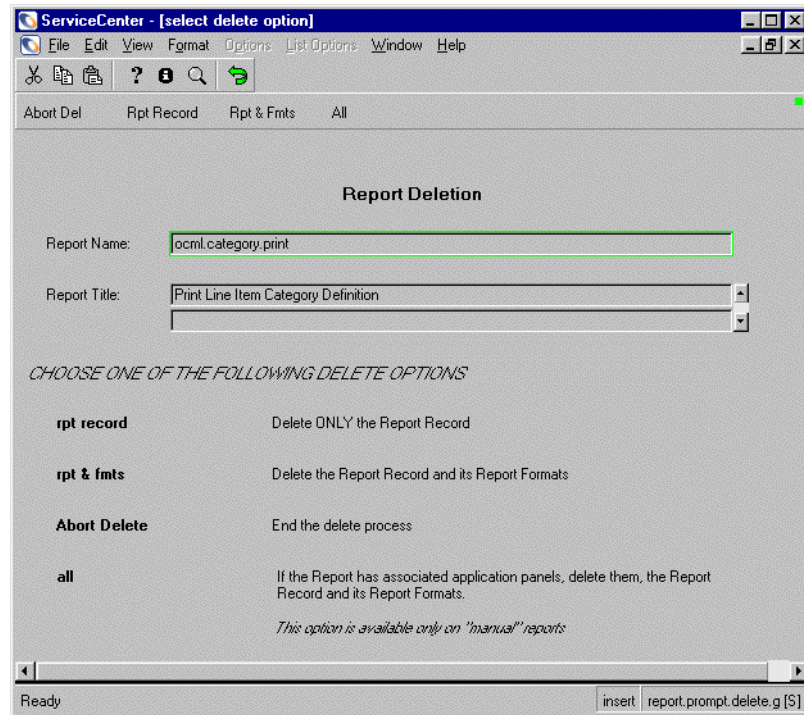


Figure 10-29: Report Deletion form with Delete options

How to Use Stacked Queries for Sorting

Report Writer bases report generation on the records retrieved from the primary file by the definition of the *primary query* or the definition of the *stacked queries*. Different rules apply to each of these situations and each has advantages and disadvantages over the other.

If one *primary query* with no *stacked queries* is defined to retrieve records, then the following rules are in effect.

- A work file is *not* created.
- For optimum performance, the *primary query* must be fully keyed.
- If sort fields are defined, then that key (and that key *only*) is used to retrieve records from the database.

Note: If the *primary query* is *not* fully keyed *based on the sort key*, turnaround time will be degraded.

For example, the following keys are defined for a file:

```
category
status
open.time
assignment
category
```

and the sort fields for a report are:

```
open.time
assignment
category
```

then the query:

```
category="systems"
and
status< >="closed" (OS/390)
or
status~="closed" (Unix)
```

This is *not* fully keyed, because the sort fields will force the system to use the second key to retrieve the records. Therefore, the query that is defined is partially keyed and each record retrieved from the database (in this case all records appearing in the key list for the second key) have to be expanded to determine if the data record meets the search requirement.

Primary Query Option

Advantage of Using a Primary Query Over Stacked Queries

Choosing to use the Primary Query option over the Stacked Queries option is advantageous under the following circumstances.

- A query of *true* is defined and either:
 - No sort fields are defined (in this case the first key in the key list would be used to retrieve the data records).
 - or-
 - The sort fields are defined as a key.
- The primary query and the sort fields reference the same key and the primary query is fully keyed.
- The primary query is fully keyed and no sort fields are defined.

Stacked Queries Option

Advantage of Using Stacked Queries

Choosing to use the Stacked Queries option is advantageous under the following circumstances.

- The sorting requirements for a report are not defined as a key in the primary file's dbdict.
- The primary query does not reference the same key as the sort fields.
- The reporting requirements are such that one fully keyed query cannot be defined.

Example of Stacked Queries Option

An example of how to use the Stacked Queries option is as follows.

Example: The requirements of this report are to list all open Incident tickets for the categories of **lines**, **modems**, and **enduser**. The sort requirements are by **category**, **assignment**, **status**, and **open.time**. The primary file is **probsummary** and the following keys are defined:

```

number
*****

flag
category
subcategory
company
*****

logical.name
*****

```

The following specifications would be in effect for the report if the Stacked Queries option was used.

Stacked Queries Specifications

Unique sort sequence.

primary file: probsummary

primary query: true

stacked queries: category="lines" and
status="closed"

category="modems" and
status="closed"

category="enduser" and
status="closed"

Note that each of the queries is fully keyed. While this is not a requirement, it is strongly recommended that the queries be defined for stacked queries in a way that allows the system to meet all or most of the query requirements from a key list rather than having to expand records retrieved from a particular key to satisfy the query requirements.

Sort Fields

category
assignment
status
open.time

Note: These sort fields are *not* defined as a key in the primary file, but they will be defined as a key in the new temporary file.

If the report was run at approximately 15:00 hours on 02/07/02, the name of the work file would be created with a date stamp:

workprobsummary0207021500

Rules to Follow When Defining Stacked Queries

If one or more stacked queries are defined for a report, the following rules are in effect.

Note: Stacked queries are separate and distinct from secondary queries.

- A work file *is* created.
- Any Format Control definitions specified for any formats used in the report are still in effect.
- Any subtotalling requirements defined for the report are still in effect.
- The sort fields do not have to be defined as a key in the primary file's dbdict, but *must* be defined as fields.
- If structure names are used in one sort field, structure names must be used in all sort fields.
- The following validations are performed on all sort fields (any errors are logged in the `msglog` file):
 - The structure name (if any) referenced in each sort field must be valid.
 - The sort field must belong to the indicated structure (if one is specified).
 - The sort field must be a **scalar number, character, logical, or date/time** field. *No other sort field types are allowed.* If other types of fields are specified, the sort will be aborted.
- Each stacked query is executed against the primary file and all records are temporarily stored in the `sortwork` file.

- The primary query is executed against the **sortwork** file and under normal circumstances is defined as *true*.
- The **work** file that is created is a logical file defined with the following naming standard:
workxxxmddyhhmss

Naming Standard Definitions

work =	Constant
xxx =	The name of the primary file
mmddyhhmss =	The date and time that the work file was created.

