ServiceCenter®

SCAuto for Tally NetCensus

Version 1.0

April 2000

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

Peregrine
S Y S T E M S ®

The Infrastructure Management Company™

Contents

Preface

Chapter 1   Introduction

Chapter 2   Installation

Chapter 3   Configuration

Chapter 4   ServiceCenter Customization

Chapter 5   Customization

Chapter 6　Scripting

Chapter 7　Operation

Index

Preface

Overview

Welcome to Peregrine Systems *SCAuto for Tally NetCensus*. This product allows for automatic collection and maintenance of inventory records in the ServiceCenter database, for network as well as locally attached devices.

This product is part of the suite of SCAutomate (SCAuto) interface products that integrate ServiceCenter with premier network and systems management tools. This guide describes how to implement the SCAutomate interface for integration with Peregrine Systems' ServiceCenter.

Additional information about SCAuto can be found in the *SCAuto Applications for Windows NT and UNIX* guide.

Prerequisite knowledge

It is recommended that implementers of SCAuto for Tally NetCensus have specific knowledge in the following areas:

- Experience installing and customizing Tally NetCensus.
- Knowledge of ServiceCenter and the operating system platform on which it is installed.
- Experience customizing ServiceCenter, or the equivalent training.
- Knowledge of ServiceCenter Event Services.

Contacting Peregrine Systems

Contact one of the Peregrine Systems Customer Support offices listed here if you have questions about, or problems with, ServiceCenter or SCAuto for Tally NetCensus.

For more information about Customer Support, check the support web site: **http://support.peregrine.com** Please contact Customer Support for an account on this site.

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

North and South America

To get help immediately, call Peregrine Customer Support at:

> **(1) (800) 960-9998** (North America only)

> **(1) (858) 794-7428** (North and South America)

For information or answers to questions about ServiceCenter or SCAuto for Tally NetCensus, send a fax or e-mail to:

> Fax: **(1) (858) 794-6028**
> E-mail: **support@peregrine.com**

Send materials that Peregrine Systems Customer Support requests to:

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

Note: Countries outside North and South America are covered by regional offices. Customers should contact the regional office under which their country is listed.

United Kingdom regional office

Great Britain, Greece, and South Africa

Peregrine Systems Ltd.
1st Floor
Ambassador House
Paradise Road
Richmond, Surrey, Great Britain, TW9 1SQ
Phone: **0800 834770** (toll free)
or: **0181 334 5844**
E-mail: **uksupport@peregrine.com**

France regional office

France, Spain, Italy, Greece, and Africa (except South Africa)

Peregrine Systems
Tour Franklin-La Défense 8
92042 Paris La Défense Cedex, France
Phone: **+33 (0) (800) 505 100** (International Toll Free)
E-mail: **frsupport@peregrine.fr**

Germany regional office

Germany and Eastern Europe

Peregrine Systems GmbH
Bürohaus Atricom
Lyoner Strasse 15,
60528 Frankfurt, Germany
Phone: **0049 6966 8026917**
or: **0800-2773823** (in Germany only)
E-mail: **psc@peregrine.de**

Nordic regional office

Denmark, Norway, Sweden, Finland, and Iceland

Peregrine Systems A/S
Naverland 2, 12 SAL
DK-2600 Glostrup
Denmark

Denmark	Phone:	(**+45) 80307676**
Sweden, Norway, Iceland and Finland	Phone:	**(+45) 77317776**
	E-mail:	**nordic@peregrine.com**

Benelux regional office

Netherlands, Belgium, and Luxembourg

Peregrine Systems BV
Botnische Golf 9a
3446 CN Woerden
Netherlands

Netherlands	Phone:	**0800 0230 889** (toll free in the Netherlands)
Belgium and Luxembourg	Phone:	**00800 7474 7575**(toll free in Belgium and Luxembourg)
	E-mail:	**benelux.support@peregrine.com**

Asia-Pacific regional offices

Australia, Hawaii, Hong Kong, Japan, Korea, Malaysia, New Zealand, Singapore

AustraliaPhone:**(800) 146-849**

HawaiiPhone:**(1) (800) 960-9998**

Hong KongPhone:**(800) 908056**

JapanPhone:**(0044) 221-22795**

SingaporePhone:**(800) 1300-949** or **-948**

E-mail:**apsupport@peregrine.com**

Chapter 1   Introduction

Overview

SCAuto for Tally NetCensus is a uni-directional gateway between ServiceCenter and Tally System's NetCensus, providing automatic migration of information from NetCensus to ServiceCenter. Running as a Windows NT service, it uses Component Object Model (COM) objects and ActiveX scripting to provide maximum flexibility and extensibility.

Using Microsoft ADO, SCAuto for Tally NetCensus selects information from the NetCensus database using standard SQL.

To provide maximum flexibility and ease of customization, the SCAuto for Tally NetCensus product uses state-of-the-art Windows technologies:

- OLE Automation, which defines dynamic COM interfaces to permit scripting languages to use COM objects

- The Microsoft JScript scripting language, which provide rapid application development

SCAuto for Tally NetCensus leverages these technologies to provide a robust, user-tailorable gateway without the inflexibility normally associated with C/C++ applications.

