ServiceCenter®

SCAuto for SPECTRUM

Version 4.1

(for SPECTRUM versions 4 and 5)

July, 1999

Peregrine Systems, Inc.
3611 Valley Centre Drive
San Diego, CA 92130

Peregrine SYSTEMS®

The Infrastructure Management Company™

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.

This edition applies to:
versions 3, 2.X and 1.4 of the licensed program of **ServiceCenter**®
release 4.1 of **SCAuto for SPECTRUM**, and
versions 4 and 5 of **SPECTRUM**

Contents

Preface

Chapter 1 Introduction

Chapter 2 Installation

Chapter 3 Operation

Chapter 4 Using SCAuto for SPECTRUM

Preface

Introduction

Welcome to Peregrine Systems *SCAuto for SPECTRUM*. This guide describes how to implement the SCAuto for SPECTRUM interface for integration with Peregrine Systems' ServiceCenter.

The SCAuto for SPECTRUM product is part of the suite of SCAutomate (SCAuto) interface products that integrate ServiceCenter with premier network and systems management tools. This interface is based on event messages sent over a TCP connection to the ServiceCenter server. Additional information about SCAutomate can be found in the *SCAutomate Applications for Windows NT and UNIX Guide.*

Knowledge Requirements

This guide assumes you have:

- Working knowledge of ServiceCenter applications, ServiceCenter Client/ Server, and the SPECTRUM graphical user interface. While some procedures for these applications are explained, others are referenced. Refer to the appropriate ServiceCenter documentation for a more detailed explanation.
- Familiarity with SPECTRUM and its components.
- Working knowledge of a GUI or text-based environment.
- (As an Administrator) a thorough knowledge of the operating system where ServiceCenter, SCAutomate, and the SCAuto for SPECTRUM product will be installed and implemented, as well as a basic understanding of ServiceCenter applications and Event Services.

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

Contacting Peregrine Systems

Contact one of the Peregrine Systems Customer Support offices listed here if you have questions about or problems with ServiceCenter systems.

North and South America

To get help immediately, call Peregrine Customer Support at:

(800) 960-9998 or **(619) 794-7402**

For ServiceCenter questions or information, send a fax or e-mail to:

Fax: **(619) 794-6028**
E-mail: **support@peregrine.com**
Hours: 5:00 A.M. to 5:30 P.M. PST, Monday through Friday

Send materials that Peregrine Systems Customer Support requests to:

Peregrine Systems, Inc.
ATTN: Customer Support
3611 Valley Centre Drive
San Diego, CA 92130

Europe

Great Britain

Peregrine Systems Ltd.
1st Floor
Ambassador House
Paradise Road
Richmond, Surrey, Great Britain, TW9 1SQ
Phone: **+44 (0) 181-332-9666**
Fax: **+ 44 (0) 181-334 5890**
E-mail: **uksupport@peregrine.com**
Hours: 08:00h to 18:00h GMT, Monday through Friday

France

Peregrine Systems
Tour Franklin-La Défense 8
92042 Paris La Défense Cedex, France
Phone (International Toll Free): **+33 (0)800 505 100**
Fax: **+33 (0)1 47 73 11 61**
E-mail: **frsupport@peregrine.com**
Hours: 08:00h to 18:00h, Monday through Friday

Germany

Peregrine Systems GmbH
Burohaus Atricom
Lyoner Strasse 15,

60528 Frankfurt, Germany
Phone: **+49-(0)69-66-80-260**
Fax: **+49-(0)69-66-80-2626**
Hours: 08:00h to 17:00h, Monday through Friday

Denmark

Peregrine Systems A/S
Naverland 2, 12 SAL
DK-2600 Glostrup
Denmark
Phone: **+45-4-346-7676**
Fax: **+45-4-346-7677**
Hours: 08:30h to 17:00h, Monday through Friday

Holland/
Netherlands/
Benelux

Peregrine Systems BV
Botnische Golf 9a
3446 CN Woerden
the Netherlands
Phone: **+31-348-437070**
Fax: **+31-348-437080**
Hours: 08:30h to 17:30h, Monday through Friday

Asia-Pacific

E-mail: **apsupport@peregrine.com**
Phone (U.S.): **619-794-7402**
Phone (Australia): **800-146-849**
Phone (Hong Kong): **800-90-8056**
Phone (Singapore): **800-1300-949-948**
Phone (Japan): **0044-221-22795**

Note: Only the Peregrine Systems European Customer Support staff is multilingual and can provide technical support to customers in their native language.

Chapter 1 Introduction

Overview

SCAuto for SPECTRUM is one of several ServiceCenter automation (SCAuto) products designed to enhance Peregrine Systems' HelpDesk and Inventory applications, and provide HelpDesk facilities to the SpectroGRAPH user.

SCAuto for SPECTRUM is composed of component daemons which handle inventory and problem reporting automation, as well as facilities to integrate ServiceCenter applications into the SpectroGRAPH operational environment. SCAuto for SPECTRUM utilizes supplied SpectroGRAPH platform API and command facilities in its operation. No modifications need to be made to the network manager.

Requirements

Integrated operational communication with ServiceCenter requires

1. the ServiceCenter client software, which allows SCAuto for SPECTRUM to invoke a ServiceCenter client.
2. The ServiceCenter client support on your network manager platform (which matches your server)
3. ServiceCenter application level A9602 or later (which contains the ServiceCenter Event Manager). Refer to *Chapter 2, Installation*, for more information on compatibility.

SCAuto for SPECTRUM components include:

- A topology/client daemon that performs inventory functions.
- A trap daemon that gathers problem information and dynamic inventory changes.
- Operational integration facilities for the supported platforms.
- Utility programs to refresh inventory and archive SCAuto for SPECTRUM files.

Figure 1-1 provides an overview of SCAuto components.

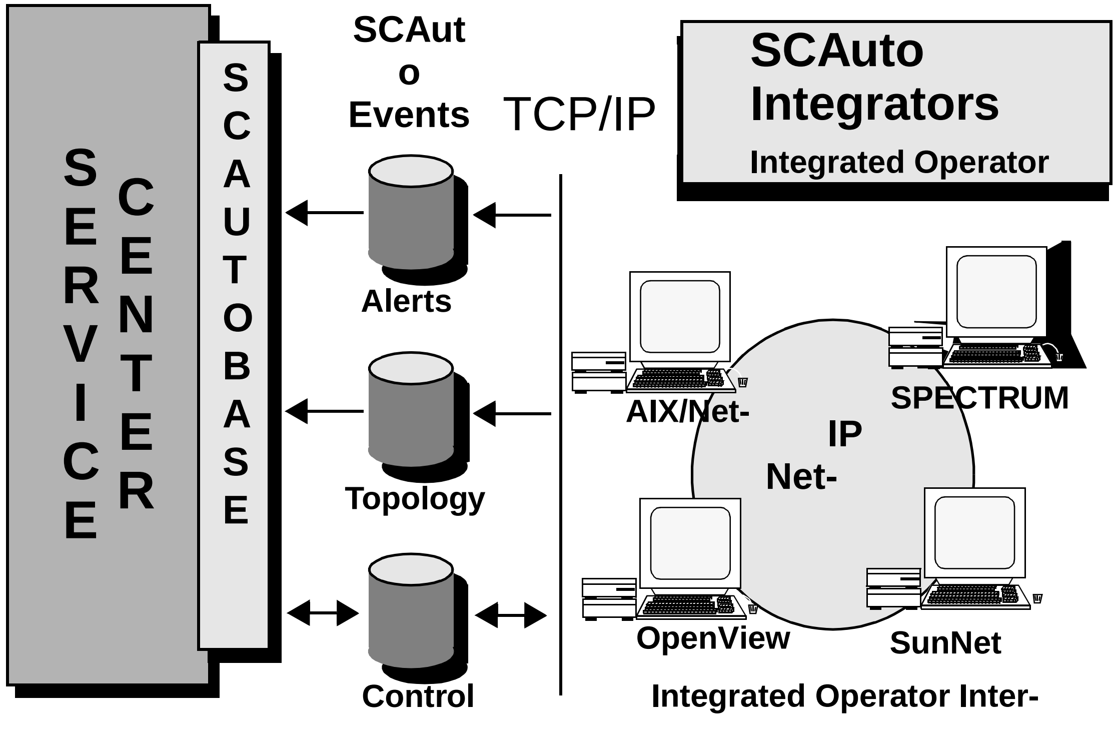


Figure 1-1 SCAuto Overview

Topology Management Overview

The primary function of the topology component, or daemon, is to create and maintain inventory records in ServiceCenter. The inventory records created correspond to the objects discovered by the network management platform. An inventory record is created and maintained for each selected model type occurrence in the network with the corresponding connections. Each record is maintained as a ServiceCenter device record, with special fields for the connection and control information. The connection data is important to graphical, path determination, outage analysis, and dependency propagation applications. The connection relationships possible include:

- Container (contained in relationship)
- Hierarchical (parent, child relationship)
- Point-to-point (peer relationship).

SCAuto for SPECTRUM topology management transforms the SPECTRUM management elements into corresponding device types in ServiceCenter. Different device types are required to represent the elements of the various networks. The major ServiceCenter inventory device types created by SCAuto for SPECTRUM are found in the files located in:

<SPECTRUM Root Directory>/scauto/maps/inventory directory of SPECTRUM

where <SPECTRUM Root Directory> is the *$SSHOME* for SPECTRUM version 4 and *$SPECROOT* for SPECTRUM version 5.

ServiceCenter devices are created for each model type mapfile supplied. The mapfiles contain tokens and attribute IDs and their associated Event Service map position. This permits each model to associate various attributes to particular ServiceCenter inventory fields.

The IP environment normally provides specific configuration information through the MIB (Management Information Base). If MIB information is maintained as a function of network administration, ServiceCenter inventory records can be maintained automatically. Those records will contain current contact and location information. The dynamic changes to IP inventory are kept current by topology event records created by the network manager discovery/management programs. The dynamic updates ensure the IP network is up to date and the current status is reflected in your ServiceCenter database.

All topology information is forwarded dynamically to ServiceCenter Event Service facilities, providing the automated network discovery information to any ServiceCenter application. The dynamic update, add and delete of network component information results in more accurate and timely information in ServiceCenter databases. This process also provides a reduction in the labor intensive processes to enter and maintain this data.

SCAuto for SPECTRUM topology utilizes standard inventory *add* and *update* events as described in the ***ServiceCenter Event Services Manual.*** The Event Manager is the ServiceCenter component which maps the input events and gives control to the RAD applications that process the events. In the case of inventory, an *icma* (inventory control management add) standard event is created and placed in the *eventin* file. The event scheduler reads the *eventin* file and maps the ServiceCenter event data into the device and attribute files. The scheduler would then perform a background inventory add operation. Refer to the ***ServiceCenter Event Services Manual*** for more information on the Event Manager and its standard facilities.

A firm understanding of the ServiceCenter Event Manager is helpful in reviewing the subsequent table which maps various element field names to ServiceCenter inventory records.

Problem Management Overview

The Problem Management component is comprised of a set of programs that interface with the network manager to dynamically open, close and update ServiceCenter problems. Problem actions are based on SPECTRUM alarms. Alarms are issued in SPECTRUM to notify the network manager of a specific event for a device or software. An event or sequence of events may be evaluated by SPECTRUM to indicate a problem. SCAuto for SPECTRUM allows user-specified filters to identify specific types of alarms for opening, closing, or updating problems. Two filtering types are provided: **local** and **global** filters.

Local filters are employed at the alarm origin and allow blocking of alarms based on *model type*, *cause code*, *model name*, and *condition*.

Specialized or **global** filters are specified on the ServiceCenter server to block problem reporting. Filtering can be based on: time of day, event type, event data, frequency of occurrence, thresholds, and time considerations. Because filters are specified on the server, all problem management components associated with the server are centrally administered. The Event Manager's standard facilities supply filtering formats to create, update and delete event management filters.

If a problem is not solved by alarm filtering or ServiceCenter filter specifications, user-written RAD functions or expressions can be used.

The SCAuto base TCP/IP environment is used to forward problem reporting information to the ServiceCenter server, including ***pmo*** (problem management open) and ***pmc*** (problem management close) events. The server's Event Manager passes the event data through the standard Problem Management application to open, update or close a problem.

Only one problem is opened per reporting component per SCAuto for SPECTRUM problem manager. All subsequent traps received from a component with an open problem ticket, are considered an update or a close of the opened problem.

For example, an alarm is received indicating a problem should be opened in ServiceCenter. SCAuto for SPECTRUM checks the Event Manager filters. If all is well, a problem open event (pmo) for the device is created and sent to the Event Manager, which opens a problem in ServiceCenter.

A subsequent alarm arrives, again the filters are checked and a problem update event is sent to the Event Manager. The Event Manager checks the event and notes it is from SCAuto. A check determines if a problem for this specific device was opened by this SCAuto for SPECTRUM problem manager. If a problem is open, it is updated. If no problem exists, a new problem is opened.

An alarm is received indicating the problem has been resolved. SCAuto for SPECTRUM creates a problem close event (pmc) and sends it to the Event Manager. The Event Manager closes the problem for the specified device.

Problems opened by SCAuto for SPECTRUM contain the information specified in the alarm map file specification for the specific model type indicated in the alarm or a DEFAULT map file is used. The alarm map files are specified in <SPECTRUM Root Directory>/scauto/maps/alarms and contain a format of token and attributes consistent with the topology component mapping.

Operations Integration Overview

The Operations Management components provide assorted facilities that integrate ServiceCenter into the window or console facilities of your network management platform. This integration provides a uniform operator interface, and reduces the operator entered information required in utilizing ServiceCenter facilities from the network management platform.