- The **work** file is defined in physical file FILE3. Therefore, caution should be exercised when using stacked queries which will cause all records from any large file to be read and placed in the **sortwork** file.
- The **work** file will be scheduled for deletion two hours after its creation with a class of **report** and an operator name of **sortsubr**. It is recommended that the system administrator periodically review the dbdict file to ensure that all work files have been deleted. Be sure to only delete those work files that completely follow the naming standards of the sort work files.
- The sort routine copies all unique keys from the primary file to the **sortwork** file and adds the sort fields as **nulls&duplicates** key. All **no nulls** and **no duplicates** keys that display in the key list before the first unique key are copied to the **sortwork** file. If **no unique**, **no nulls**, or **no duplicates** keys exist, the first key in the key list is copied to the **sortwork** file.
- If the same record is retrieved using different queries and at least one unique key is defined for the **sortwork** file, the duplicate record is not added to the **sortwork** file.
- All error messages issued from the sort routine are logged in the **msglog** file.
- A message stating how many records were added to the **sortwork** file, based on the execution of each stacked query, is issued to the **msglog** file.

Report Writer Options Menu Selections

The following options are available within the various Report Writer forms.

Option	Description
Add	Allows you to modify selected records in a file and add them to the database, while leaving the original records in place.
Copy/Rename	Allows you to create a new record by copying an existing record and renaming the new record.
Delete	Allows you to delete all the records appearing in a record (or QBE) list. Query for a certain subset of records and select mass delete. You are prompted to confirm the action.
Export to Excel	Exports all the records displayed in the record or QBE list directly to Microsoft Excel. Basically, you can export information to any product that has DDE (Dynamic Data Exchange) support. This DDE function requires Excel 95 or later. Note: This option is available only on Windows clients.
Export to Text File	Exports all the records displayed in the record or QBE list to an external file in text format. Choose a file name and text delimiter in the dialog box displayed. The file is exported to the RUN directory in your client installation. This option is not part of the DDE support and can be run from clients other than those operating in Windows.
Forms Designer	Allows you to go from the <i>report</i> form into the Forms Designer utility to change the format of a new or existing report.
Maintenance Schedule	Allows you to modify the current schedule for this report.
Mass Unload	Allows you to unload all the records displayed in a record or QBE list to an external file.
Modify Columns	Allows you to edit column headings in a record or QBE list.
Print	Allows you to print either the record/QBE list or all the records.
Save As Inbox	Allows you to save the current list as an inbox.
Set Query	Brings you to the <i>report.query.maint</i> form to add a query or modify an existing query.

Option	Description
Totals	Brings you to the <i>report.totals</i> form of the existing report.
Update	Allows you to update the current record.
View Spooled Reports	Allows you to view spooled reports. When you select to View Spooled Reports , the <i>spoolheader</i> form is opened. From here you click Find to search for records.

11 | Benchmark Utility

CHAPTER

Overview

The ServiceCenter Benchmark Utility allows you to simulate high user activity loads on a particular ServiceCenter system, so that you can:

- Gauge the limitations of the hardware on which the system is running.
- Determine whether or not the system can handle your expectations regarding how many users can be on the system while retaining acceptable response times.

The Benchmark Utility launches a predetermined number of custom sample users on a designated system. These “users” will perform a specific number of Incident Management ticket opens, updates, closes, and queries in a specified period of time. The resulting performance information (including elapsed time, CPU time, and number of string copies) for each of these actions is recorded for later evaluation.

System Requirements

To use the ServiceCenter Benchmark Utility, you must be running a ServiceCenter system with a run-time environment (RTE) release of 3.0 SP2 or later. In addition, you must either be running an application release of A9902 or later, or load the *benchmark.unl* file supplied by Peregrine Systems.

Preparing to Run the Benchmark Utility

This section outlines the tasks you will need to perform before running the Benchmark Utility. These tasks include, but are not limited to:

- Updating the *sc.ini* file
- Creating ServiceCenter operators for the Benchmark utility to use
- Creating Benchmark configuration records for each operator

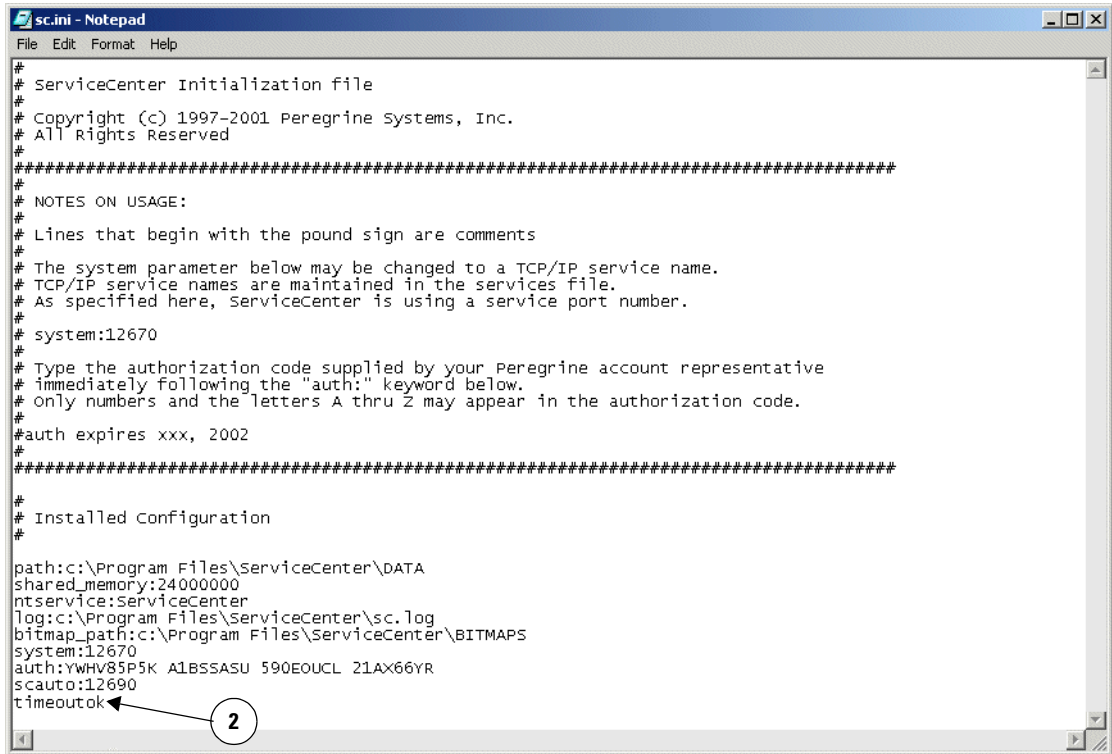
Updating the *sc.ini* File

The *sc.ini* file on the ServiceCenter Server must be revised to include the parameter *timeoutok*.