Theory of Operation

SCAuto for Tally NetCensus is designed to propagate information automatically from NetCensus to ServiceCenter. It works on demand by selecting information from the NetCensus database using Microsoft ADO and then converting the information received into ServiceCenter event messages to update the ServiceCenter repository.

SCAuto for Tally NetCensus does not update the NetCensus database in any way. It requires only shared read-only access to the database in order to select the data.

Extensibility

By modifying the script, the SCINFO mapping table and ServiceCenter's Event Services registrations and mappings, additional ServiceCenter fields can be populated with any data contained in the NetCensus database. Propagation of Inventory information must comply with ServiceCenter workflow rules; as a result, some modification of data policy may be required. In addition, data contained in the NetCensus database that has no corresponding repository definition in ServiceCenter can be transferred simply by extending the ServiceCenter database dictionary definition.

Although SCAuto for Tally NetCensus is delivered with a script, the gateway is not limited to this script. The gateway can be customized to run different scripts or additional scripts, simply by changing the *sctally.cfg* file.

Architecture

Note the following:

- There are three major components—ServiceCenter, NetCensus, and SCAuto for Tally NetCensus (the gateway between the two).

- Each component can be installed on the same or a different server.

- Communication between the SCAuto for Tally NetCensus gateway and ServiceCenter is based on the SCAuto API, which uses TCP (see chapter 5).

- Communication between the SCAuto for Tally NetCensus gateway and the NetCensus database is accomplished using Microsoft ADO, which uses ODBC.

- ServiceCenter can be running on any supported OS platform (Windows NT, UNIX or MVS.

- The SCAuto for Tally NetCensus gateway requires a Windows NT server platform. It does not run on UNIX.

Product Components

SCAuto for Tally NetCensus provides a set of OLE Automation objects and redistributes the latest version of the Microsoft Data Access objects, including ADO. These objects are then exploited by some JScript routines (source code provided) which implement a mapping between the NetCensus database and the ServiceCenter repository.

SCAuto for Tally NetCensus has a modular architecture which includes the following building blocks:

- **SCEVENTS.DLL** - an OLE Automation object which encapsulates SCAUTO.DLL, the ServiceCenter API, which uses TCP sockets to communicate with ServiceCenter.

- **SCRUTIL.DLL** - an OLE Automation object which provides utility functions for logging messages, obtaining run-time parameters, etc.

- **SCTALLY.JS** - an ActiveX script written in JScript which propagates information from NetCensus to ServiceCenter by performing these steps:

  - Extract information from NetCensus by querying the database via ADO using SQL select statements embedded in the JScript code

  - Map the NetCensus data from ADO objects into a ServiceCenter event object

  - Send the ServiceCenter event data to ServiceCenter

- **WSCRIPT.EXE** - The Microsoft Windows Scripting Host.

- **JSCRIPT.DLL** - Microsoft 5.0 ActiveX scripting engine.

- **SCEVMON.EXE** - an daemon which manages communication of ServiceCenter events between SCAuto for Tally NetCensus and ServiceCenter. It provides event message queuing so that ServiceCenter can be started and stopped without affecting the operation of the gateway.

- **SCTALLYSRV.EXE** - A Windows NT service to run the above described components continuously in the background.

Customization

Customization of the SCAuto for Tally NetCensus gateway is accomplished by modifying the supplied SQL mapping tables and/or ServiceCenter database schemas and event mapping definitions. The SCINFO table supplies information necessary to map NetCensus information to ServiceCenter's data repository based upon instructions provided in ServiceCenter's *eventmap* table.

It should not be necessary for you to modify the NetCensus scripts except to extend the mapping scope.

Chapter 2   Installation

Overview

To operate SCAuto for NetCensus, the following products must be installed:

- ServiceCenter
- NetCensus Manager 2.93
- SCAuto for NetCensus
- Windows Script 5.0
- Microsoft MDAC 2.1 GA SP 2

This document provides installation instructions for only the last three items in the list above. It is assumed the first two items (ServiceCenter and NetCensus) are already installed on your system. If not, be sure to install them before proceeding with the instructions provided in this document.

Installation checklist

Verify the following items before proceeding:

- ServiceCenter 2.1 SP 3E2 or later RTE is installed on your system, and all Event Services applications are at the same level.
- NetCensus 2.93 is installed on your system and you know the what folder includes the *.dbf files.
- ServiceCenter is properly configured to run Event Services, including the SCAUTOD server.
- You have collected the information needed to configure the gateway:
    - ServiceCenter hostname
    - TCP port number of the ServiceCenter SCAUTOD server
    - NetCensus database information described immediately above

Installing SCAuto for Tally NetCensus

To install SCAuto for Tally NetCensus:

1. Insert the SCAuto for Tally NetCensus CD and run *setup.exe*.

    The Setup program runs until the Welcome screen is displayed.
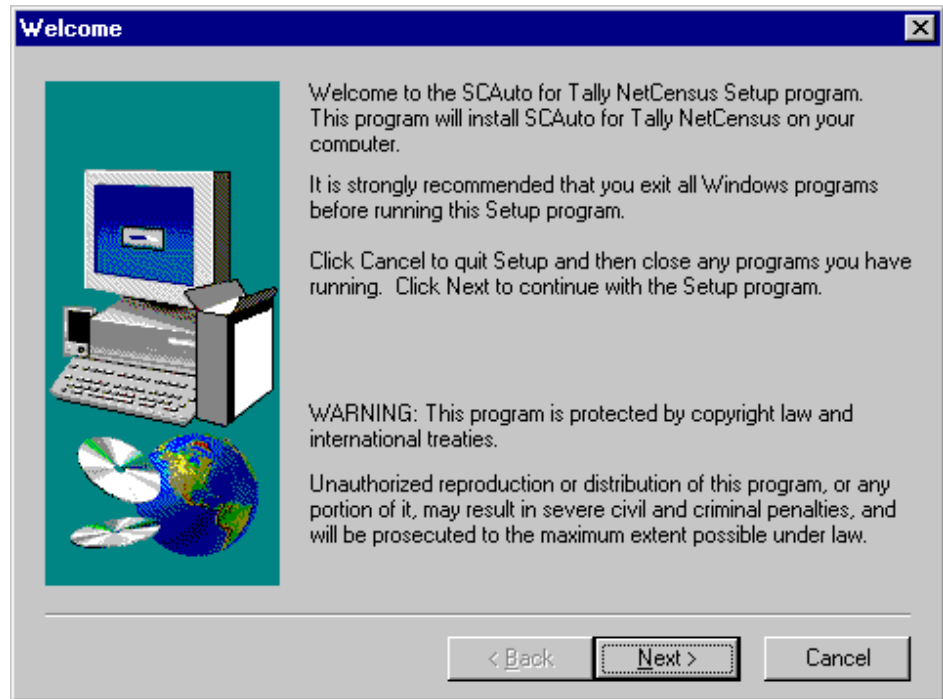
Figure 2-1. Welcome screen

2. Click **Next**.

    The Readme Information screen is displayed.

3. Click **Next**.

The User Information screen is displayed.
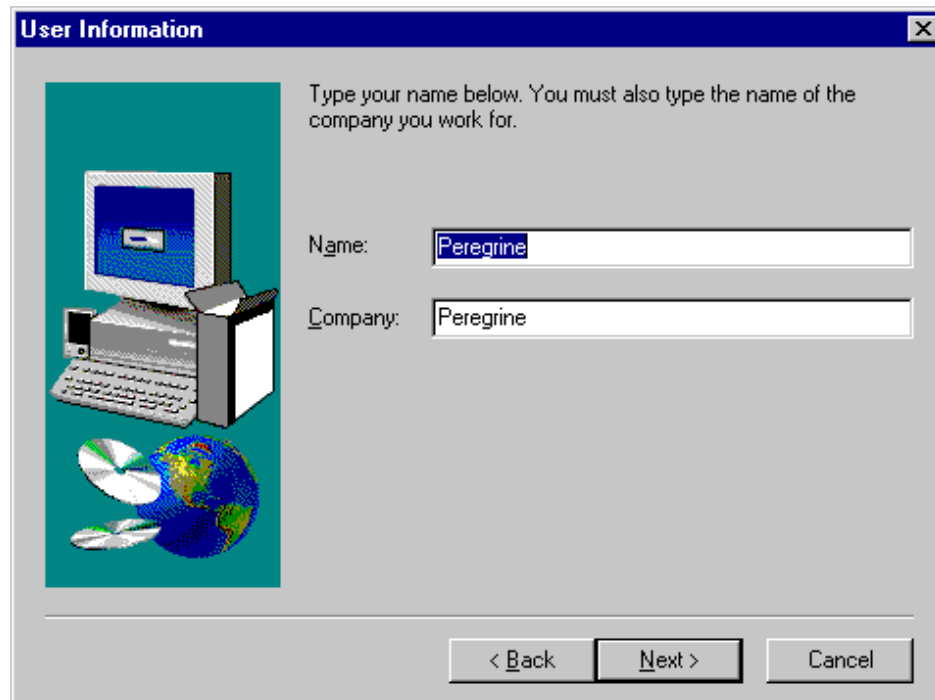
Figure 2-2. User Information screen