Integration is done by including ServiceCenter functions on your network management platforms menus, and when possible, eliminating redundant operator entry of previously selected object information. ServiceCenter GUI client provides access to all ServiceCenter facilities and selected streamlined services.

Note: This capability is available only if the ServiceCenter client is installed, and the ServiceCenter cut-throughs are enabled when SCAuto for SPECTRUM is installed.

The streamlined services allow you to bypass menus and application screens and invoke a requested application with passed variables to display a desired output window. For example, a Console operator selects a router element icon in the Console window and would like to request a list of ServiceCenter open problems for this element. The operator selects the ServiceCenter menu and chooses the open problems menu option. The selected router name is passed to a *streamlined* problem lookup. Any problems for that router are displayed in a separate ServiceCenter client GUI window.

Refer to *Chapter 4, Using SCAuto for SPECTRUM,* for information on the facilities provided on a specific platform.

SCAuto for SPECTRUM System Flow

SCAuto for SPECTRUM runs as daemon processes under the *userid* of the console or database to be monitored. It can run as a standard foreground or as a background process. The foreground execution capability insures that all error messages are displayed and allows you to run as a standard user when testing.

The supplied method of starting the SCAuto for SPECTRUM daemons is to select the ***_SCAuto_Start...*** function from the * menu. This selection runs *ssIPASpm*, the Process Monitor which starts the two daemons, *ssIPASd* and *ssAlarm*. Refer to Figure 1-2. The *ssAlarm* daemon should be run whenever the SPECTRUM server is running as they capture and log alarms reported dynamically. The *ssIPASd* daemon runs asynchronously to SPECTRUM and can be started or stopped with no loss of information when restarted.

The *ssAlarm* function uses the Alarm Notifier to capture the alarms and using supplied scripts filters and logs alarms to be processed by *ssIPASd*. As alarms are added, they are examined to determine if any SCAuto for SPECTRUM filters are present. When an alarm is selected, it is analyzed, reformatted and time stamped. A standard record is written to the *ssIPAS_alarms* file for processing.

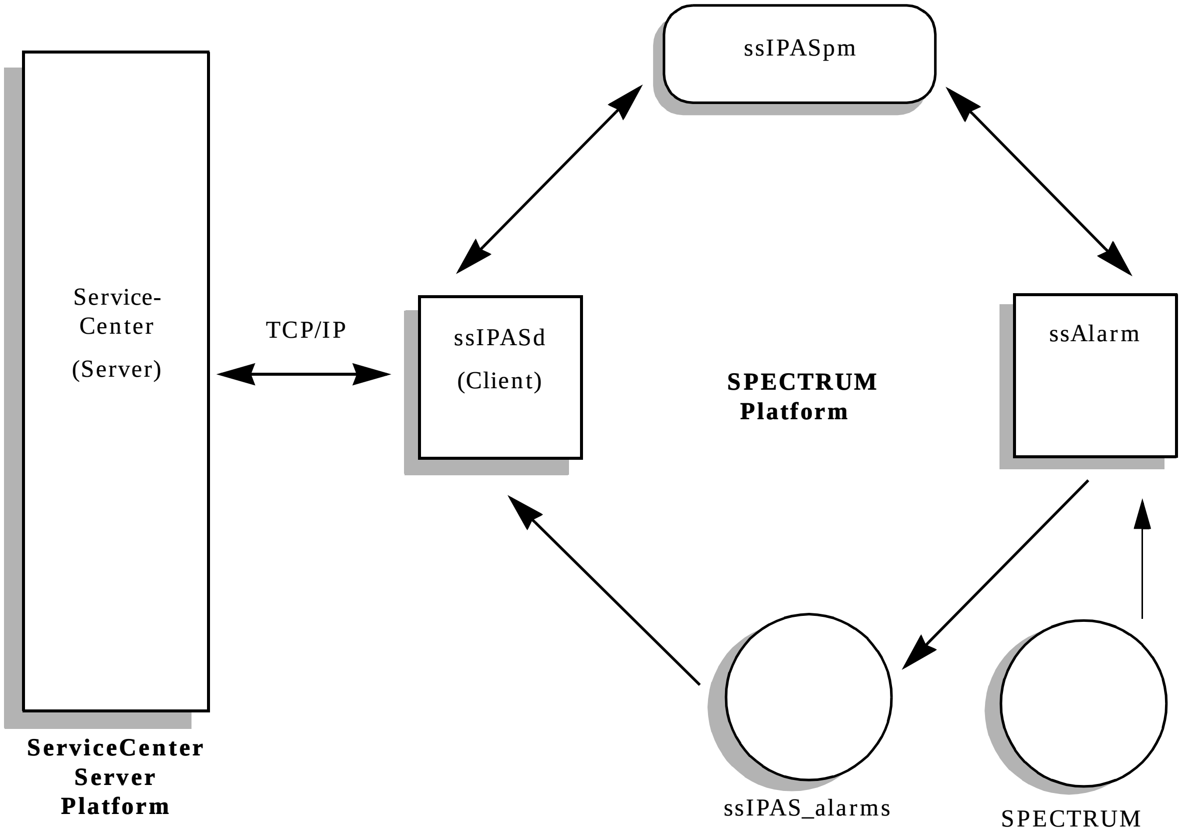


Figure 1-2 SCAuto for SPECTRUM System Flow

When the ServiceCenter Interface daemon starts (*ssIPASd),* it connects to ServiceCenter via the SCAuto base TCP/IP facility. Upon connection, the *ssIPASd* daemon ensures the user is licensed for use and opens or creates its required files. One such file, *ssIPASd.log,* is created or updated with each daemon execution and is especially useful in problem determination.

Next, the daemon determines if a refresh is requested (via the *ssIPAS_chkpt* file), or a resynchronization is needed. If a refresh is requested, all supported objects are read and objects are updated or added in ServiceCenter. If resynchronization is requested, a checkpoint record is read and *ssIPAS_alarms* is positioned. Log records are then processed from the checkpoint forward to open or close problems, or update inventory. Once the initial processing is complete, the daemon waits for a new problem or inventory event to occur.

Whenever *ssIPASd* has updated the *eventin* table in ServiceCenter, an Event Manager scheduler examines the new records and performs the appropriate actions to open, update or close a problem ticket, or update inventory files. This sequence continues until the daemons are killed or stopped using the ***SCAuto_Stop..*** selection on the ***** menu.

Chapter 2 Installation

Overview

This chapter provides information, instructions, and verification procedures for installing ServiceCenter SCAuto for SPECTRUM.

The sections are organized as follows:

- **Installation Notes -** General statements and conditions necessary before installation can be started.
- **Installation Procedures -** Steps required to install and configure the SCAuto for SPECTRUM ServiceCenter access product.
- **Installation Verification -** Steps involved in checking installation for proper configuration and setup of required parameters.
- **SCAuto Modifications -** Descriptions of SCAuto event data filtering and mapping conventions

Installation Notes

- SCAuto for SPECTRUM requires approximately 10 MB of hard disk space.
- If you plan to use ServiceCenter functions from SPECTRUM, validate that a standard ServiceCenter client is installed and runs with your ServiceCenter server from the platform you intend to run SCAuto for SPECTRUM.
- SCAuto for SPECTRUM must run on a platform where SpectroGRAPH and the CLI Interface are installed.
- Obtain an SCAuto for SPECTRUM authorization code from your Peregrine Systems Account Executive or Customer Support and restart your ServiceCenter server with this code in the ServiceCenter servers current *sc.ini* file.
- The installation asks for the name of a user that will own the SCAuto for SPECTRUM files. If you wish to create a new user for this, do so before starting the installation. It is suggested that this be the same user that owns the SPECTRUM files. The *root* user cannot be used for this purpose. You can also specify a group for the SCAuto files by entering **user.group** when prompted for a user name.

Installation Procedures

This section provides the SCAuto for SPECTRUM installation instructions.

1. Use the **tar** command to extract the tar file on your SCAuto for SPECTRUM CD to a temporary install directory.
2. If the ServiceCenter client is installed, test it and ensure that it can connect to the desired server from the network manager platforms.
3. Obtain an SCAuto for SPECTRUM authorization code from your Peregrine Account Executive or Peregrine Customer Support at (800) 638-5231.
4. Restart your target ServiceCenter server with the proper authorization code.
5. The tar file on the SCAuto for SPECTRUM product CD contains all files required to install and run SCAuto for SPECTRUM. You may need root authorization to complete the installation. An install program (**INSTALL**) is provided which creates your execution directories. Once you have the proper file authorizations, execute the following command as the SPECTRUM user:

   INSTALL

Note: Ensure you have changed to the *tar directory* directory (from step 1) before running the INSTALL program. If you are not in this directory, the installation will fail.

6. The install application INSTALL prompts you for the answers to some key requirements. The responses must be accurate for successful operation.
7. Review the output produced from INSTALL and correct any error situations, permissions, etc. You can rerun INSTALL without a problem. Even though INSTALL always attempts to do a complete install, no problems will occur on steps that were previously accomplished.
8. Update the *sc.ini* file in *<SPECTRUM Root Directory>/scauto* with your SCAuto Base server name and ServiceCenter client specifications. See *ServiceCenter Quick Installation Guide, Client/Server For UNIX*.

At this time, you should also update your *sc.ini* with your SCAuto specifications. The SCAuto parameters are identified with a **#@** prefix:

#@scinv:yes|no

Do you want the automated inventory functions? The default is **yes**.

d#@scauto:

The service name in */etc/services* or port number of your SCAuto base. If the base is on the SPECTRUM platform only the service name is required. If on a separate computer specify *hostname.servicename*.

#@scprob:yes|no

Should SCAuto automatically open and close problems? The default is **yes.**

#@scdatefmt: 1|2|3

The date format corresponding to ServiceCenter date formats to be used by SCAuto for SPECTRUM.

d**#@scevsufx:event suffix**

Alphanumeric suffix to be appended to events created by this machine running SCAuto. The default is **no suffix.**

If you specify a suffix your event tables in ServiceCenter, Event Manager must be modified to include a set of new event types, *pmosuffix*, *pmcsuffix*, *icmasuffix*, and *icmdsuffix*. This may require you to do some of your own RAD coding. Therefore, do not specify a suffix unless you fully understand the implications.

#@sclocation:default location

The specified location is utilized by SCAuto when no location is specified in the MIB.

#@sccategory:problem category

The specified category is used by SCAuto on problem open and close.

9. Define symbolic links to ServiceCenter binaries (if not done by installation script):
   - **ln -s /<ServiceCenter directory>/RUN/scclient <SPECTRUM Root Directory> scauto/scclient**
   - **ln -s /<ServiceCenter directory>/RUN/scclient <SPECTRUM Root Directory>/scauto/scguimtf**
   - **ln -s /<ServiceCenter directory>/RUN/scclient <SPECTRUM Root Directory>/scauto/scgui.uid**
10. Ensure that all network manager operators have *<SPECTRUM Root Directory>/scauto* directory specified in their PATH environments.
11. Update the **<ServiceCenter directory>/scauto/config** with the **VNM** host name of your SPECTRO server and the **landscape handle** to be utilized by SCAuto for SPECTRUM(ssAlarm).
12. Review the filters in **<ServiceCenter directory>/scauto/filters** and update if required.
13. Review the maps in **<ServiceCenter directory>/scauto/maps** and update if required.

Refer to the next section to verify the installation and the modifications section for filter and map formats.

Installation Verification

This section includes installation checks and possible solutions for installation problems.

1. Start your ServiceCenter server.
2. To start SCAuto, enter:

   <SPECTRUM Root Directory>/scauto/ssIPASpm -start

   This command normally is executed when you select the SCAuto start option from your SPECTRUM * menu. This command starts the SCAuto daemons ssAlarm, which is the SCAuto logger, and ssIPASd which is the SCAuto client daemon. Verify that the two daemons are running as follows:

   a. The ssAlarm daemon creates SCAuto log records in *<SPECTRUM Root Directory>/scauto/ssIPAS_alarms*. If log records are located in this file after events are detected, the logger is operating correctly.

   b. Start the SCAuto Base on the ServiceCenter Server platform. The ssIPASd daemon connects to the SCAuto Base and determines if the ServiceCenter server is licensed for SCAuto for SPECTRUM. If so, the events are created and inserted in the *eventin* table on the ServiceCenter server. If the *eventin* records are present at the server, check the Event Manager installation using the ServiceCenter Event Manager installation procedures.

   c. Verify that events are appearing in the *eventin* table by logging in as a ServiceCenter client and selecting Event Services.

   d. Select **Review Input** then click on the **Search** icon to display all input events.

   e. Search for the events with the **user-id** field of *SS-<hostname>* where hostname identifies the machine SCAuto is running on. If these records are found, SCAuto and ServiceCenter are communicating.

   If you are having problems with an **ssIPASd** connection, verify that the */etc/services* file has your SCAuto Base specification and that the SCAuto Base is up and running on the ServiceCenter server platform.

3. If the ServiceCenter cut-throughs were enabled when SCAuto for SPECTRUM was installed, verify the SCAuto ServiceCenter operational facilities by starting SpectroGRAPH.

   Your network manager platform windows are displayed.

4. Verify the * menu bar contains the **ServiceCenter** options. Select a component on one of your network views and check the ***Utilities*** menu selection for Service Center selections. Refer to *Chapter 4, Using SCAuto*, to verify the various options in this facility.

You should become familiar with the utilities and filtering functions after the basic installation is complete. These post-installation capabilities are discussed later in this manual.