To revise the *sc.ini* file:

- 1 Open the *sc.ini* file in Notepad or any text editor.
- 2 Type *timeoutok* on a separate line in the file.

3 Select File > Save.



```

sc.ini - Notepad
File Edit Format Help
#
# ServiceCenter Initialization file
#
# Copyright (c) 1997-2001 Peregrine Systems, Inc.
# All Rights Reserved
#
#####
#
# NOTES ON USAGE:
#
# Lines that begin with the pound sign are comments
#
# The system parameter below may be changed to a TCP/IP service name.
# TCP/IP service names are maintained in the services file.
# As specified here, ServiceCenter is using a service port number.
#
# system:12670
#
# Type the authorization code supplied by your Peregrine account representative
# immediately following the "auth:" keyword below.
# Only numbers and the letters A thru Z may appear in the authorization code.
#
#auth expires xxx, 2002
#
#####
#
# Installed Configuration
#
path:c:\Program Files\ServiceCenter\DATA
shared_memory:24000000
ntservice:ServiceCenter
log:c:\Program Files\ServiceCenter\sc.log
bitmap_path:c:\Program Files\ServiceCenter\BITMAPS
system:12670
auth:ywHV85P5K A1BSSASU 590EOUCL 21AX66YR
scauto:12690
timeoutok

```

Figure 11-1: sc.ini file

Creating ServiceCenter Operators

You will need to create one or more operators in the ServiceCenter system where you will be running the Benchmark test.

Use the following guidelines to create these operators:

- The user account must be set up without a password.
- The user must have all capability words needed to open, update, close, and query records in Incident Management.
- The user's initial application must be benchmark.run.

For this example, the operator name is JOE.ADMIN.

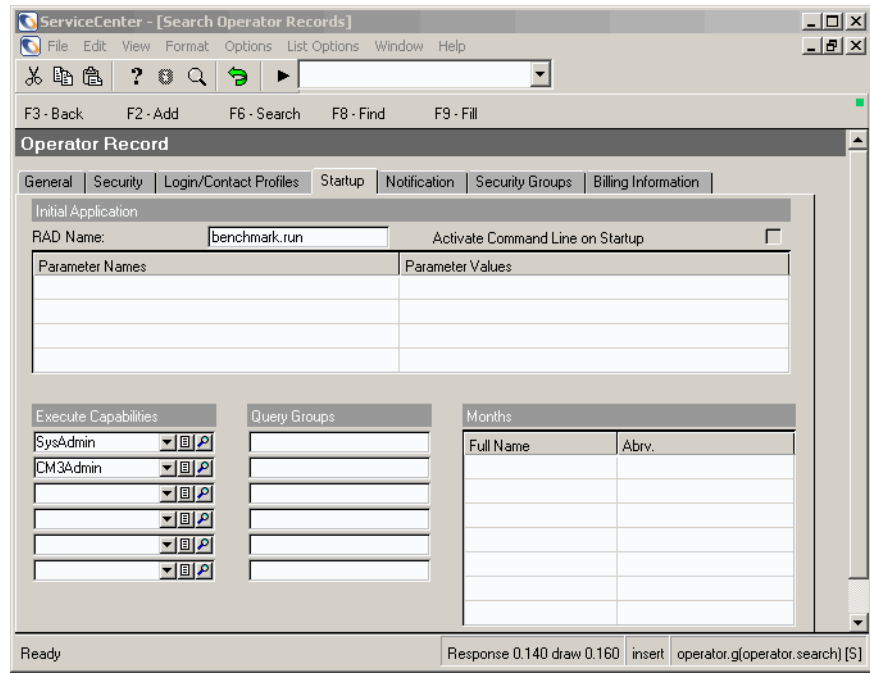


Figure 11-2: Operator file with initial application designated

Although only one user is needed to run the utility, multiple operators can be created to simulate different user types. For example, one user can be configured to open more tickets than others, while another may only perform queries.

Accessing the Benchmark Utility Menu

Once the operator records have been created, all other options that you need to configure and run the Utility, and to view the test results, are available from the Benchmark Utility Menu.

To access the Benchmark Utility Menu in Text Mode:

- 1 Type `*mBENCHMARKTXT` on the command line.
- 2 Press **Enter**.

To access the Benchmark Utility Menu in Windows Mode:

- 1 In the system administrator's home menu, select the Utilities tab.
- 2 Click **Maintenance**.

The Maintenance menu is displayed.

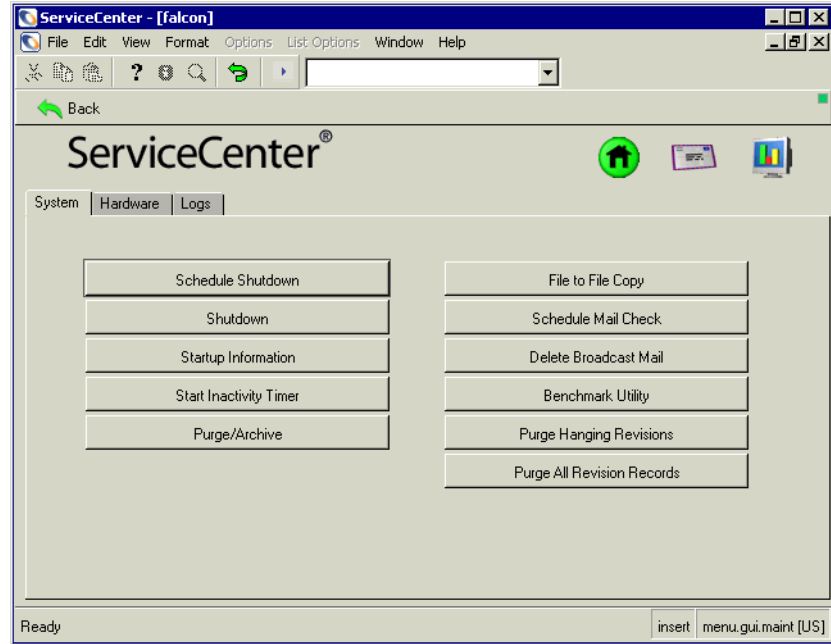


Figure 11-3: System tab of the Maintenance menu

- 3 Click **Benchmark Utility**.

The Benchmark Utility Menu is displayed.