4. Type your name and your company name, then click **Next**.

The Choose Destination Location screen is displayed, which shows the directory to which SCAuto for Tally NetCensus will be installed.
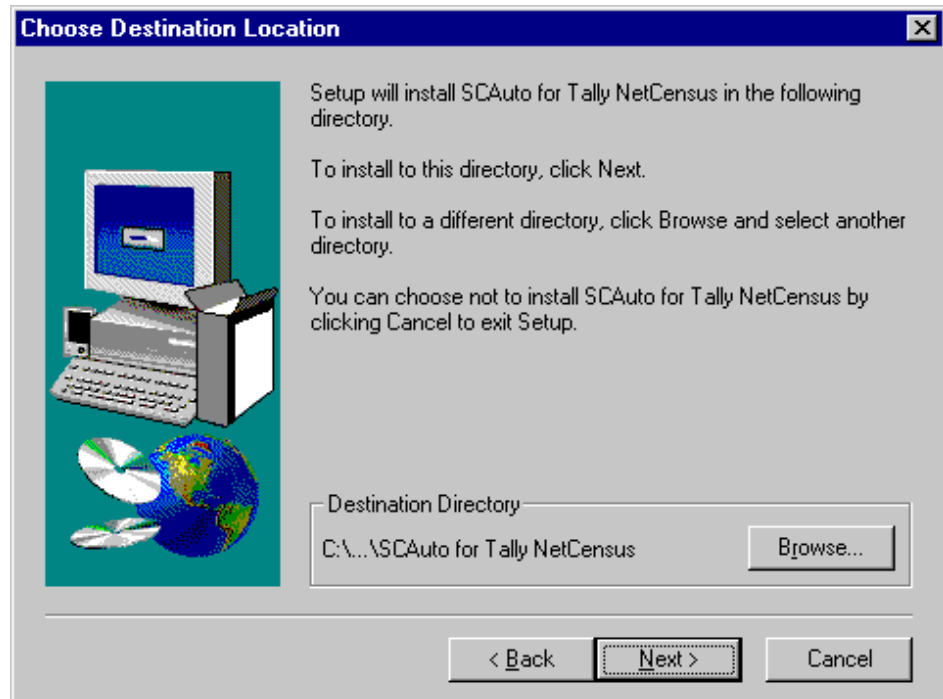
Figure 2-3. Destination Location screen

5. Ensure the Destination Directory is correct.

   • If so, click **Next**.

   • If not, click **Browse** and select a different directory. Then click **Next**.

The Select Program Folder screen is displayed, where you select the program folder to which you want program icons added.
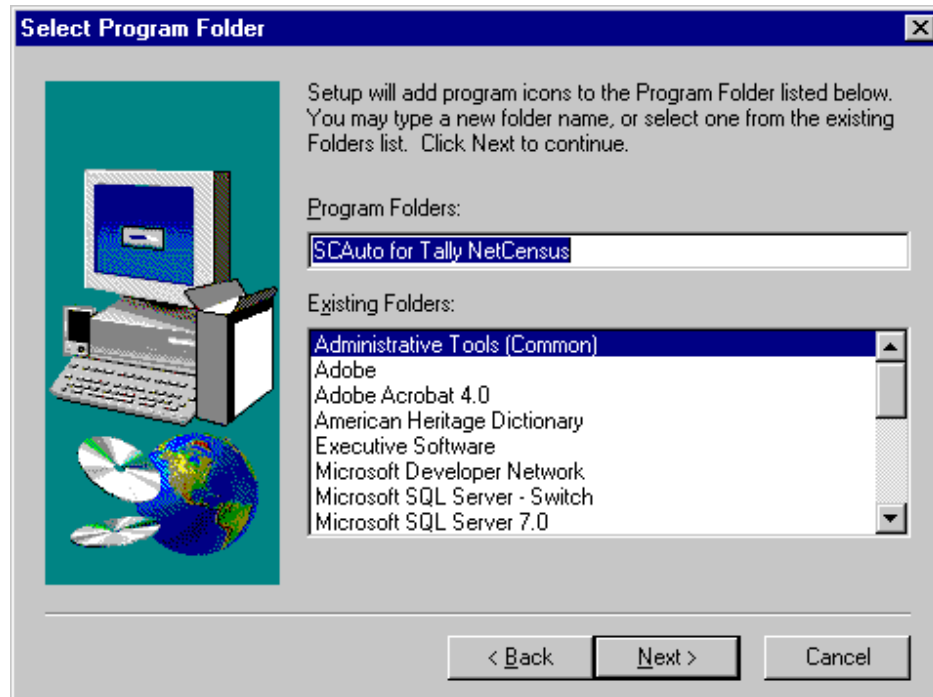
Figure 2-4. Select Program Folder screen

6. Select a program folder for your SCAuto for NetCensus system, then click **Next**.

The Start Copying Files screen is displayed, indicating that Setup is ready to start copying the SCAuto for Tally NetCensus program files.
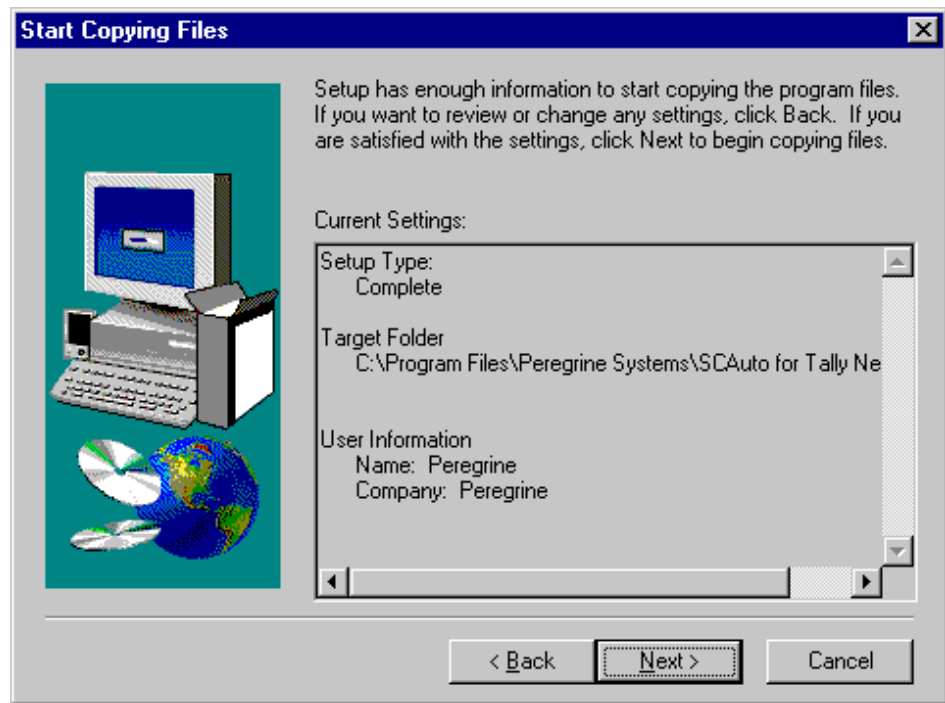
Figure 2-5. Start Copying Files screen