SCAuto Modifications

SCAuto dynamically creates the following events in ServiceCenter:

- pmo (problem open)
- pmc (problem close)

Filtering

The event type and subsequent processing is determined by the log record information in ssIPAS_alarms. The creation of a log record determines if the alarm is ignored (filtered), or what type of operation should be performed.

The files that determines the alarm filtering are located in the <SPECTRUM Root Directory>/scauto/filters subdirectory. The filters are inclusive in the following order ssIPAS_mtype, ssIPAS_mname, ssIPAS_cause, and ssIPAS_color.

These filters correspond to the fields presented by the AlarmNotifier to the problem open alarm script. Each value in a filter is a single entry followed by a newline character. The value specified should be identical to the value presented by the AlarmNotifier. Once a filter is created only those values specified will create log records and subsequently open problems. Alarm filtering is usually set to only create problem tickets on key elements in the network such as routers, gateways, and various servers.

Mapping

There are two types of mapping provided in SCAuto for SPECTRUM

- Inventory maps located in <SPECTRUM Root Directory>/scauto/maps/inventory
- Alarm maps located in <SPECTRUM Root Directory>/scauto/maps/alarms

The format of a map is a combination of tokens and attribute IDs positionally specified to correspond to the Event Services input map for the specific event type. Sample maps are provided which display the tokens and syntax for alarms and inventory.

The samples assume the use of standard Event Services events **pmo** (problem open) and **icma** (Inventory Add). You should review these event maps in Event Services and correlate them to the SCAuto samples and understand the relationships prior to creating your own maps.

You can review the Event Services maps by:

1. Starting a Service Center client (<SPECTRUM Root Directory>/scauto/scclient -G)
2. Selecting Event Services from the Main Menu.
3. Selecting Maps on Event Services Menu
4. Enter the map name (inventory or problem)
5. Compare the field positions and names to supplied samples
6. Once the relationship and tokens are understood you're ready to create your own maps

Inventory will only be build if a map is provided for the model type.

Alarms will be created using the alarm map for a specific model type, or the DEFAULT alarm map if a model type is not specified. The following sample maps are included in <SPECTRUM Root Directory>/scauto/maps and define syntax and tokens available in SCAuto mapping services.

Mapping Examples

```
############################################################
#This file contains the inventory mapping for Pingable as referenced by
#the file name. It contains a list of attributes and tokens to be mapped
#against the corresponding evtype icma(inventory add) in Event Services.
#The $DELIM token should only be used to indicate an omitted field,
#delimiters will automatically be placed after all other fields.
# Inventory tokens are:
# $CONT  - place container field(if known) else a delimiter
# $DATE - generate a date time mm/dd/yy hh/mm/ss for the position
# $DEFCAT - use the default category
# $DELIM - generate a delimiter(omitted entry)
# $ENDPT1 - place enpoint one if known(peer to peer)
# $ENDPT2 - place enpoint two if known(peer to peer)
# $LH - place landscape handle
# $MH - place model handle
# $MTYPE - place model type
# $PARENT - place parent name if known
# $SCAUTO - place this scauto id
# All attributes must be in the form 0xnnnnnn
#############################################################
0x1006e       1       logical.name
$DELIM        2       vendor
$PARENT       3       pparent
$DELIM        4       model
$MH           5       net.name
$DELIM        6       serial
$DELIM        7       location
$DELIM        8       group
$MTYPE        9       type
$DELIM        10      estatus
$DELIM        11      pstatus
$DELIM        12      ID
$DATE         13      last.update
$SCAUTO       14      updated.by
$DELIM        15      description
$DELIM        16      view.name
$LH           17      container
$DELIM        18      ept1
$DELIM        19      ept2
$DELIM        20      pcount
$DELIM        21      nondevice
$DEFCAT       22      problem.category
$DELIM        23      table.name
$DELIM        24      physical.addr
$DELIM        25      objid
$DELIM        26      domain
$DELIM        27      protocol
0x1027f       28      protocol.addr
$DELIM        29      contact
$DELIM        30      subtype
$DELIM        31      icount
```

Figure 2-1. Sample Inventory Map Pingable Model Type

```
#############################################################
#This file contains the inventory mapping for GnSNMPDev as referenced #by the file name.
It contains a list of attributes and tokens to be mapped #against the corresponding
evtype icma(inventory add) in Event Services.
#The $DELIM token should only be used to indicate an omitted field, #delimiters will
automatically be placed after all other fields.
# Inventory tokens are:
# $CONT  - place container field(if known) else a delimiter
# $DATE - generate a date time mm/dd/yy hh/mm/ss for the position
# $DEFCAT - use the default category
# $DELIM - generate a delimiter(omitted entry)
# $ENDPT1 - place enpoint one if known(peer to peer)
# $ENDPT2 - place enpoint two if known(peer to peer)
# $LH - place landscape handle
# $MH - place model handle
# $MTYPE - place model type
# $PARENT - place parent name if known
# $SCAUTO - place this scauto id
# All attributes must be in the form 0xnnnnnn
#############################################################
0x1006e        1       logical.name
0x230683       2       vendor
$PARENT        3       pparent
0x10031        4       model
$MH            5       net.name
0x10030        6       serial
0x23000d       7       location
$DELIM         8       group
$MTYPE         9       type
$DELIM         10      estatus
$DELIM         11      pstatus
$DELIM         12      ID
$DATE          13      last.update
$SCAUTO        14      updated.by
0x10052        15      description
$DELIM         16      view.name
$LH            17      container
$DELIM         18      ept1
$DELIM         19      ept2
$DELIM         20      pcount
$DELIM         21      nondevice
$DEFCAT        22      problem.category
$DELIM         23      table.name
0x110df        24      physical.addr
$DELIM         25      objid
$DELIM         26      domain
0x1196c        27      protocol
0x1027f        28      protocol.addr
0x23000c       29      contact
$DELIM         30      subtype
$DELIM         31      icount
```

Figure 2-2. Sample Inventory Map GnSNMPDev Model Type

```
#############################################################
#This file contains the inventory mapping for a DEFAULT problem if no #custom alarm map
exists. Prior to the use of this map a check is made to #determine if a custom map is
present by searching this directory for a #alarm modeltype file name. It contains a list
of attributes and tokens to #be mapped against the corresponding evtype pmo(problem open
event) in #Event Services.
#The $DELIM token should only be used to indicate an omitted field, #delimiters will
automatically be placed after all other fields.
# Problem tokens are:
# $MNAME  - place model name from alarm log record
# $DATE - generate a date time mm/dd/yy hh/mm/ss for the position
# $DEFCAT - use the default category
# $DEFLOC - use the default location
# $DELIM - generate a delimiter(omitted entry)
# $MTYPE - place model type from alarm log record
# $ALERTID - place alert ID from alarm log record
# $CAUSE - place cause code from alarm record
# $COLOR - place alert color from alarm record
# $CDESC - on problem open information from spectrum cause file.
#                - on problem update, or close the alarm status field
# $ASSIGN - place repairman from spectrum alarm
# $SCAUTO - place this scauto id
# All attributes must be in the form 0xnnnnnn
#############################################################
$MNAME       1       logical.name
$MNAME       2       netname
$ALERTID     3       reference number
$CAUSE       4       cause code
$CDESC       5       problem description
$DELIM       6       reserved
$DELIM       7       reserved
0x1027f      8       network address
$SCAUTO      9       type of alert
$DEFCAT      10      category
$DELIM       11      domain
$DELIM       12      objid
$DELIM       13      version
$MTYPE       14      model
$DELIM       15      serial
$DELIM       16      vendor
$DEFLOC      17      location
$DELIM       18      contact name
$DELIM       19      contact phone
$DELIM       20      resolution
$ASSIGN      21      assignee name
```

Figure 2-3 DEFAULT Alarm Map

Chapter 3 Operation

Overview

This chapter covers operation procedures for starting and stopping SCAuto for SPECTRUM.

The material is organized as follows:

- **Starting SCAuto for SPECTRUM -** Instructions on using the ssIPASpm application and initiating the SCAuto for SPECTRUM daemons.
- **Stopping SCAuto for SPECTRUM -** Instructions on stopping the SCAuto for SPECTRUM utility and capturing the ssIPASd ID.
- **SCAuto for SPECTRUM Utilities -** Descriptions of the SCAuto for SPECTRUM utilities and their operation:
  - Archive Utility
  - Refresh Utility

Starting SCAuto for SPECTRUM

SCAuto for SPECTRUM is normally controlled by ssIPASpm, a supplied application.You should require special permissions to execute any of these commands.

1. Start the SCAuto base on your ServiceCenter sever platform
2. To start the SCAuto for SPECTRUM daemons, select ***Start SCauto*** from the SPECTRUM ✱ menu or issue the following command:
3. **<SPECTRUM Root Directory>/scauto/ssIPASpm -start**
4. You can also start the daemons outside of the ssIPASpm process management by issuing the following commands:

   <SPECTRUM Root Directory>/scauto/ssIPASd&

   <SPECTRUM Root Directory>/scauto/ssAlarm config&

Stopping SCAuto for SPECTRUM

This section provides the steps for stopping SCAuto for SPECTRUM.

1. You can stop all agents by issuing the following command.

   a. To stop all agents, issue this command or select ***Stop SCauto*** from the ***** menu:

      <SPECTRUM Root Directory>/scauto/ssIPASpm -stop

2. To stop the daemons separately, first issue the following command to get the necessary data to complete a **kill** command:

   /bin/ps -e |grep ss

   Extract the ssIPASd or the ssAlarm process ids, or both. Substitute those values in the following command:

   kill -9 *pid1* [*pid2*] [pid3]

   Where *pid1* is the ss*IPASd* process id and *pid2* is the ssAlarm process id.

SCAuto for SPECTRUM Utilities

This section describes the SCAuto for SPECTRUM utilities and their operation.

SCAuto for SPECTRUM Archive Utility

An **Archive** utility is provided to archive the *ssIPAS_alarms* file, which can be deleted or put to external media. The ss*IPAS_alarms* file could grow infinitely based on the disk available. Periodically, you should remove processed records and save the disk. The archive utility produces two files: an archive file containing all processed records, and a new ss*IPAS_alarms* with all unprocessed records.

In order to run the archive, the daemons (ssIPASd and ssAlarm) must be stopped with the following command.

```
<SPECTRUM Root Directory>/scauto/ssIPASarc
```

The utility prompts you, then executes after you have responded. After the process is complete, a new date/time stamped archive file is placed in the *<SPECTRUM Root Directory>/scauto* directory which can be removed, copied, etc.

SCAuto for SPECTRUM Inventory Refresh Utility

The **SCAuto for SPECTRUM Inventory Refresh Utility** is provided to allow the entire inventory or segments of the inventory to be refreshed in ServiceCenter. Initially, the inventory is created by the ssIPASd daemon when the **#@scinv:yes** parameter is specified and a checkpoint (ssIPAS_chkpt) cannot be located on the log tape (ss*IPAS_alarms*). You can update or recreate this inventory without impacting the ssIPASd daemon and its running environment.

When ssIPASd initializes, *ssIPAS_chkpt* is opened. If it is not present or the record it contains cannot be found in *ssIPAS_alarms*, the ssIPASd performs a restart. A restart assumes that SCAuto for SPECTRUM is to start over by initializing inventory and processing the log from this new point. This should only happen at the initial start of SCAuto for SPECTRUM after installation.

The SCAuto for SPECTRUM Inventory Refresh utility can run concurrently with the SCAuto for SPECTRUM daemon (ssIPASd). The utility attaches to the specified server and creates inventory events (icma) to update or create the specified inventory.

The Inventory Refresh Utility can be started by:

```
<SPECTRUM Root Directory>/scauto/ssIPASr -s SCAUTO_server -t
invtype / all
```

Where:

-s SCAUTO_server is the SCAUTO base server you wish to update. This should be specified as *host.servname* if the server is on a remote host.

-t invtype / all is the specific inventory type to refresh or all types. Refer to <SPECTRUM Root Directory>/scauto/maps/inventory for specific inventory types created by SCAuto for SPECTRUM.

For example, the following command refreshes the GnSNMPDev type inventory records for the network in using SCAUTO base server *scauto01*.

```
<SPECTRUM Root Directory>/scauto/ssIPASr -s scauto01 -t
GnSNMPDev
```

Note: The SCAuto Base must be running on the ServiceCenter server platform for the refresh utility to function.

Chapter 4 Using SCAuto for SPECTRUM

Overview

SCAuto for SPECTRUM provides an enhanced operator interface to ServiceCenter that integrates ServiceCenter facilities into the Spectrum system. From a Spectrum window, you can access a number of ServiceCenter screens to gather information related to the current window or selected element.

This capability is only available if the ServiceCenter cut-throughs were enabled during the SCAuto for SPECTRUM installation. If the cut-throughs were not enabled, a client must be started from the UNIX command line instead.

Note: SCAuto for SPECTRUM operates under the A9503 or later version of ServiceCenter 1.3.x. Refer to the appropriate ServiceCenter documentation for more information on using ServiceCenter.

SCAuto for SPECTRUM provides access to ServiceCenter through two methods. The primary ServiceCenter tools appear within the ***** (asterisk) pulldown menu in a Spectrum window (Figure 4-1). Another set of tools appear within the second (middle) button popup menu for a selected element (Figure 4-1).

Note: In order to access a ServiceCenter window, Spectrum must not be running from the *root* user account.

Important: Screen captures and functions may change from release to release. Reference the current ServiceCenter documentation for your specific platform for the latest operational details.