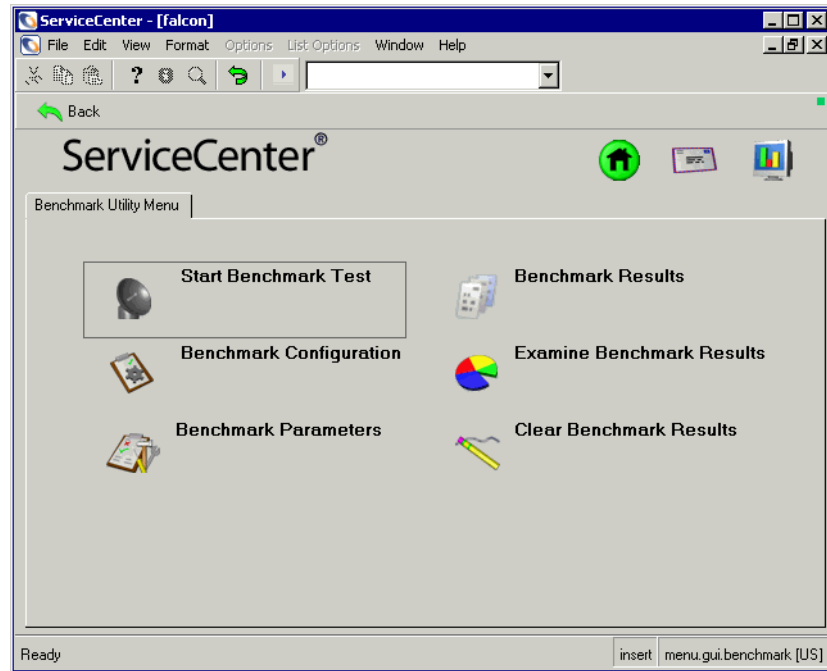


Figure 11-4: Benchmark Utility Menu

Creating Benchmark Configuration Records

Each user you create will run a specific number of Incident Management opens, updates, closes, and queries in a specified period of time. To accomplish this, a Benchmark Configuration record must be created for each user.

To create Benchmark configuration records:

- 1 On the Benchmark Utility Menu, click **Benchmark Configuration**.
- 2 Fill out each of the fields, referring to *Fields on the Benchmark Utility Start Form* on page 345 for a description of each field.

Note: When you click in a field in the Transaction List column, a drop-down list of possible transaction types is displayed.

Field	Description
Operator Name	The name of the operator. In Windows mode, all available benchmark operators will appear in the drop-down list next to the Operator Name field.
Time Period of One Cycle	The amount of time in which one set of operations will be performed. This field should contain a relative time (for example, 01:00:00).
Cycles to Run	The number of cycles to perform before the operator is logged out.
Transaction List	The types of transactions the operator will perform. There must be at least one transaction type entered. The possible types are imopen, imupdate, imclose, and imquery.
# of Transactions	The number of transactions to run for the specified transaction type. The number of imupdate and imclose transactions must be equal to or less than the number of imopen transactions.

3 Click Add.

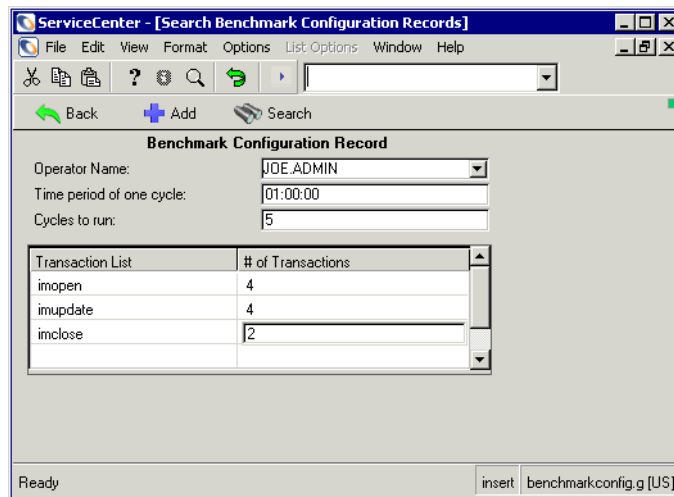


Figure 11-5: Configuration record

Starting a Benchmark Test

To start a Benchmark test:

- 1 On the Benchmark Utility Menu, click Start Benchmark Test.

A dialog box is displayed, which asks for information the utility needs to begin the test. See *Fields on the Benchmark Utility Start Form* on page 345 for a description of each of the fields in this dialog box.

Note: The # of Connections and Login ID fields are required fields.

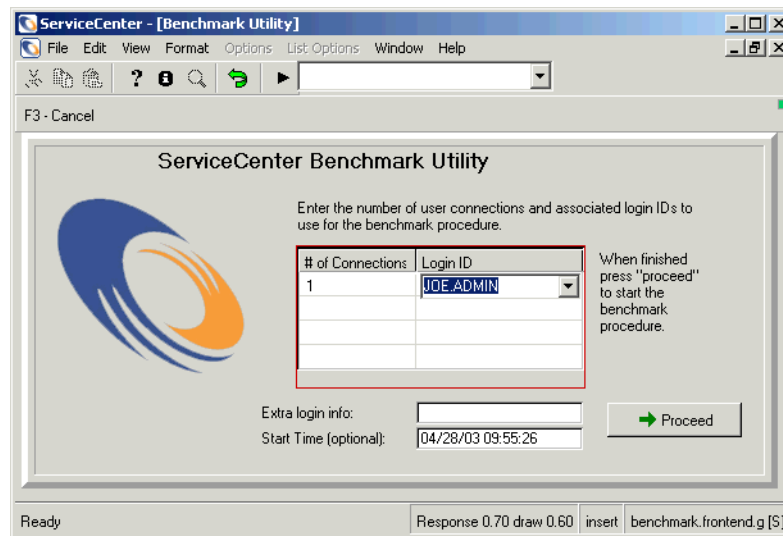


Figure 11-6: ServiceCenter Benchmark Utility

- 2 After you have finished filling out the fields, click **Proceed**.

A prompt is displayed, which asks if you want to clear all existing Benchmark information from the current system.

- 3 Click Yes to proceed.

The system will clear any existing Benchmark data from previous tests. It will then begin launching the Benchmark processes and logging on the users.

A prompt is displayed when the process is complete.

- 4 Click **OK**.

The Benchmark Utility will now process each transaction as set up in each user's configuration record.