7. After verifying that the current settings displayed are correct, click **Next** to start copying the files.

The Start Copying Files screen is displayed, indicating that Setup is ready to start copying the SCAuto for Tally NetCensus program files.
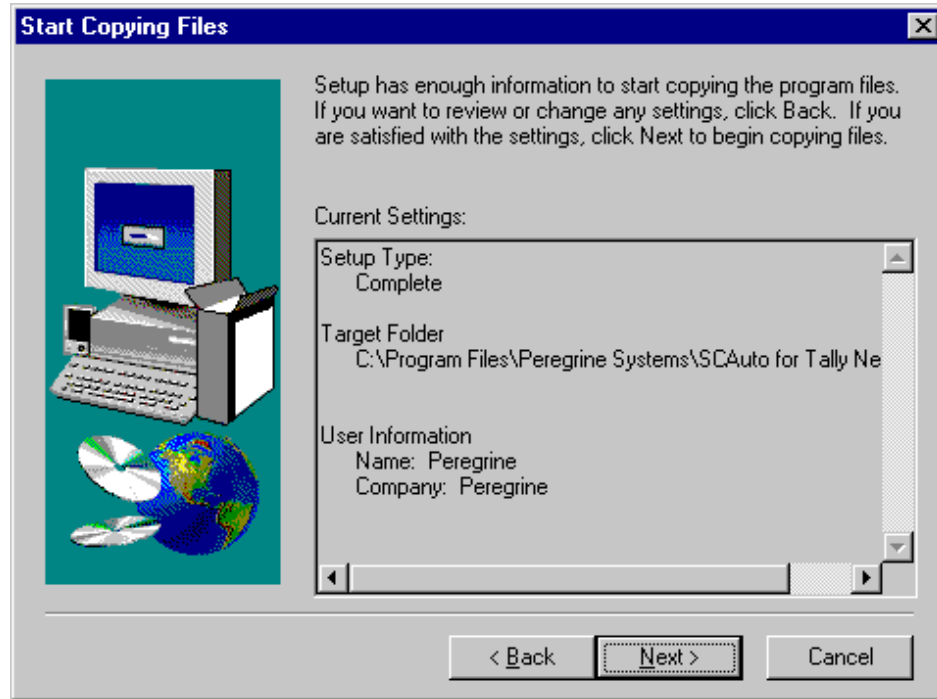
Figure 2-6. Start Copying Files screen

After all files are installed on your system, a dialog is displayed allowing you to edit your *sctally.ini* file settings. Refer to the provided Help or the section on the *sctally.ini* file in Chapter 3 of this guide for further information.
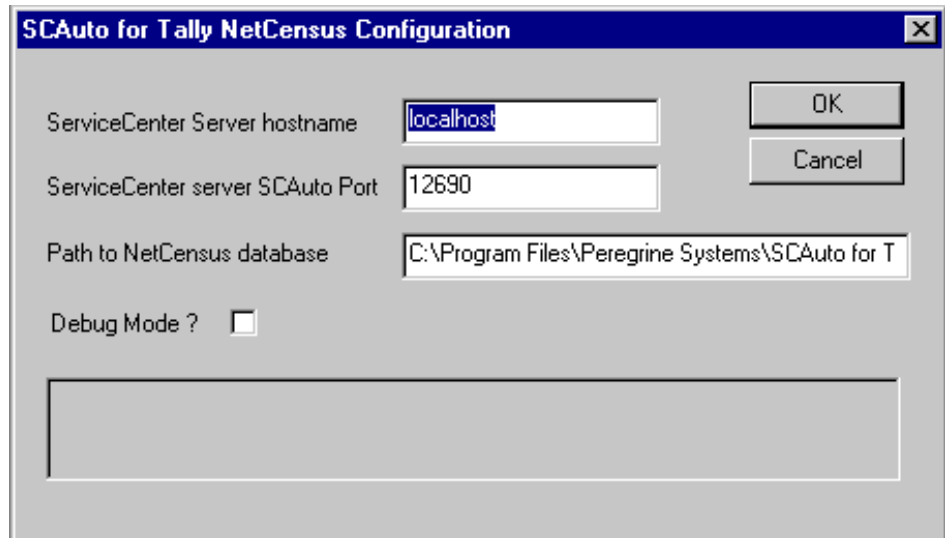
Figure 2-7. Configuration screen

8.   Ensure all settings are correct, then click **Next**.

Note: These settings can be changed later by selecting **Programs>SCAuto for NetCensus>Edit SCTALLY.INI** from your Windows Start menu.