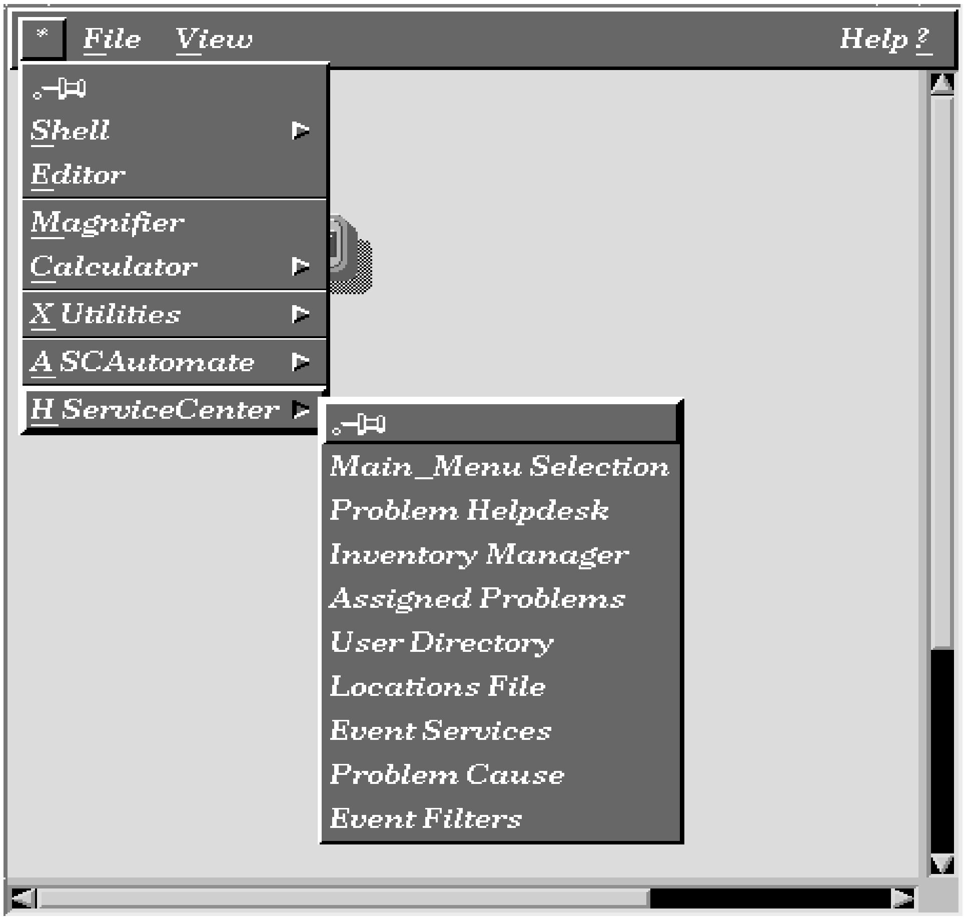


Figure 4-1. SCAuto for SPECTRUM ServiceCenter Submenu

Access the primary SCAuto for SPECTRUM tools from the **ServiceCenter** submenu in the Spectrum ***** (asterisk) menu as shown.

Primary tools give you general access to ServiceCenter. The **SCAutomate** submenu allows you to control the SCAuto for SPECTRUM daemons.

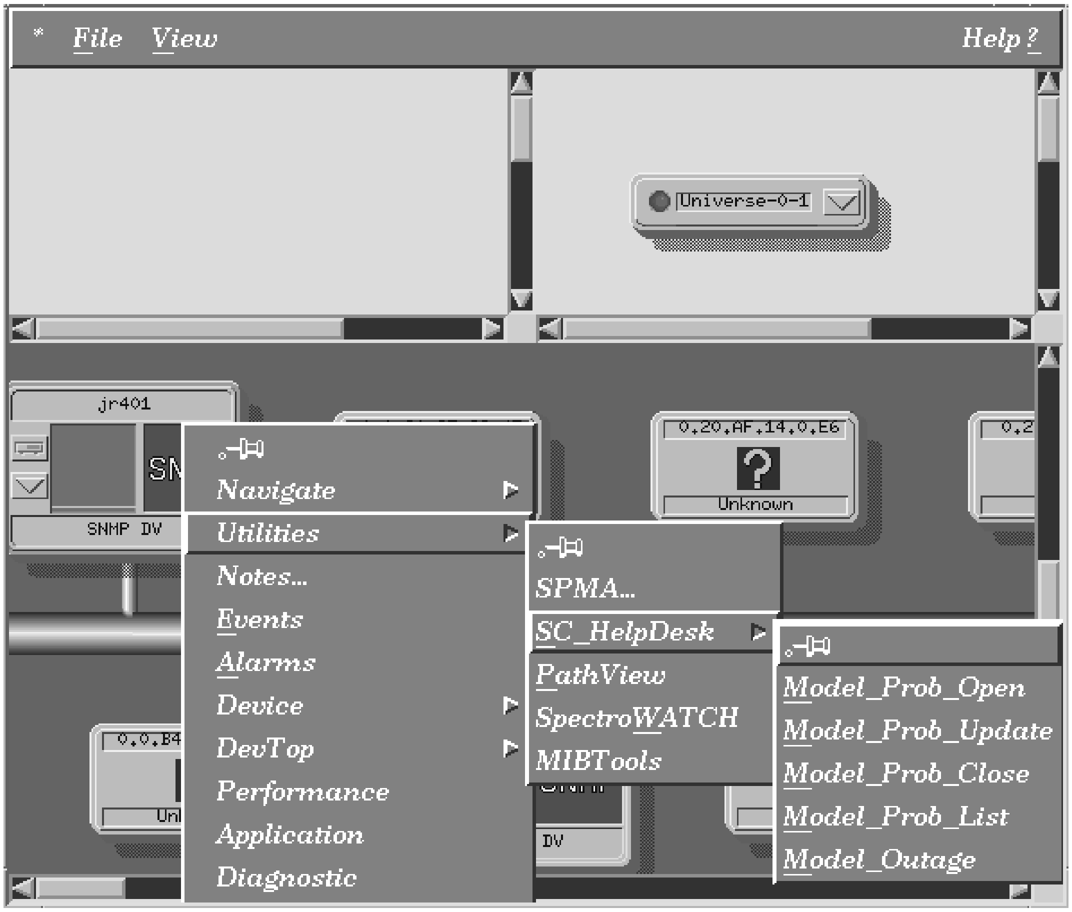


Figure 4-2. SCAuto for SPECTRUM Popup Menu

Access element-oriented SCAuto for SPECTRUM tools by selecting an element and clicking the second (middle) mouse button.

To access ServiceCenter, select **Utilities -> SC_HelpDesk**. A list of ServiceCenter options is displayed in a third pop-up menu.

These options give you direct access to ServiceCenter functions that operate on the selected element in Spectrum

Requirements

Before using SCAuto for SPECTRUM, you should have a good working knowledge of:

- ServiceCenter applications.
- ServiceCenter Client Server.
- Spectrum operations.

While some procedures for these applications are explained, others are referenced. Refer to the appropriate documentation for detailed information.

Mouse Conventions

The mouse buttons are used as follows when operating ServiceCenter under SCAuto for SPECTRUM, assuming a right hand, three-button mouse.

- Left (1) button - activates and selects the options from the **ServiceCenter** menu in the Spectrum windows. In ServiceCenter windows, button 1 is used to place cursor in a field.
- Middle (2) button - activates the popup menu in a Spectrum window. To select an option, keep the button depressed until the option is highlighted, and release the button
- Right (3) button - closes the selected window.

SCAutomate Submenu

This menu provides two options—Start SCAuto and Stop SCAuto.

Start and Stop SCAuto

The SCAuto for SPECTRUM implementation requires two daemons running to connect and provide inventory and problem information to ServiceCenter. These daemons may be started using the selected menu items. Please check the logfile or your console X_window for any error messages. The daemons are *ssIPASd* and *ssAlarm*, which were described earlier. If any problem arises using the menu functions, the daemons may be started as a background or foreground Unix process.

ServiceCenter Submenu Options

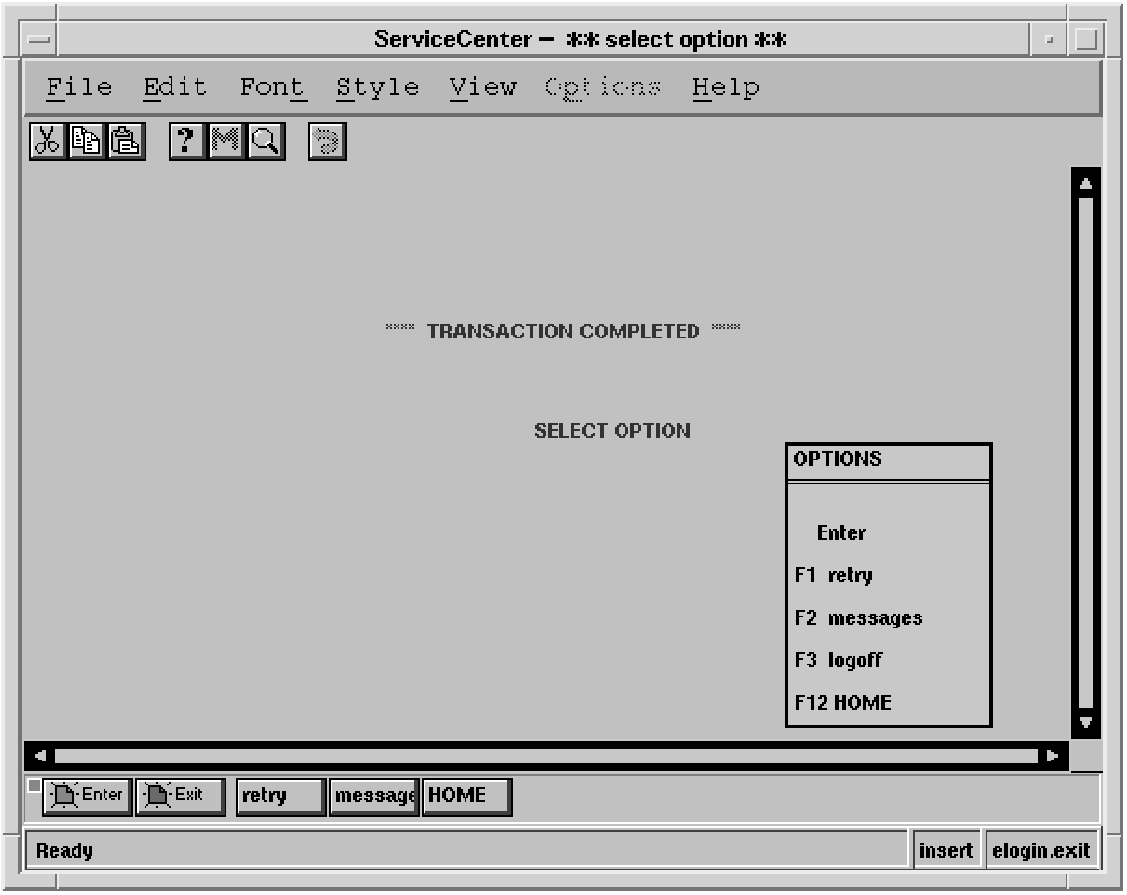
The ServiceCenter menu options take you directly to the ServiceCenter applications from Spectrum, without logging on. These express services save the time of logging in and navigating through ServiceCenter to get to these applications. The following sections provide a brief description of the menu options and screens.

Note: While some of the ServiceCenter options are mentioned in this manual, you should refer to the ServiceCenter documentation for complete instructions on using these options.

To use a ServiceCenter application under SCAuto for SPECTRUM:

1. Select an option from either the **ServiceCenter** menu in a Spectrum window or from the SC_HelpDesk popup menu for a selected element.
2. The appropriate ServiceCenter window is displayed. Many of these screens are discussed in the remainder of this chapter.
3. To leave the application, select **F3 end** from the popup menu or press the **F3** key. This takes you to the previous screen or a logout screen.

4. When the *login.exit* screen is displayed (Figure 4-3), select **F3 logoff** from the popup menu or press the **F3** key to exit the ServiceCenter session.



ServiceCenter – ** select option **

**** TRANSACTION COMPLETED ****

SELECT OPTION

Ready

Figure 4-3. Login.exit Screen

Main_Menu Selection

The main menu (Figure 4-4) provides access to the various applications within ServiceCenter.

By pressing the third mouse button, you can access a pop-up menu or function keys with two options: **F1 Command** and **F3 logoff**.