Fields on the Benchmark Utility Start Form

Field	Description
# of Connections	The number of connections to start for a specific operator.
Login ID	The operator name (from the Benchmark Configuration database) for which the system will start the specified number of sessions.
Extra Login Info	<p>Any specific login information that must be added to a standard command line to connect to the correct ServiceCenter system. This should be used if the default <i>sc.ini</i> file on the user's machine does not point to the correct server.</p> <p>Note: The system will attempt to start a session from the same place as the current session was started (for example, <code>-system:name.12670</code>).</p>
Start Time (<i>optional</i>)	<p>The time at which the Benchmark operators will begin performing their transactions. If this time is not specified, the users begin their transactions immediately upon login. There is a programmed one second delay between starting each user session, so it may be in the tester's best interest to delay the actual start of the transactions so that all users can finish logging onto the system before starting the actual transactions.</p> <p>Once a start time has been set, any additional Benchmark processes started from other machines will automatically use the previously set start time. This allows you to simulate tests using multiple machines while synchronizing the transactions on all machines.</p>

While the test is running, you can monitor the progress of the test. Or you can wait until the test is complete to view the test results. The following sections describe these procedures.

Monitoring the Progress of the Test

You can check on the progress of the Benchmark test from three locations while the test is running:

- ServiceCenter system monitor
- Windows Task Manager (NT only)
- Benchmark Utility Results form

To use the ServiceCenter system monitor:



- ▶ In the upper-right corner of the **Benchmark Utility Menu**, click **System Status**.

A list of all users currently logged into the system is displayed (Figure 11-7 on page 346). *FALCON*, the administrator, is running the test.

Command	User Name	PID	Device ID	Login Time	Idle Time
	CLIENT-12670	944	SYSTEM	04/28/03 09:50:46	00:00:01
	spool	1376	SYSTEM	04/28/03 09:50:47	00:01:32
	CLIENT-12690	1492	SYSTEM	04/28/03 09:50:48	00:00:02
	report	1524	SYSTEM	04/28/03 09:50:48	00:00:27
	problem	1432	SYSTEM	04/28/03 09:50:49	00:00:28
	change	1508	SYSTEM	04/28/03 09:50:50	00:00:28
	sla	1540	SYSTEM	04/28/03 09:50:52	00:00:27
	agent	1448	SYSTEM	04/28/03 09:50:53	00:00:23
	falcon	1548	Express-Windows 2000 Professional	04/28/03 09:50:53	00:00:00
	marquee	1580	SYSTEM	04/28/03 09:50:54	00:00:23
	lister	1596	SYSTEM	04/28/03 09:50:55	00:00:21
	linker	1572	SYSTEM	04/28/03 09:50:56	00:00:20
	event	1608	SYSTEM	04/28/03 09:50:57	00:00:22
	availability	1612	SYSTEM	04/28/03 09:50:58	00:00:21
	contract	1620	SYSTEM	04/28/03 09:50:59	00:00:19
	ocm	1624	SYSTEM	04/28/03 09:51:00	00:00:18
	alert	1712	SYSTEM	04/28/03 09:51:02	00:00:17
	sync	1672	SYSTEM	04/28/03 09:51:03	00:00:17
	JOE.ADMIN	1804	SYSTEM	04/28/03 09:56:56	00:00:24

Figure 11-7: System Status window

To use the Windows Task Manager:

- 1 Right-click your system task bar, and select **Task Manager** from the menu.
- 2 Click the **Performance** tab to view CPU usage information.

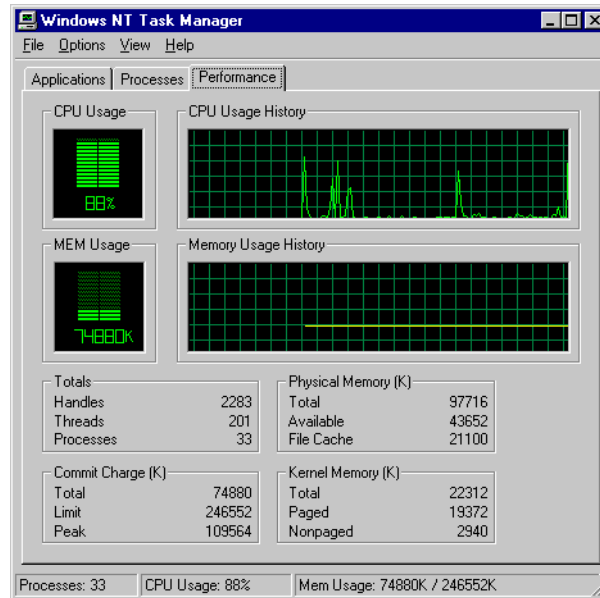


Figure 11-8: Windows NT Task Manager—CPU performance

Viewing the Benchmark Test Results

Once the Benchmark Utility has finished processing all of the user tasks, you can view the results of the test using the Benchmark Utility. The test results are presented in three formats:

Format	Description
Elapsed time	Average, minimum, maximum, and total time (in seconds) for each transaction.
CPU time	CPU load for each transaction, including the average, minimum, maximum, and total time (in seconds).
String copies	Average, minimum, maximum, and total number of string copy operations performed. This data indicates how much work the application did during the transactions, because almost all RAD statements generate one or more string copy operations.

You can access the test results in two ways:

- Use Examine Benchmark Results to view complete test results or portions of the results by date and time.
- Use Benchmark Results to view specific data based on parameters you define.

To use Examine Benchmark Results:

- 1 In the Benchmark Utility Menu, click Examine Benchmark Results. A dialog box (the *benchmark.avg.prompt* form) is displayed.

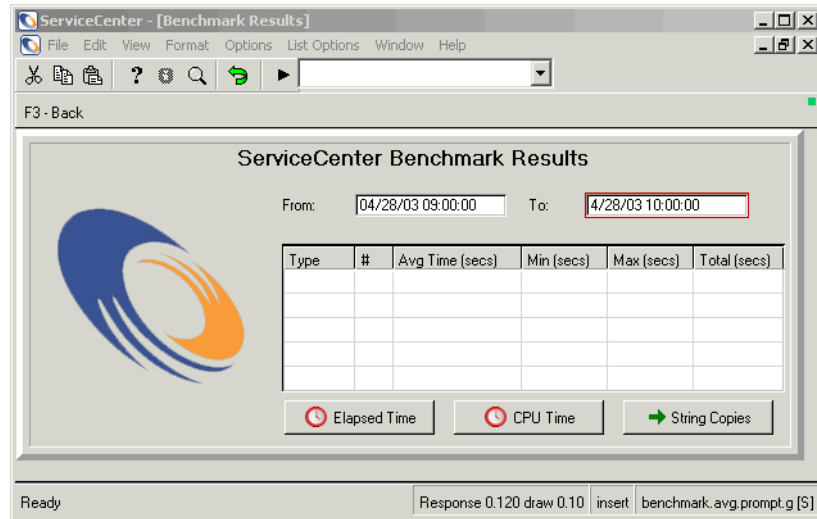


Figure 11-9: Benchmark Results form

- 2 If the test is currently running, just add the start time in the **From** field (in the format of: mm/dd/yy hh/mm/ss), and the statistics will be displayed. Leave these fields blank to view all of the test results.
- 3 Click **Elapsed Time**, **CPU Time**, or **String Copies** to select the particular data you want to view.