The Setup Complete screen is displayed when Setup has finished installing SCAuto for Tally NetCensus on your computer.
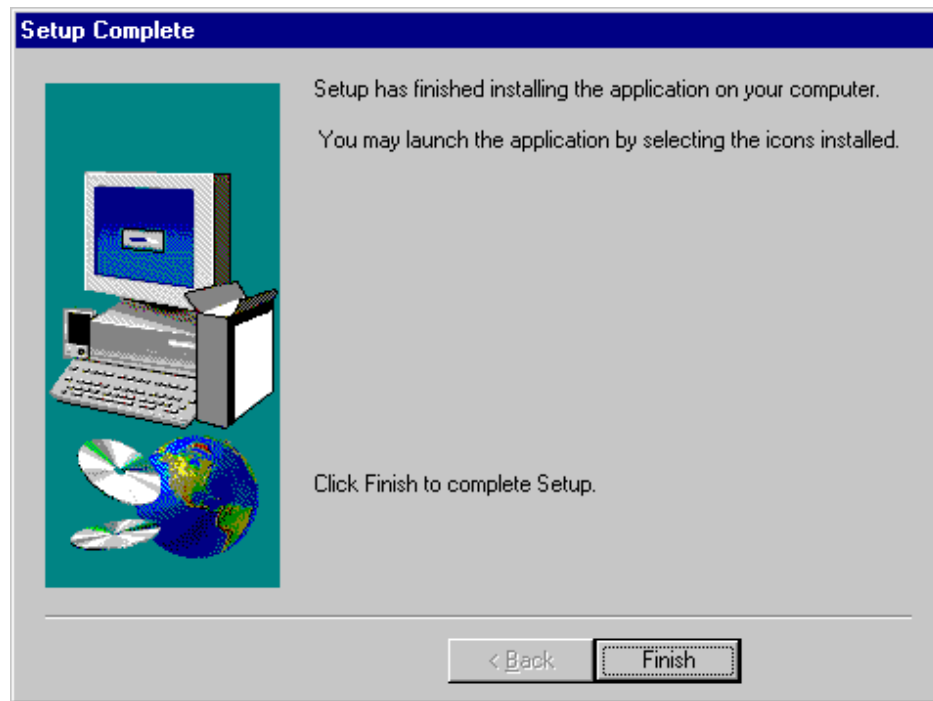
Figure 2-8. Setup Complete screen

9. Click **Finish** to exit.

   SCAuto for Tally NetCensus is now installed.

10. After finishing the installation and configuration of all components, refer to Chapter 7 of this guide for further information on operating your SCAuto for Tally NetCensus system.

Note: If you already have your own custom script, you will want to copy this into your installation*bin* directory.

Windows Script 5.0

SCAuto for NetCensus requires Windows Script 5.0, which includes Windows Scripting Host and the 5.0 scripting engines, including JScript 5.0.

At the conclusion of the setup procedure described above, the Microsoft setup procedure for Windows Script 5.0 (US English version) will be automatically launched. If you wish to obtain and install a more current version or a version for a different locale, cancel this part of the setup and then do the following:

1. Access the web site http://www.microsoft.com/msdownload/vbscript/scripting.asp

2. Despite the *vbscript* in the URL, this takes you to the Windows Script page, which invites you to download **ste50en.exe** (900K) which is the US English version, or an equivalent localized version, such as **ste50de.exe**, or **ste50fr.exe**. The download includes Windows Script 1.0, JScript 5.0, VBScript 5.0 and other required files.

3. Download the appropriate Windows Script file and install it. You must be running version 5.0 or later of the Windows Scripting engines to successfully run the Microsoft JScript code provided with SCAuto for NetCensus.

4. While you are here, you may wish to download the free script debugger, unless you have Visual Studio Professional or Enterprise Edition, which includes an enhanced debugger.

5. We also recommend you download the Microsoft documentation package for JScript 5.0

Microsoft MDAC (ADO)

SCAuto for NetCensus also requires MDAC 2.1 GA SP 2 or later.

At the conclusion of the Windows Script part of the setup, the Microsoft setup procedure for MDAC 2.1 GA SP 2 will be automatically launched.

Loading ServiceCenter Data

Two small unload files are included with the installation of SCAuto for Tally NetCensus. Both files must be loaded into your ServiceCenter system before starting the SCAuto service.

sctally.unl

This file contains SC Event registration and mapping information, as well as a new database descriptor, several forms and a link record. Check the list of records below and, for those that exist in your database, rename them before loading this file. Be sure to rename the *pcfiles* table in ServiceCenter.