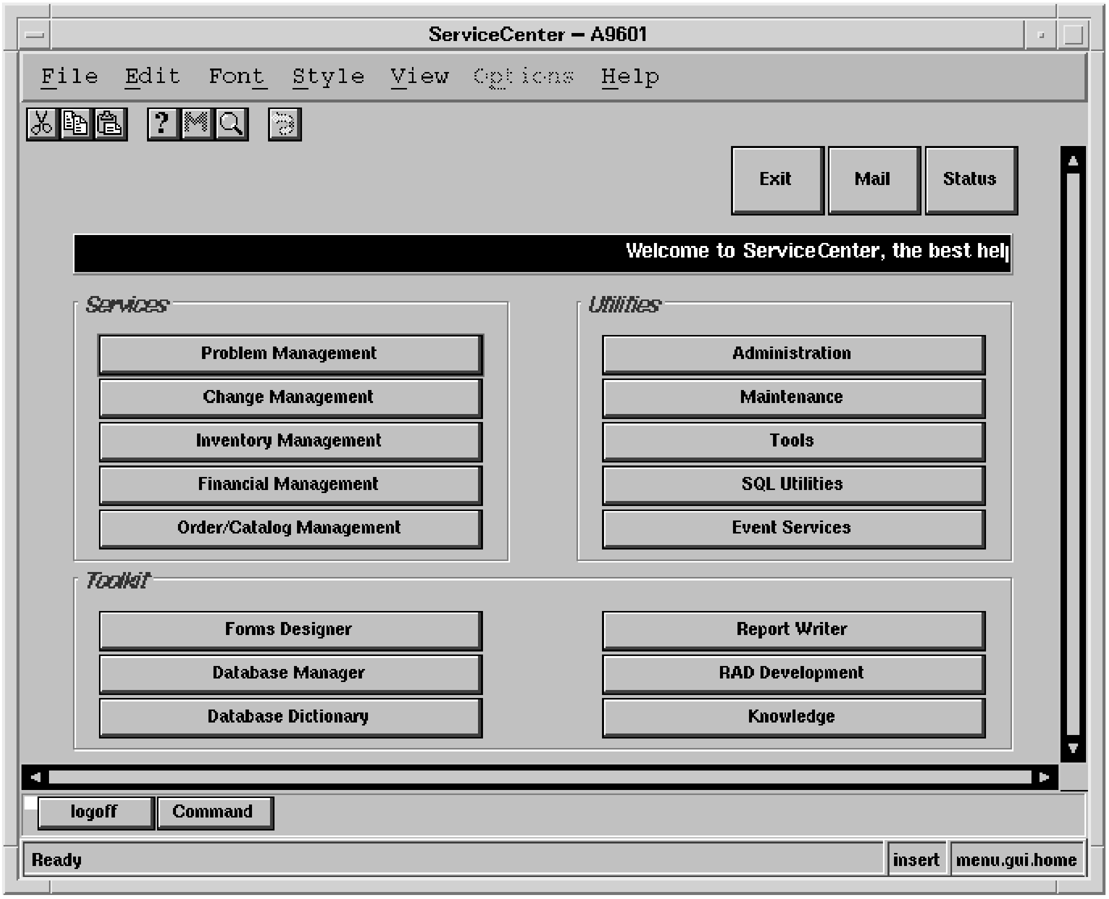


Figure 4-4. ServiceCenter Main Menu

Problem Helpdesk

The Problem Helpdesk option accesses the Problem Management screen (Figure 4-5).

The option for the buttons at the bottom of the screen are also available through the function keys or the popup menu:

- **F3 Back**
- **F5 New**
- **F6 Search**
- **F7 Views**
- **F8 Find**
- **F9 Fill**.

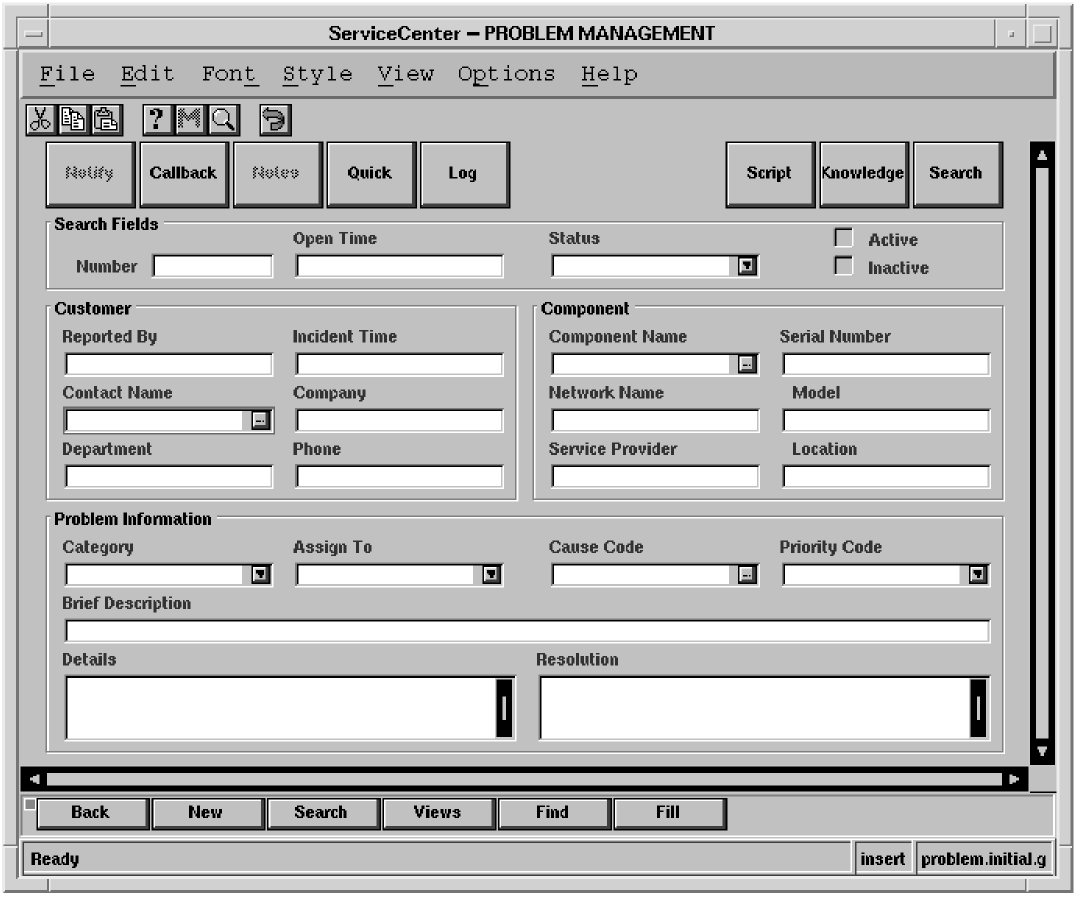


Figure 4-5. Problem Helpdesk Screen

Inventory List

This menu option accesses ServiceCenter's Inventory/Configuration Management application (Figure 4-6).

The **Back** option button at the bottom of the screen is also available through the popup menu or by using **F3**.

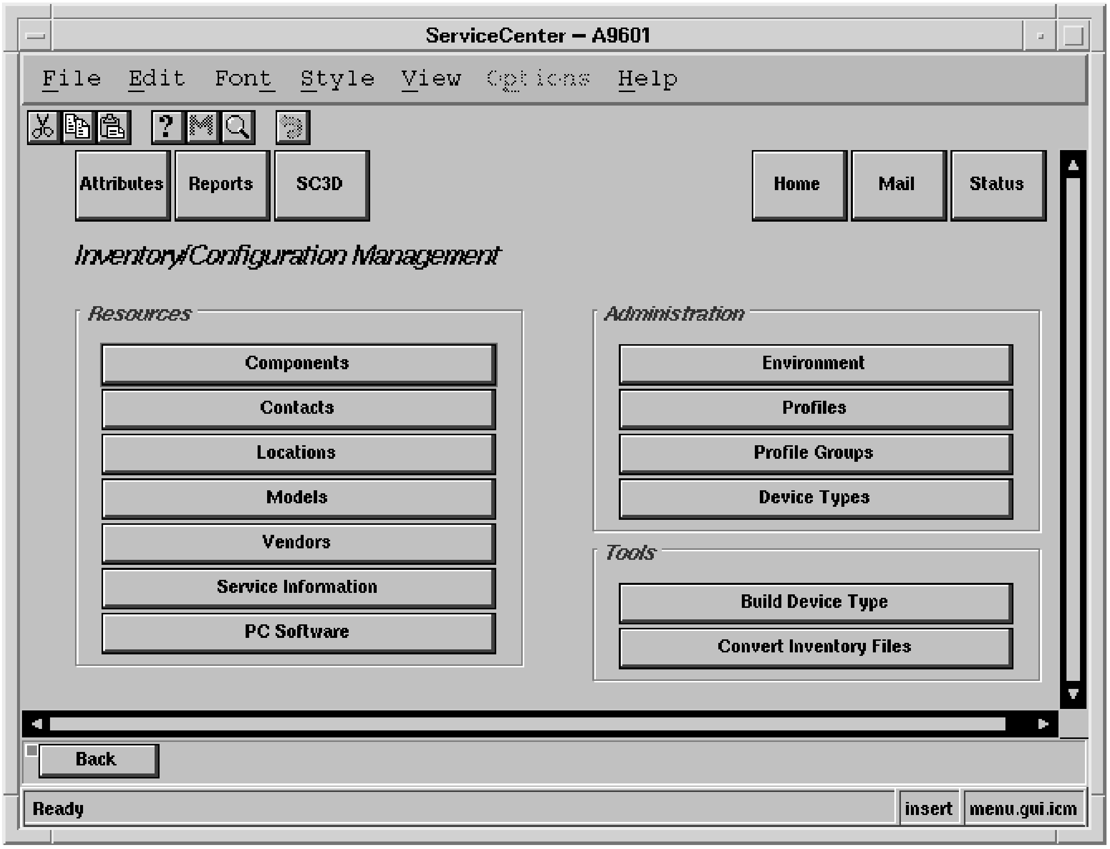
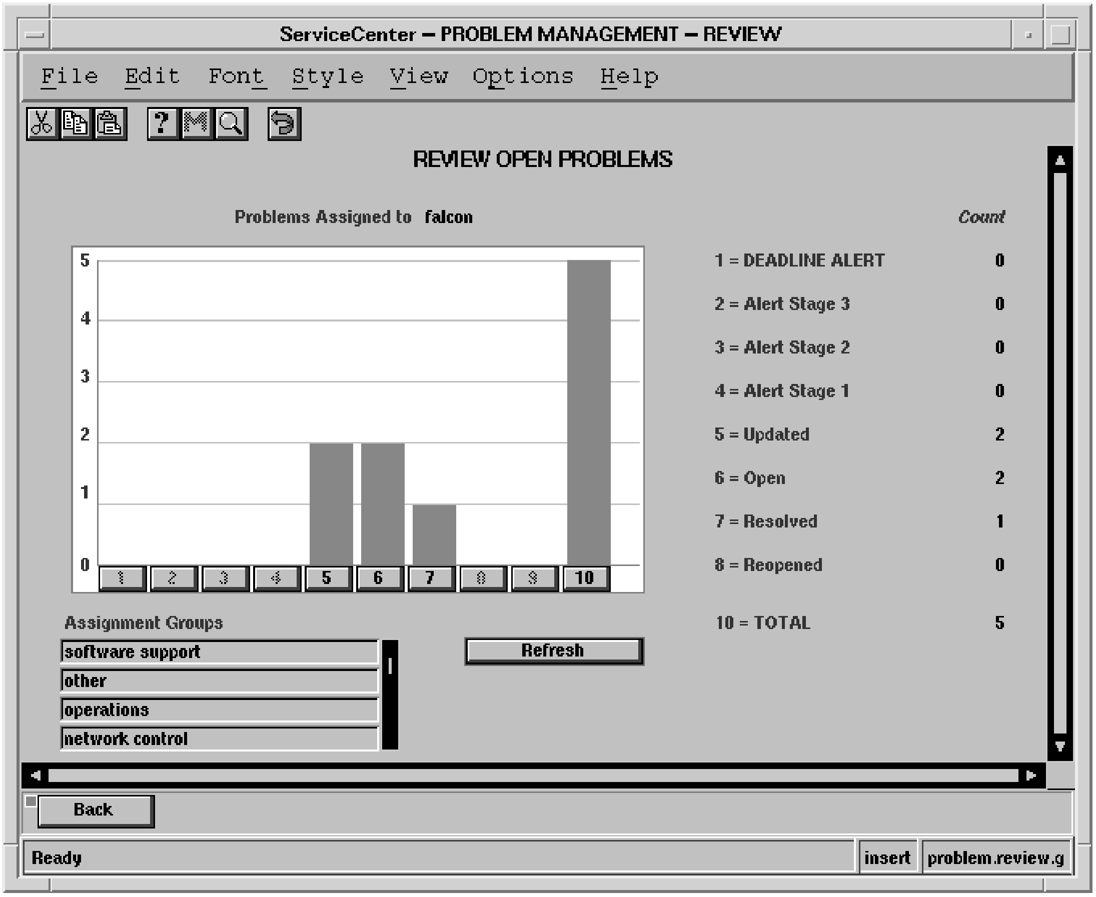


Figure 4-6. Inventory/Configuration Management Screen

Assigned Problems

This option accesses the Problem Management Review screen (Figure 4-7). This screen displays a summary of the problems assigned to the current user. If no problems are assigned to this user, you can press Enter and a prompt window asks you for the name of another user.

The **Back** and **OK** option buttons at the bottom of the screen are also available through the popup menu or by using **F3** and **F2**, respectively.



ServiceCenter – PROBLEM MANAGEMENT – REVIEW

REVIEW OPEN PROBLEMS

Problems Assigned to falcon

	Count
1 = DEADLINE ALERT	0
2 = Alert Stage 3	0
3 = Alert Stage 2	0
4 = Alert Stage 1	0
5 = Updated	2
6 = Open	2
7 = Resolved	1
8 = Reopened	0
10 = TOTAL	5

Assignment Groups

software support

other

operations

network control

Ready | insert | problem.review.g

Figure 4-7. Assigned Problems Screen

User Directory

This option accesses the User Contact Information in the ServiceCenter database. The **User Contact** screen (Figure 4-8) allows you to add, update, or query for a particular user. This screen is blank when first accessed. To query for user information, enter known data in the appropriate field and select **Query** from the **Options** popup menu.

To get a user list, press **Enter** after the blank screen appears. A user list is displayed. Double-click on the desired user to get the User Directory information for that user.

The option for the buttons at the bottom of the screen are also available through the function keys or the popup menu:

- **Enter**
- **Search**
- **F2 Save**
- **F3 Back**
- **F8 Find**
- **F9 Fill**.