The Benchmark Results fields are filled with the requested data.

You can further define the data you want to view by using the Benchmark Results form. The Results form allows you to view the results of each transaction and see trends or changes in performance between individual transactions.

To use Benchmark Results:

- 1 On the Benchmark Utility Menu, click **Benchmark Results**.
- 2 In the form displayed, fill in the fields to select particular data to view.

For example, you might want to see all tickets opened by operator *JOE.ADMIN*.

- a Type *imopen* in the **Transaction Type** field.
- b Type *JOE.ADMIN* in the **Operator ID** field.
- c Click **Search**.

A record list of all the *imopen* transactions performed by user *JOE.ADMIN* is displayed (Figure 11-10 on page 350). The first record in the list is displayed in the form. You can use the data in the record list to compare elapsed times between transactions.

The screenshot shows a window titled "ServiceCenter - [Benchmark]". The window contains a search form with the following fields and values:

Transaction Type:	imopen
Operator ID:	JOE.ADMIN
Date/Time:	04/28/03 09:57:50
Elapsed Time:	0.871
ID:	IM10001
PID:	1804
Users on System:	
CPU Seconds:	0.471
String Copies:	0
Bytes Sent:	0
Bytes Received:	0

At the bottom of the window, the status bar shows: "Ready" on the left and "Response 0.70 draw 0.30 insert benchmark(db.view) [S]" on the right.

Figure 11-10: imopen results

Clearing Test Results

To clear the results of a Benchmark test:

- 1 On the **Benchmark Utility Menu**, click **Clear Benchmark Results**.
A prompt is displayed to confirm that you want to clear the test results.
- 2 Click **Yes** to confirm the request.

Stopping a Test



System Monitor

To stop a running test:

- 1 Click the **System Status** utility button on the menu.
- 2 Click **System Monitor** to access the process.
- 3 Place the cursor in the Command column next to the listed task.
 - Type **s** next to the task to view the process and schedule record details.
 - Type **k** next to the task to kill that user's session or task.
- 4 Click **Execute Commands**.
 - If **s** is executed, a form is displayed providing the status of the item.
 - If **k** is executed, a confirmation message is displayed in the status bar.

A Keyboard Shortcuts

APPENDIX

This appendix provides a list of keyboard shortcuts for certain functions in ServiceCenter:

Command	Action
Shift—Ctrl+B	Displays button IDs in the pull-down Options menu.
Shift—Ctrl+C	Displays any field captions that are not shown on the screen, because of the increased font size.
Shift—Ctrl+D then Shift—Ctrl+E	Activates the RAD debugger.
Shift—Ctrl+F	Changes the color of the focus frame around buttons and fields. Place your cursor in a field, and then continue to press SHIFT—CTRL+F to cycle through your color choices until you find the color you want to use for the focus frame.
Shift—Ctrl+I	Toggles between tabs within the current ServiceCenter menu.
Shift—Ctrl+L	Sets the cursor focus to the command line prompt.
Shift—Ctrl+N	Returns the screen to its default width, narrowing the font size by 25% for English screens and 50% for all other languages.
Shift—Ctrl+P	Toggles between server and client printing. The symbols [P] (client) and [S] (server) appear in the lower-right corner of the screen following the form name.

Command	Action
Shift—Ctrl+R	<p>Toggles the Response Time Monitor (RTM) parameter and activates the statistics in the status bar of the current screen.</p> <ul style="list-style-type: none"> ■ Response: This is the response time of the server and the time spent executing the RAD code when the system leaves one screen and enters another. ■ Draw: This is the time the system takes to redraw the screen when changing screens. This does not include network time.
Shift—Ctrl+S	<p>Performs a 30-second network test to determine the approximate number of transactions per second and bandwidth of the network connection between the client and the ServiceCenter host. The test consists of 15 one-second tests, sending and receiving 8,000 bytes of data. At the end of each second, the status bar is updated with the number of bytes transferred.</p> <p>When the test is complete, the following message is displayed in the status bar: <i>Client Server transmissions per second <number of transmissions> approximate bandwidth <number of bytes transferred></i>.</p> <p>Note: For optimal performance, the number of transmissions should be above 250 and the number of bytes above 1 megabyte.</p>
Ctrl+Tab	Rotates among all open windows in the current session.
Shift—Ctrl+W	<p>Widens the screen to accommodate the proportions of different languages. In the basic system, English screens are widened by 25%. All other languages are widened to 50%. When you select a language at login, the screen automatically adjusts to that language's default screen width.</p>
Shift—Ctrl+Z	<p>Note: Brings up a separate window with client statistics (Figure A-1 on page 355). Settings from the ServiceCenter sc.ini file are displayed, as is the ServiceCenter logfile. If the log is large and/or there is not enough memory on the machine, it could take longer than expected to display.</p>

```

ServiceCenter - Client Information Popup Window - [Read Only]
File Edit
Log file is D:\Program Files\SC5.1\ServiceCenter\sc.log
Log file for this session

-----Logfile opened-----

1280 01/06/2003 15:51:55 Process scguiw32 5.1.dev (0138) System: minerdev2k.12670 (0x61e13c00)
1280 01/06/2003 15:51:55 on PC running Windows 2000 Professional (5.0 Build 2195) from minerdev2k
(10.2.3.55)
1280 01/06/2003 15:51:55 connected to express server at 10.2.3.55:12670
1280 01/06/2003 15:51:58 Initialized (ServiceCenter) as a DDE client and server
16 started on socket 592.
1616 01/06/2003 15:51:55 Attaching to resources with key 0x61e13c00
1616 01/06/2003 15:51:56 Process scenter 5.1.dev (0156) System: 12670 (0x61e13c00)
1616 01/06/2003 15:51:56 on PC running Windows 2000 Professional (5.0 Build 2195) from minerdev2k
(10.2.3.55)
1616 01/06/2003 15:51:56 connected to express client scguiw32 5.1.dev at 10.2.3.55:1255
1616 01/06/2003 15:52:06 User falcon has logged in and is using a Concurrent license (1 out of a maximum 10)
1616 01/06/2003 15:52:21 System being shutdown by falcon from Express-Windows 2000 Professional
1616 01/06/2003 15:52:21 User falcon has requested a system shutdown
1616 01/06/2003 15:52:21 Requested kill for pid(1508) being issued by pid(1616)
1508 01/06/2003 15:52:21 Kill signal received
1508 01/06/2003 15:52:21 Process termination in progress
1616 01/06/2003 15:52:21 Waiting for 1 users to terminate
1508 01/06/2003 15:52:25 Express listener shutting down
1508 01/06/2003 15:52:25 Process termination in progress
1508 01/06/2003 15:52:25 Process termination in progress
1508 01/06/2003 15:52:25 xprt_send: Error sending data over TCP connection., socket operation on
non-socket, errno(10038)
1508 01/06/2003 15:52:25 xvmsg_send: Send failed, errno = 0
1508 01/06/2003 15:52:25 Process termination cleanup complete for pid(1508)
1616 01/06/2003 15:52:26 ----- Cache Statistics -----