SC Table Name	Primary Key	Description/Change
eventregister	DBApcfiles	This record registers the new event type.
eventmap	tally pcfiles	These records provide mapping for the new DBApcfiles event.
format	device.pc.g	The table on the Hardware tab has been modified.
	pc.software.files	The form references the *pcfiles* rather than the *pcsoftware* file.
	pc.software.files.g	New fields have been added.
	software.vj.g	New fields have been added.
	pcfiles.examine	New form provides a view of all fields in *pcfiles*.
link	device.pc	References a different file for the software virtual join.
dbdict	pcfiles	Adds additional fields and keys.

axcesinv.unl

This file contains a change to the RAD application *axces.inventory*. With this change, when an inventory event cannot be added to or updated in the ServiceCenter database the application will exit and notify you by placing error in the status (*evstatus*) field of the eventin record. This makes it easier to select records that were processed abnormally, and prevent those records from being deleted if the Registration's delete flag is enabled. Be sure to rename the existing *axces.inventory* RAD routine to *axces.inventory.<yourname>* before loading this file.

Chapter 3   Configuration

Overview

To configure SCAuto for Tally NetCensus, you may need to make changes to the following files:

sctally.ini

Changes to this file are rarely needed once a few key parameters are correctly set. These include the identities of the ServiceCenter and Tally NetCensus servers, and the userid and password required for connection to the Tally NetCensus database. The latter parameters should have been correctly set when you ran the configuration dialog at the end of the installation process. If you need to change those, it is recommended you rerun the configuration dialog rather than hand-edit the sctally.ini file. However, you may need to modify the sctally.ini file by hand for certain things, such as activating debug logging. A discussion of the required sctally.ini file parameters follows. A complete list of *.ini* parameters is also given in Appendix A.

sctally.cfg

Changes to the this file should not be necessary. An exception might be the specification of some transient NT command you wish executed at service start-up, or specification of additional scripts to be run as part of the gateway.

sctally.ini File

The following parameters in the *sctally.ini* file may need to be configured:

sessid:SC_TNC

Unique identifier given to this instance of SCAuto for Tally NetCensus.

sceventserver:localhost.12690

The *sceventserver* parameter defines the TCP/IP hostname and TCP port number of the ServiceCenter SCAUTOD server. The two values must be specified together, separated by a period. The SCAUTOD TCP port number must not be confused with the *other two TCP port numbers* used by ServiceCenter (express client server and full client server). The value *localhost* in this example indicates that the ServiceCenter server is running on the same machine as the gateway. The *12690* value is the default TCP port used by the SCAUTOD server in ServiceCenter.

stopeventname:SCTNC.StopEvent

The *stopeventname* parameter specifies the name of the Win32 event to be used to synchronize SCAuto NetCensus processes.

noeventsfromsc:1

The *noeventsfromsc* parameter, not implemented in version 1.0, specifies whether to ignore events sent by ServiceCenter. When scripts are provided to implement a bidirectional dialog between Tally NetCensus and ServiceCenter, set this parameter to 0 to enable the dialog.

logmaxlen:5000000

The *logmaxlen* parameter limits the size of the message log file (defined in the log parameter) to the number of bytes specified. When the limit is reached, first-in messages are removed from the top of the file.

eventlogmaxlen:5000000

The *eventlogmaxlen* parameter limits the size of the *scevents.to.<server>.<port>* log file where events are logged to the number of bytes specified. When the limit is reached, any events not yet sent to ServiceCenter are copied to a temporary file, the original file is removed, and the temporary copy is renamed back to *scevents.to.<server>.<port>*. If the file is still too large, the event will be discarded and the process will terminate.

sleep:3600

The *sleep* parameter specifies the interval in seconds between Tally NetCensus database polls.

retrymax:3

The *retrymax* parameter specifies the number of times the SCAuto service will attempt to restart a failing process before giving up and shutting down the service.

lastrundate:0

The *lastrundate* parameter specifies the interval in seconds between Tally NetCensus database polls.

tally_dbpath:

The *tally_dbpath* parameter defines the path to the Tally NetCensus database.

> **Important:**   The following parameters in the *sctally.ini* file do not need to be modified. These should only be changed at the direction of Peregrine support personnel.

log:sctally.log

Indicates that the name of the message log file is to be "sctally.log".

debugscautoevents:0

If 1, additional messages are logged in the indicated log file.

event_map_dir:EventMap

Specifies the directory (relative to installation directory) where event maps are kept.

Note: We build the a parameter for the database containing the SCINFO mapping table based on the assumption that these are contained in an *.MDB* file supplied by Peregrine, called SCTALLY.MDB. If this table is moved by the customer then a hard-coded parameter called *dbconnect2* must be supplied in the *sctally.ini* file.

sctally.cfg File

The *sctally.cfg* file is read by the SCAuto for Tally NetCensus service manager program (SCTALLYSRV.EXE) at startup, and controls the commands and processes executed by the service. Each line in the file must be a valid Windows NT command.

There are two types of commands that can be specified:

- Transient NT commands, such as might be issued from an NT command line. These must be prefixed with *CMD /c*. Once execution of the command finishes, it terminates. These can be used to perform maintenance tasks (such as moving or renaming a log file) each time the service is started.

- Long-running commands, which execute until the service is shut down. The long running commands relevant to SCAuto for Tally NetCensus are discussed below.