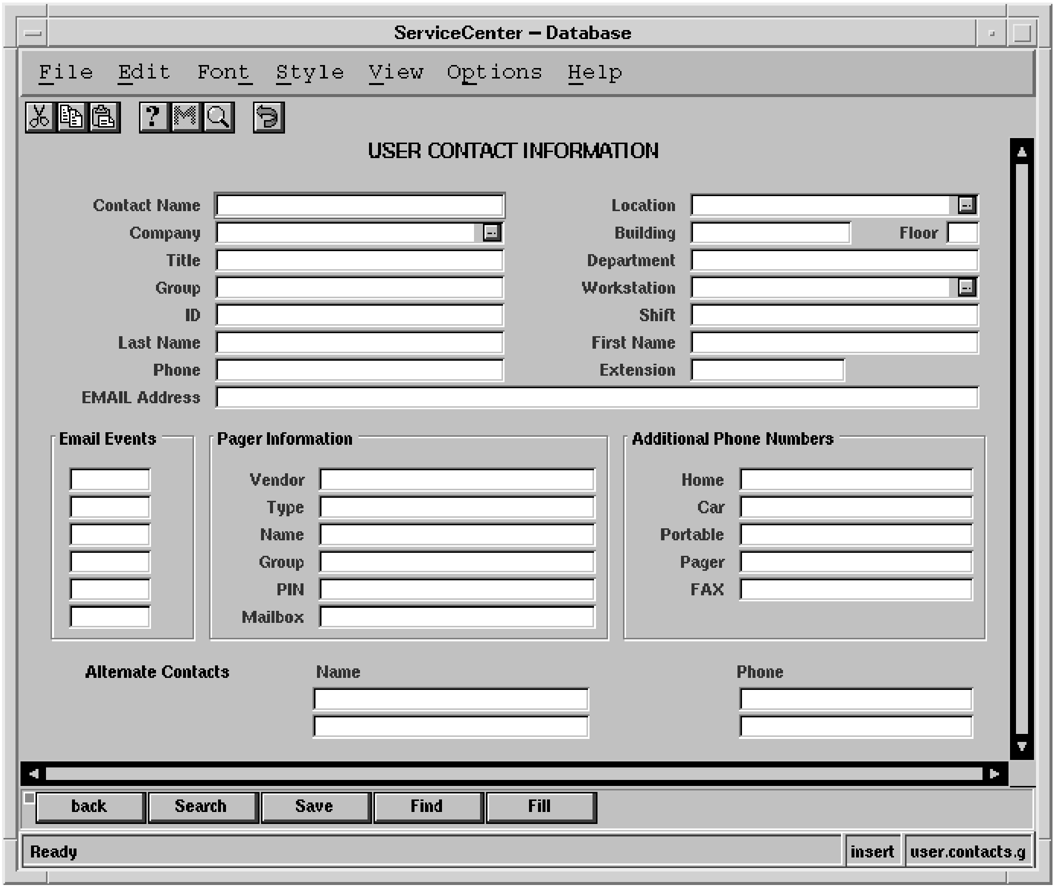


Figure 4-8. User Contact Information Screen

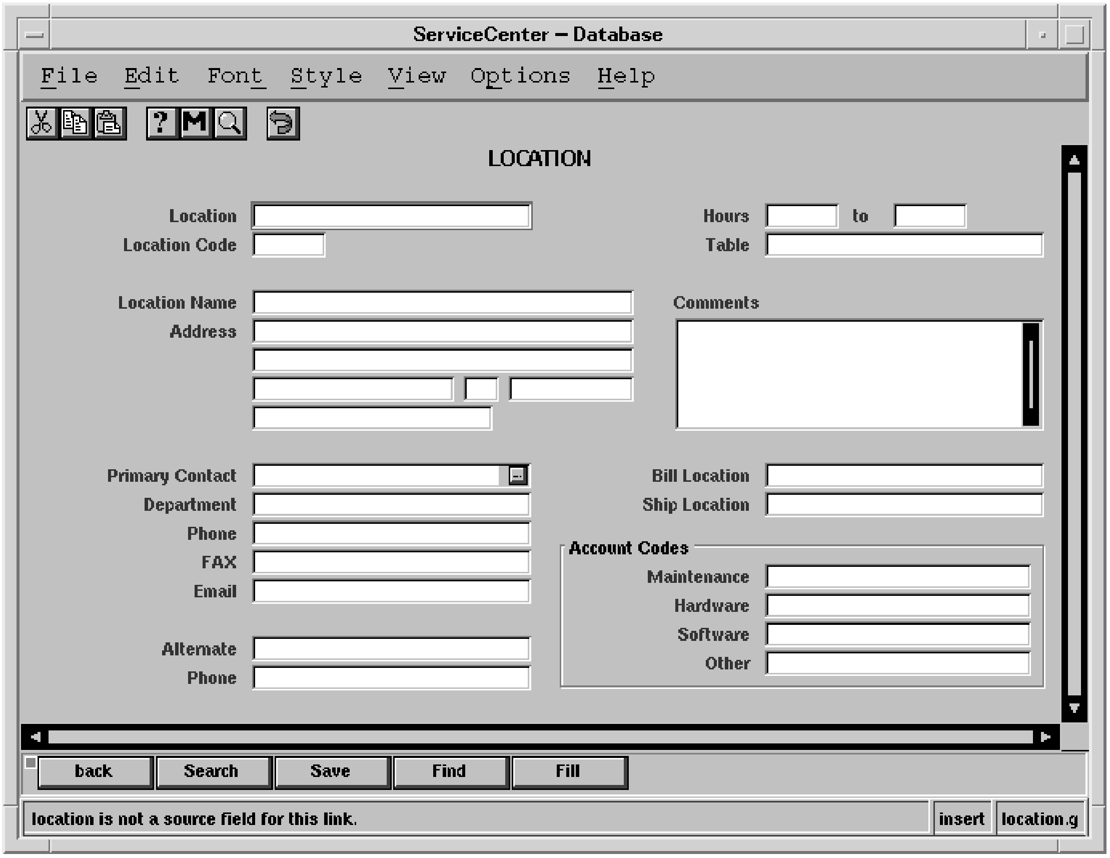
Locations File

This menu option displays location records from the ServiceCenter database, much like an address book. When the **location** screen (Figure 4-9) is first accessed, the screen is blank. To find location data, enter the **location name** and select ***find***.

To view a list of locations, press **Enter** while in the blank screen. A summary list is displayed. Double-click on the desired location to see the data for that location. The location screen also allows the user to search, edit, add to and update location information.

The option for the buttons at the bottom of the screen are also available through the function keys or the **Options** popup menu:

- **Enter**
- **Search**
- **F2 Save**
- **F3 Back**
- **F8 Find**
- **F9 Fill**.



location is not a source field for this link.

Figure 4-9. Location Screen

Event Services

This option accesses ServiceCenter's Event Services application. From this window (Figure 4-10), you can configure and review ServiceCenter events, such as email, inventory additions and problem updates.

The **Back** option button at the bottom of the screen is also available through the popup menu or by using **F3**.

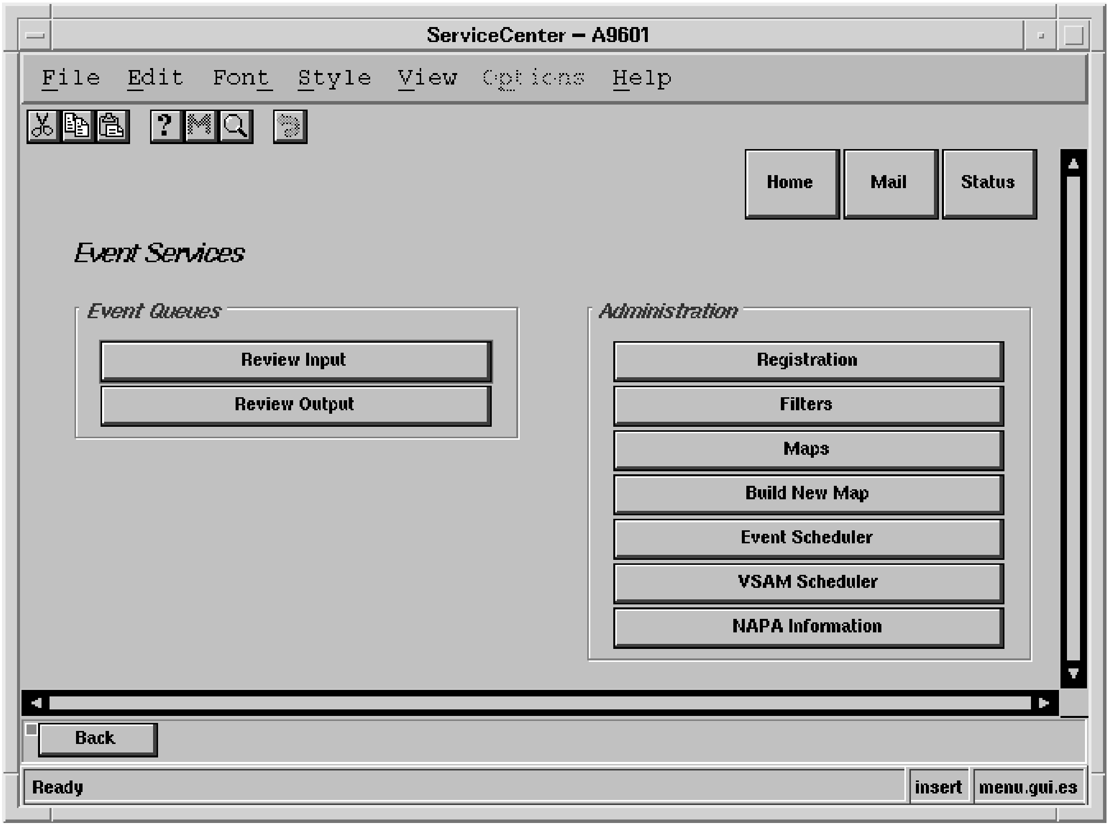


Figure 4-10. Event Services Screen

Probable Problem Cause

This menu option allows you to query ServiceCenter for the probable cause of a problem. When first accessed, a blank *probable cause* screen appears. If you press **Enter**, a *probable cause* list appears. You can select one of the listed probable causes by double-clicking on it.

You can also query ServiceCenter by entering syntax in a field, such as **Key Words**, **Description**, or any of the other fields.

Once a query or selection is completed, a complete *probable cause* screen is displayed (Figure 4-11). The **Resolution** field lists any solution that has been determined for the problem.

The option for the buttons at the bottom of the screen are also available through the function keys or the popup menu:

- **Enter**
- **Search**
- **F2 Save**
- **F3 Back**.

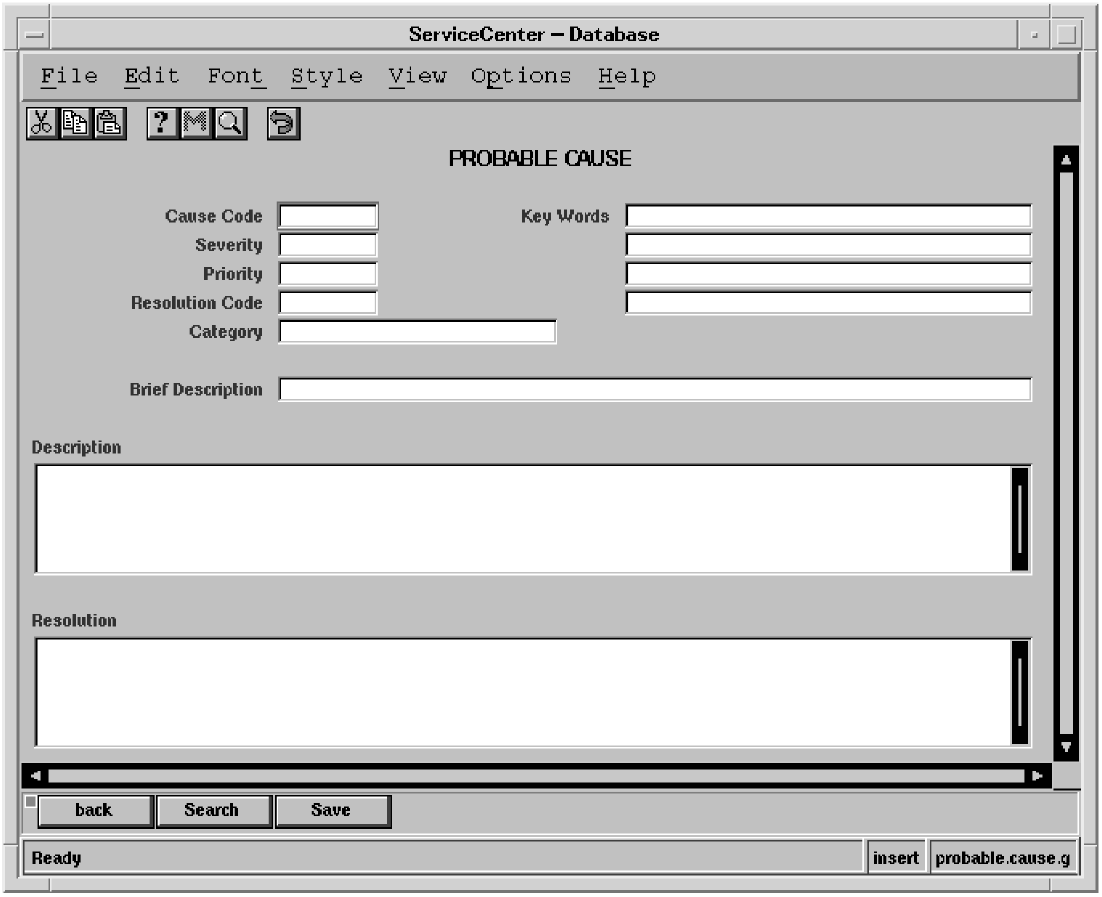


Figure 4-11. Probable Cause Screen

Event Filters

This menu option takes you to the Event Filters screen (Figure 4-12). In this screen, you can set ServiceCenter and SCAuto event filters, or query for an existing filter. All fields in the setup screen are optional, therefore you can either set one field, or all fields, or a combination of fields. This provides flexibility in creating filters. Multiple filters can be set to seek problems under different conditions. Refer to the ***ServiceCenter Event Services Manual*** for more information on the Event Manager application.