```

Figure A-1: Client information from the log file

Index

Symbols

*SYSDEFAULTS record 232–234

A

accessing Forms Designer 276
Add Format Control option 192
Add/Replace Mode 185
archive
 records in foreground 38
 using Database Dictional Copy/Rename option 54
array type 186
ASCII 0a delimiter 139

B

Benchmark Utility
 accessing the menu (text) 340
 accessing the menu (Windows) 341
 clearing test results 351
 configuration records, creating 342
 creating ServiceCenter operators 339
 monitoring the test progress 346
 using System Status 346
 using Windows Task Manager 347
 preparing to run the Utility 338
 sc.ini file, updating 338
 starting a test 344
 system requirements 338
 timeoutok parameter 338
 user account guidelines 339

viewing test results 348
 using Benchmark Results 350
 using Examine Benchmark Results 349

C

calculations 166
Change Management
 data mapped files 53
 Purge/Archive 52–53
company record
 LDAP login template 230–231
 template, defining 212
confirmation screen 57
copying
 archiving files 54
 reports 326–327

D

data
 sample 10
data map 38, 50, 56, 63
 accessing 57–63
 field descriptions 60–63
 files in pre-built maps 53
 map lists 56–57
 Type 0 63, 73, 76, 78, 82, 84
 Type 1 64, 70, 72, 74, 75
 Type 2 65
Unload Script Utility 18, 19, 21, 23
unload.map script 89, 90

- using with Incident Management 53
- using with network management applications 52–53
- data.map application 67
- database
 - loading 127–136
 - background 134–136
 - foreground 128–134
 - messages 131–134
 - unloading records 93–127
 - Database Dictionary record (no data) 117–119
 - form field definitions 101–103
 - messages 98–100
 - multiple databases 111–115
 - multiple records from one database 103–105
 - multiple records from single file (value list) 105–110
 - QBE only to Excel 115–116
 - Report Writer report 119–127
 - single records 95–103
- database unload 67
- date format 151, 186
- deleting
 - Report Writer reports 327

E

- education services 12
- error messages
 - load manager 247
- export descriptor record
 - creating 148–153
 - modifying 152–153
 - printing 161–162
- exporting
 - accessing export utility 140–148
 - export descriptor record
 - creating 148–153
 - modifying 152–153
 - printing 161–162
 - using Format Control with 162–168
 - export utility fields 145–146
 - overview 138–139
- records
 - in background 159–161
 - in foreground 153–158
 - rules used 139–140

F

- field names
 - Add to Map List 63
 - Array? 61
 - Arrayed Structure? 61
 - Field Name 61
 - Filename 60
 - Map Cond 63
 - Type 61
- field.names
 - Desc 60
- fields
 - LDAP mapping 213
 - names
 - Map Cond 91
 - Unload Script Utility 17–18
- filename 57
- files
 - commonly used in Unload Script Utility 20–21
 - Database Dictionary copy/rename option 54
 - LDAP mapping 213
 - loading
 - background 134–136
 - foreground 128–134
 - Report Writer
 - spoolheader and spool files 320
 - unloading 93
 - Database Dictionary only 117–119
 - multiple records from one database 103–105
 - multiple records using value list 105–110
 - QBE list to Excel 115–116
 - records from multiple databases 111–115
 - Report Writer reports 119–127
 - single record 95–103

- Format Control
 - accessing
 - from Report Writer 290–291
 - export utility, use with 162–168
 - import utility, use with 185, 192
 - processes 167–168
 - subheaders 293
 - used in a matrix report 301
 - formats
 - Report Writer 275
 - forms
 - protecting in Unload Script Utility 21–23
 - Forms Designer
 - accessing 276
- I**
- import descriptor record 185
 - deleting 188
 - Format Control, use with 192
 - modifying 183–184
 - printing 191
 - import function
 - understanding 172
 - importing records
 - background 189
 - foreground 188
 - Incident Management
 - data mapped files 53
 - Purge/Archive 52–53
 - Purge/Archive with data map 53
 - initialization (sc.ini) file
 - alternate LDAP servers 236
 - modifying for telephony 201
 - initialization expressions 164
 - initialization file
 - sc.ini 354, 355
 - installation
 - load manager 242–245
- L**
- LDAP
 - *SYSDEFAULTS record 232–234
 - administrator capabilities 234–235
 - base attribute string 221
 - base directory 213, 218, 221, 226
 - creating records from LDAP data 225–234
 - data source selection 222, 227, 234
 - defined 211
 - directory 222
 - distinguished name (DN) 213, 226
 - initialization parameters
 - ldapdisable:1 237
 - ldapsrv 236
 - ldapauthbase field 226
 - mapping
 - fields 213
 - files 213
 - multiple base directories 213
 - MVS 211
 - operator record 227, 230
 - passwords, authenticating 226
 - primary data source controls 227, 234
 - queries 222
 - sldapbase field 213
 - sldapconfig file 215
 - server
 - alternates defined 236
 - defining 215–219
 - disabling 237
 - host address 218, 220, 236
 - port address 218, 221, 236
 - server interface, overview 211–213
 - template 212, 227–231
 - applying to company record 230–231
 - creating 227–230
 - viewing 227, 234
 - updating mapped fields 213
 - ldapauthbase 234
 - load manager
 - architecture 241
 - documentation, related 240
 - error messages 247
 - function described 241
 - installation 242–245
 - overview 239–240
 - prerequisites 240
 - system requirements 240
 - using 242, 246