The filter setup screen contains these fields.

Event Type Allows you to specify an existing or custom event code to define the filter. ServiceCenter contains eight standard events.

User Name Allows you to specify the user name as defined in the **EV User** field in the event record. A blank user name will match any user name.

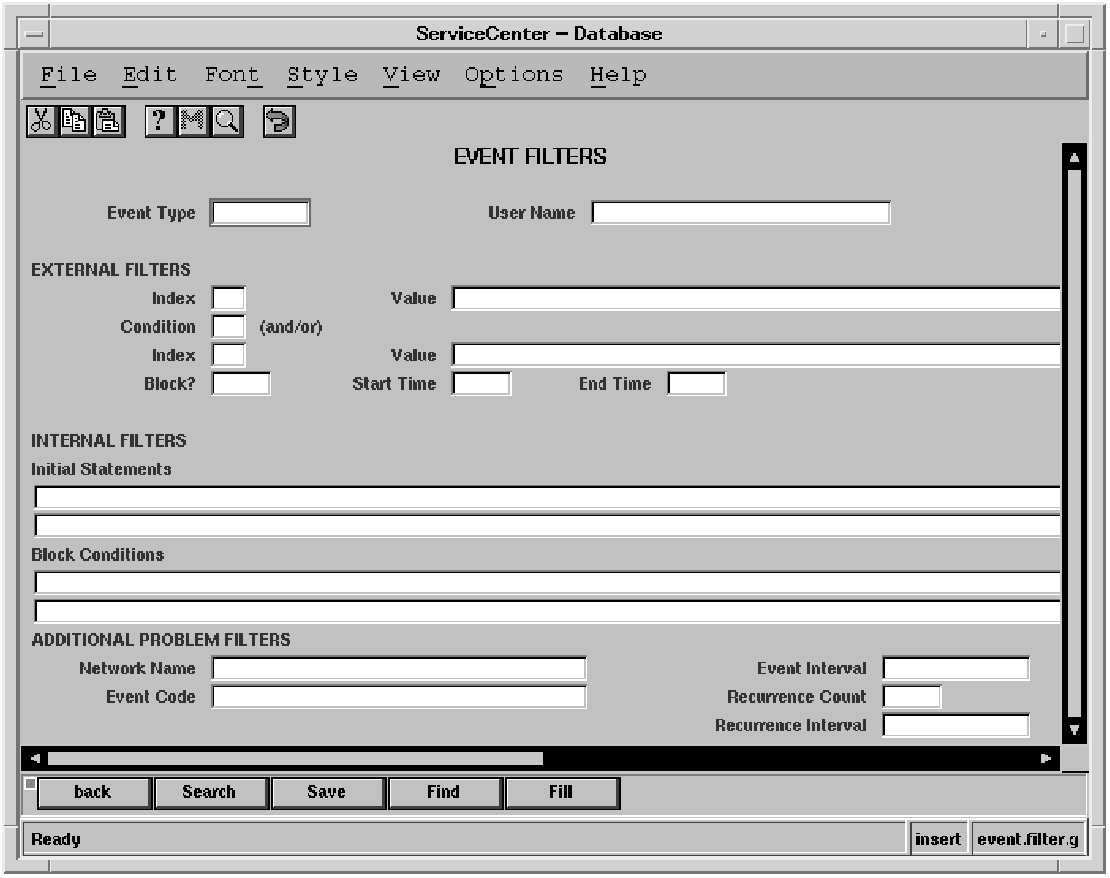


Figure 4-12. Event Filter Screen

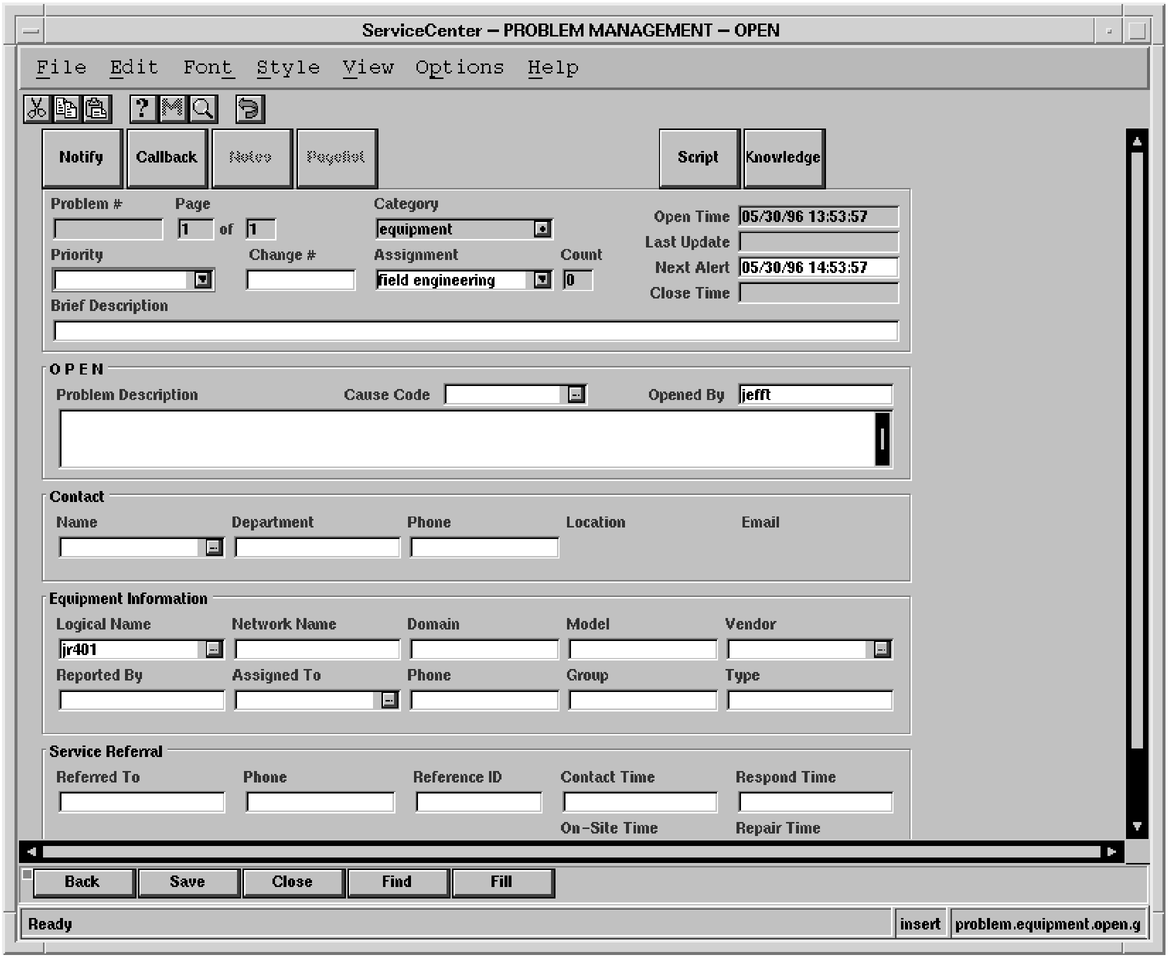
Popup Menu

The popup menu accesses ServiceCenter applications for the selected Spectrum element. Select an element and hold down the second mouse button with the cursor on the icon. Select **Utilities -> SC_HelpDesk** to access the ServiceCenter options.

Open a Problem (Model_Prob_Open)

This menu option allows you to open a problem in ServiceCenter for the selected object. When the option is selected, the *problem open* ServiceCenter screen is displayed (Figure 4-13). The popup menu allows you to open a problem, plus other standard ServiceCenter functions associated with opening a problem.

To exit the screen, select **F3 end**. This takes you to the logout screen. Select **F1 EXIT** then to end this ServiceCenter session.



Page 1 of 1

Category: equipment

Assignment: field engineering

Count: 0

Open Time: 05/30/96 13:53:57

Next Alert: 05/30/96 14:53:57

Opened By: jefft

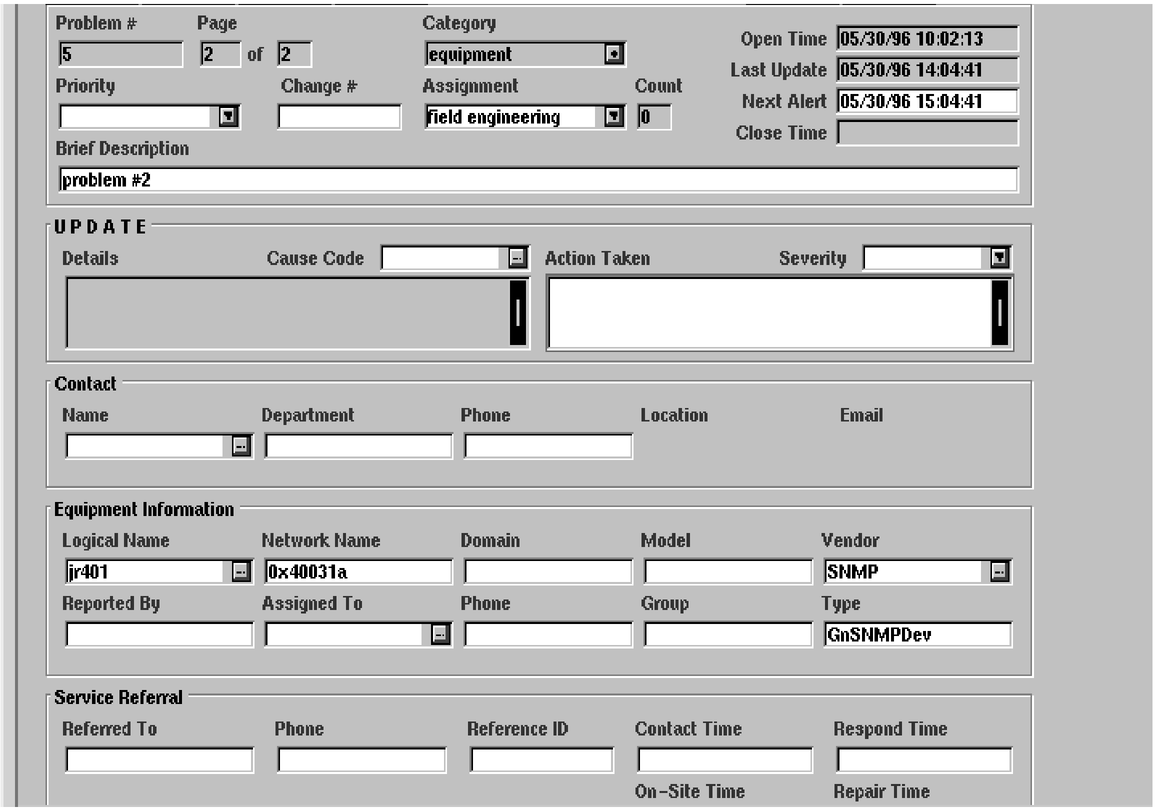
Logical Name: jr401

Ready | insert | problem.equipment.open.g

Figure 4-13. Open a Problem Screen

Update a Problem (Model_Prob_Update)

The ***Update A Problem*** menu option allows you to update an open problem in ServiceCenter for the selected object. When the menu option is selected, a list of ServiceCenter problems for the device (Figure 4-14) is displayed if multiple problems are open for the device. Double click on the desired problem in the list. The *problem.update* ServiceCenter screen is displayed (Figure 4-14). If only one problem has been opened for the selected device, the problem list screen is skipped and the problem update screen is displayed after the menu option is selected. If no problems are open for the selected device, a screen is displayed with a message stating that no problems are open for the device. The problem update screen contains a popup menu that allows you to update a problem, plus other standard ServiceCenter functions associated with problem management.



Problem # 5

Page 2 of 2

Category equipment

Open Time 05/30/96 10:02:13

Last Update 05/30/96 14:04:41

Next Alert 05/30/96 15:04:41

Assignment field engineering

Count 0

Brief Description problem #2

Logical Name jr401

Network Name 0x40031a

Vendor SNMP

Type GnSNMPDev

Figure 4-14. Update a Problem Screen

Close a Problem(Model_Prob_Close)

This menu option allows you to close an open problem in ServiceCenter for the selected object. When the option is selected, the *problem close* ServiceCenter screen is displayed (Figure 4-14). The popup menu allows you to close an open problem, plus other standard ServiceCenter functions associated with problem.

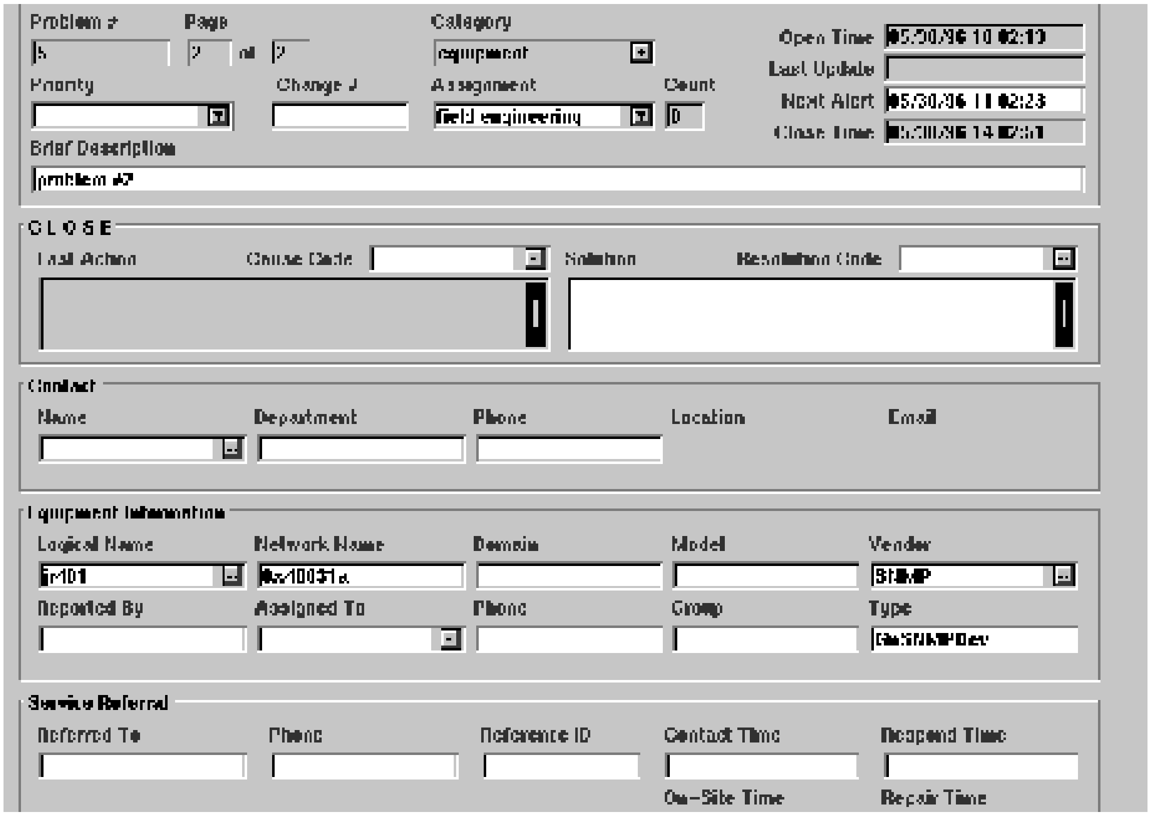
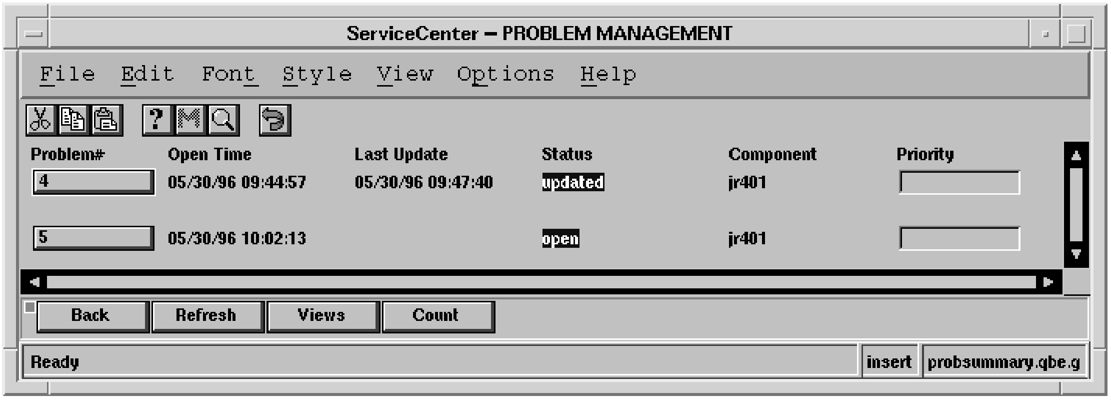

Problem # 5
Page 2 of 2
Category equipment
Open Time 05/30/96 10:02:10
Next Alert 05/30/96 11:02:23
Close Time 05/30/96 14:02:51
Assignment field engineering
Count 0
Brief Description problem #2
Logical Name rr101
Network Name 0x10031a
Vendor SNMP
Type GnSNMPDev

Figure 4-15. Close a Problem Screen

Problem List (Model_Prob_List)

This menu option provides a list of problems open in ServiceCenter for the selected object. When this option is selected, a **problem list** is displayed (Figure 4-16). A popup menu provides options for the problem list.

To exit the screen, select **F3 end**. This takes you to the logout screen. Select **F1 exit** in the logout screen to end this ServiceCenter session.


ServiceCenter – PROBLEM MANAGEMENT
File Edit Font Style View Options Help
Back
Refresh
Views
Count
Ready
insert
probsummary.qbe.g

Problem#	Open Time	Last Update	Status	Component	Priority
4	05/30/96 09:44:57	05/30/96 09:47:40	updated	jr401	
5	05/30/96 10:02:13		open	jr401	

Figure 4-16. Problem List Screen

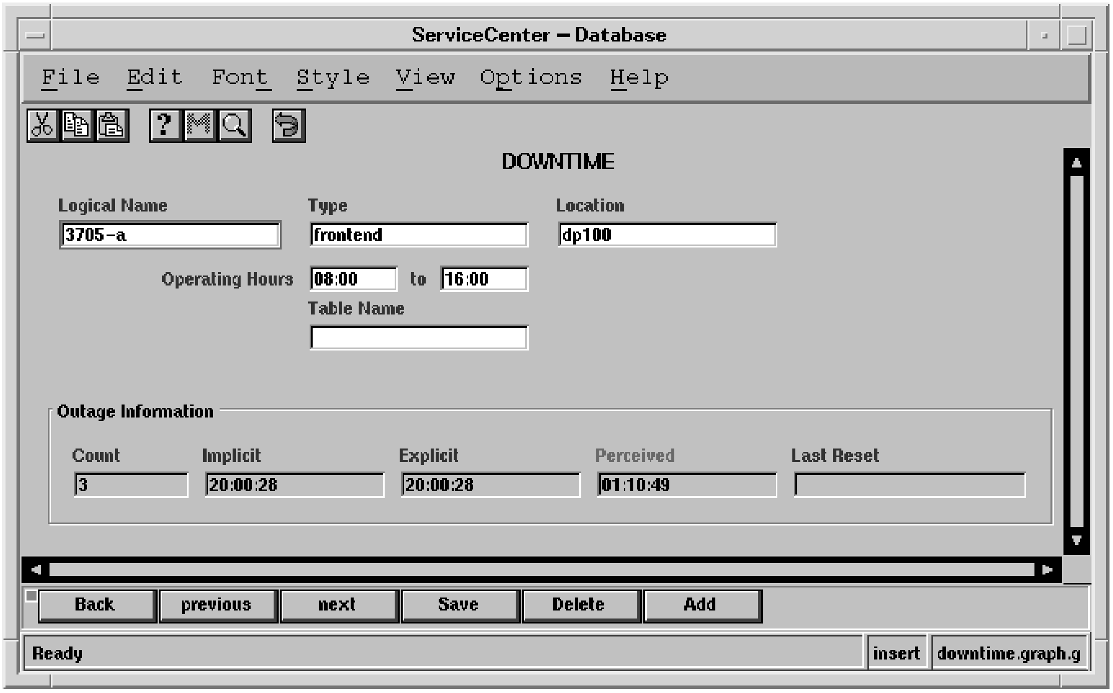
Down Time (Model_Outage)

This menu option provides the availability time for the selected object. When you first access Downtime, a blank **availability** screen appears. Enter the object's logical name in the **Logical Name** field. The **Logical Name** is the same as the object's label on the object on the submap, or the network name assigned to the object. A list of availability times for the named object is displayed.

If you do not know the **Logical Name** of the object, press **Enter** to get a general *availability qbe* list.

Double-click on the desired entry in the list to get the availability information.

Once an availability record is selected, an availability screen appears (Figure 4-17).



DOWNTIME

Logical Name	3705-a	Type	frontend	Location	dp100

Operating Hours 08:00 to 16:00

Table Name

Outage Information

Count	Implicit	Explicit	Perceived	Last Reset
3	20:00:28	20:00:28	01:10:49	

Ready | insert | downtime.graph.g

Figure 4-17. Downtime Screen

Chapter 5 Troubleshooting

SCAuto for SPECTRUM Support Information

This chapter provides the information necessary to obtain Peregrine support for the SCAuto for SPECTRUM product.

General Problem Isolation

Some common problems can occur when executing SCAuto for SPECTRUM for the first time, changing platforms, etc. Use the following suggestions to isolate, or fix your problem prior to contacting Peregrine representatives:

- SCAuto for SPECTRUM ServiceCenter operational function relies on ServiceCenter client facilities in order to function. Run a ServiceCenter client from the <SPECTRUM Root Directory>/scauto directory. This should have been set up during installation through a symbolic link (**ln -s...**) command.

  The ServiceCenter client command is: **scclient login** *userid* **-G**. If the client window appears, you have ServiceCenter client server connectivity. If the window does not appear, your client server specifications may be incorrect. Revalidate your ServiceCenter client installation. If you are still encountering problems, report problem according to ServiceCenter client server reporting procedures.
- Check the ss*IPASd.log, ssIPASd.err, ssAlarm.log, ssAlarm.err* for any error messages.
- SCAuto requires the SCAuto base be up and running on the ServiceCenter server platform and its service name specified in /etc/services.
- Ensure permissions on <SPECTRUM Root Directory>/scauto are consistent with the execution user, group. Also, ensure this directory is in your PATH environment variable.
- Check *ssIPAS_alarms*. The *ssIPAS_alarms* should contain all log records from the SPECTRUM alarms that were not filtered by your alarm filters. Remember filters are inclusive. Ensure the ssALARM daemon is active and that it has permissions to <SPECTRUM Root Directory>/scauto.

- In some cases, system messages useful for debug are lost when SCAuto for SPECTRUM daemons are executed in the background. Run the daemons under a user with proper permissions as follows:
  <SPECTRUM Root Directory>/scauto/ssIPASd& (SCAuto for SPECTRUM ServiceCenter services daemon)
  – or –
  <SPECTRUM Root Directory>/scauto/ssAlarm (SCAuto for SPECTRUM trap daemon)

Contacting Peregrine Systems

Peregrine Systems Inc. provides support for all SCAuto for SPECTRUM users. Before contacting Peregrine Customer Support, review the following section, *Obtaining Required Data/Information*, to see if additional data is required to help diagnose the problem.

You can contact Peregrine Systems support as follows:

- For SCAuto for SPECTRUM problems or information that is needed immediately, call Peregrine Customer Support at (800 638-5231 or (619) 431-2400.
- For questions or information regarding SCAuto for SPECTRUM, use a written FAX or email.
- Send all SCAuto for SPECTRUM FAXes to (619) 431-0696.
- For information that was requested of your installation that is on tape, cartridge, etc., send to:

  Peregrine Systems Inc.
  attn: SCAuto for SPECTRUM Support
  12670 High Bluff Dr.
  San Diego, CA 92130

Obtaining Required Data and Information

This section provides detailed instructions for gathering data and information needed for the Peregrine support staff to resolve your problem in the most efficient manner possible.

Environmental Information:

- ServiceCenter Release
- SCAuto for SPECTRUM Release
- Operating System Release (i.e., Solaris, SPECTRUM Mgr)

- Type of hardware base SCAuto for SPECTRUM is running on
- Any error messages or error logs.

Error logs and files that the Peregrine support staff needs are listed on the next page.

The Peregrine Support staff can utilize the following error logs and files to resolve an SCAuto for SPECTRUM problem:

- *<SPECTRUM Root Directory>/scauto/ssIPAS_chkpt*
- *<SPECTRUM Root Directory>/scauto/ssIPAS_alarms*
- *<SPECTRUM Root Directory>/scauto/ssIPASd.log*, *ssIPASd.err*, *ssAlarm.log* and *ssAlarm.err*
- If the problem resulted in a core dump, the resulting core file is helpful in determining the problem.

Error Messages

Messages printed to <SPECTRUM Root Directory>/scauto*/ssIPASd.log* are formatted as follows:

<time> <module> <action code>: <message>

where:

<*time*> - is the time when the message was printed (using the local time).

<*module*> - is the program module causing the message.

<*action code*> - is one of the following single letter codes:

A - operator action may be required.

W - warning message.

I - informational message only.

C - checkpoint record.

Error Return Codes

The following error return codes may be indicated in a message:

101...199

Indicates a client/server problem with ServiceCenter. Verify that ServiceCenter is operating and is the same release level as SCAuto for SPECTRUM.

800

No connection established. Could indicate that initial connection to SCAUTO base failed. Verify SCAUTO base is running.

804

An error was printed to standard error. These messages are only displayed if you ran the daemons manually.

807

Malloc failed. Check the standard error.

811

Error opening communication file. Check standard error.

814

Error adding event to ServiceCenter. Internal error.

Other errors indicate a systems error. If these occur, gather the necessary information and log files (*ssIPASd.log*, *ssIPASd.err*, *ssAlarm.log*, *ssAlarm.err*, ss*IPAS_chkpt* and *ssIPAS_alarms*) then contact Peregrine Customer Support.

Sending Files to Peregrine

Files may be sent to Peregrine Systems either via magnetic tape or electronic mail. If multiple files are being sent, store the files in the *tar* format. Compressed files are acceptable.

Files can be sent on 8mm or 4mm magnetic tapes. If you are sending files with electronic mail, use the UNIX **uuencode** utility. Other formats or methods of transport must be arranged with Peregrine Customer Support.

Index

Peregrine
SYSTEMS®

The Infrastructure Management Company™