- loading
 - databases 127–136
 - background 134–136
 - foreground 128–134
- log files
 - displaying files with key commands 354, 355
 - load manager error messages 247
- lower case translation 186
- M**
- map lists 56–57
- messages
 - database unload 98–100
 - loading 131–134
 - unload complications 97–100
- Microsoft
 - TAPI 2.1 white paper 197
- MVS
 - LDAP compatibility 211
- N**
- network management applications
 - data mapping with 52–53
- O**
- operator record
 - LDAP
 - administrative rights 234
 - directory structure, defining 227
 - login template 230
- operators (symbols)
 - Unload Script Utility 18
- override query 64
- P**
- parameters
 - Built Array of Files 68
 - Built Array of Queries 68
 - Current Data Record File Var 68
 - Last Mod Date 68
 - ldapdisable:1 237
 - ldapsrvr 236
 - Override Map Name 68
 - Prompt Operator 68
- password
 - LDAP, authenticating 226
- phone options
 - computer-phone integration 195
- printing
 - key commands for client/server printing 353
- Purge/Archive 56, 67
 - accessing 36–38
 - network management applications 52–53
 - pre-defined data map files 53
 - purging records in foreground 46
 - scheduling in background 47–52
 - understanding 35
- purging
 - records in foreground 44
- Q**
- QBE list
 - query method 106
- QSAM record 139
- queries 57
 - LDAP 222
 - Report Writer 266
 - define stacked queries 332
 - primary query option 330
 - stacked queries option 330
 - use formats 268
 - use stacked queries 328
 - Unload Script Utility 18–19
- R**
- records
 - primary 56
 - purging 32–33
 - Report Writer
 - view schedule records 325
 - secondary 56
 - unloading 28–31
- Report Writer 249, 250
 - accessing 250
 - copy/rename option 326–327
 - Format Control, accessing 290–291
 - initial formats 275
 - options menu 334

- queries
 - primary query option 330
 - stacked queries option 330
 - stacked, define 332
 - use stacked queries 328
- querying 266
- report formats
 - creating and editing 264–266
- reports 293, 301
 - access report totals 278
 - delete 327
 - detail formats 298
 - grand totals 285
 - online viewing 320
 - page breaks 278
 - process in background 312
 - report totals 278
 - run in the foreground 311
 - schedule for background process 313
 - spoolheader and spool files 320
 - subtotal format 304
 - summary totals 285
 - total format 305
 - using calculations in 288–290
 - using Format Control 292
 - using simple subtotals 283
 - using subtotals 285
 - view schedule records 325
- use query formats 268
- reports
 - creating and editing Report Writer formats 264–266
 - Report Writer 301
 - access report totals 278
 - delete reports 327
 - detail formats 298
 - grand totals 285
 - online viewing 320
 - process in background 312
 - run reports in the foreground 311
 - schedule for background process 313
 - spoolheader and spool files 320
 - subtotal format 304
 - summary totals 285
 - total format 305
 - using 293
 - using 293
 - using Format Control 292
 - using simple subtotals 283
 - using subtotals 285
 - view schedule records 325
- Request Management
 - data mapped files 53
 - Purge/Archive 52–53
- S**
- sc.ini file
 - alternate LDAP servers 236
 - Benchmark Utility 338
 - displaying with key commands 354, 355
 - editing for TAPI 202
 - LDAP parameters
 - ldapdisable:1 237
 - ldapserver 236
 - modifying for telephony 201
- sldapconfig file 236
- scripting
 - unload scripts
 - creating 14–23
 - editing 23–25
 - validating 26–28
 - unload.map 89, 90
- server
 - LDAP
 - alternate 236
 - defining 215–219
 - disabling 237
- ServiceCenter
 - applications defined 9
 - knowledge requirements 8
 - loading database 127–136
 - background 134–136
 - foreground 128–134
 - messages 131–134
 - overview 9
 - subformats 13, 17, 21, 23
- system
 - default records, See *SYSDEFAULTS
- system events 206

T

- TAPI 196–209
 - architecture 199–200
 - background processes 205
 - call flow 198
 - configurations 196
 - connectivity 201
 - debugging ServiceCenter 206–209
 - editing sc.ini file 202
 - enabling 201
 - editing shortcut 203
 - making calls 206
 - Microsoft TAPI 2.1 white paper 197
 - multiple calls 204
 - prerequisites 196
 - receiving calls 203–205
- telephony 195
 - connectivity 201
 - introduction 195
 - parameter 203
 - TAPI 196–209
 - architecture 199–200
 - background processes 205
 - call flow 198
 - configurations 196
 - connectivity 201
 - debugging ServiceCenter 206–209
 - enabling 201, 202, 203
 - making calls 206
 - multiple calls 204
 - prerequisites 196
 - receiving calls 203–205
- templates
 - LDAP
 - applying 230–231
 - creating 227–230
 - overview 212
 - viewing 227, 234
- training services 12
- translation
 - upper/lower case 186

U

- unload option
 - access export function 139

Unload Script Utility

- commonly scripted files 20–21
- creating an unload script 14–23
- editing records 23–25
- fields 17–18
- introduction 13
- protected formats 21–23
- purging records 32–33
- unloading 28–31
 - validating scripts 26–28
- unload utility 56
- unloading 93
 - data 93–127
 - Database Dictionary record (no data) 117–119
 - multiple databases 111–115
 - multiple records from one database 103–105
 - multiple records from single file (value list) 105–110
 - QBE only to Excel 115–116
 - Report Writer report 119–127
 - single records 95–103
 - form field definitions 101–103
 - upper case translation 186

V

- validating
 - fields
 - Format Control 167
 - unload scripts 26–28
- validations
 - export descriptor record 167–168
- validations process (Format Control)
 - using for export descriptor record 167–168
- value-list method 105
- variables
 - \$acntr 62
 - \$last.mod.date 62, 68, 89, 91
 - \$umap.modtime 62, 68, 92
 - \$umapdata 62
 - \$umapfile 62, 92
 - Prompt Data Map 89, 90