Transient NT Commands

As distributed, no transient NT commands are specified in the *sctally.cfg* file.

Long-running Commands

The following long-running commands are relevant to SCAuto for Tally NetCensus:

SCEVMON.EXE

scevmon.exe is a background process which is responsible for communication with ServiceCenter. It runs until the service is shut down. It makes a TCP connection with the SCAUTOD server component of ServiceCenter, using the hostname and port information supplied via the **sceventserver** parameter of the *sctally.ini* file. There should be no need to modify this line of sctally.cfg. If you do, SCAuto for Tally NetCensus will probably not operate correctly.

WSCRIPT.EXE

This command invokes the Microsoft scripting host to execute the *sctally.js* script. The JScript code in *sctally.js* implements the actual logic of the gateway.

Chapter 4   ServiceCenter Customization

There should be no need to customize ServiceCenter to obtain benefits from SCAuto's integration of Tally NetCensus information. Any necessary modifications to ServiceCenter tables are included with the installation in the *Unload* sub-folder under the installation folder. The Inventory model in ServiceCenter provides a repository for many types of hardware and software objects, and a rich set of attributes for each. Should you need to extend the repository to store additional device types or attributes, the ServiceCenter Application Administration, Data Administration and System Tailoring Guides provides the information you need.

Chapter 5   Customization

Overview

SCAuto for Tally NetCensus is delivered with default Mapping Instructions that correspond to Out-of-the-Box ServiceCenter Inventory definitions. A small Microsoft Access database is supplied as *sctally.mdb* which contains the all-important mapping tables. Note well that while it is NOT necessary to have Microsoft Access installed on the machine where this database is, or where SCAuto for Tally NetCensus is installed (the MDAC components allow the script to read this database regardless), it is required if you want to view or change this database. For this reason only, it is recommended that you install a reasonably recent copy of Access on some machine where you can interactively view and update this database.

The *sctally.mdb* database contains a small table named SCINFO. The SCINFO table contains a column called SCNAME which is the ICM field name corresponding to the NetCensus data element which is defined by the key of each row of the table. All customization is done simply by updating these two tables. This database is read into memory at startup of the service; if you change the database, you must restart the service. Peregrine recommends that no change be made to the mapping instructions until after an initial run of the SCAuto for Tally NetCensus against an unmodified ServiceCenter Inventory repository.

Customizing the SCINFONAME Table

The SCINFONAME table is contained in a small Microsoft Access database that is supplied with SCAuto for Tally NetCensus as an *.mdb* file. It contains the mapping instructions that relate NetCensus information to corresponding attributes in ServiceCenter's inventory repository. The sample below shows only the default mapping instructions.

TYPE	NAME	FIELD_ORDER	SCNAME
H	NODE		domain
H	SUB_CLASS	2	feature.description
H	TAG_NUM	2	feature.id
H	UNIT	2	feature.size
H	SIZE	1	feature.size
H	MODEL	5	feature.description
H	PROD_NAME	4	feature.description
H	COMMENT	3	feature.description
H	SERIAL_NUM	3	feature.id
H	BASE_ADDR	7	feature.description
H	LOCATION_1	1	location
H	LAST_NAME	1	contact.name
H	MID_NAME	3	contact.name
H	LAN_ADDR		ipx.address
H	LOCATION_2	2	location
H	TOP_ADDR	6	feature.description
H	MANU_NAME		feature.vendor
H	PHONE		primary.phone

S	COMMENT		directory
S	MANU_NA ME		vendor
S	PROD_NAM E		application.name
S	SERIAL_NU M		serial.number
S	TAG_NUM		asset.id
S	SUB_CLASS	2	description
S	SIZE		status
S	CLASS	1	description
S	MODEL		version
H	DEPT_NAM E		dept
H	COMMENT_ 1	3	comments
H	COMMENT_ 2	2	comments
H	COMMENT_ 3	1	comments
H	FIRST_NAM E	2	contact.name
H	CONFIG_ID		logical.name
H	PROD_ID	1	feature.id
H	TYPE		subtype
H	CLASS	1	feature.description
H	DLM		last.update
S	PROD_ID		file.name
S	CONFIG_ID		logical.name

Mapping Instructions

All NetCensus data is mapped to a field (or column) in one of ServiceCenter's inventory repository tables. The name of the table is defined in the event's registry inside ServiceCenter, and the names for fields in that table can be determined by browsing the Database Dictionary. Information referenced by each row in the SCINFO table's NAME field can be mapped to one or more inventory fields as decribed in the SCNAME column. The FIELD_ORDER column determines the order in which multiple will be written to the SC field.

There are two types of mapping for NetCensus components: *hardware* and *software*. This is because NetCensus uses the same field name in different ways depending upon whether the component is a hardware or a software component. Each NetCensus field you wish to map must be identified as either a hardware or software type mapping instruction by entering *H* or *S* respectively in the TYPE column. The hardware components are mapped to a single record, using the *feature.** fields. All hardware components are written to the *device* and *pc* files, while all software components are written to the *pcfiles* file.

The simplest possible mapping instruction is an individual *eventin* field name, for example, *dept*. If the named field is a scalar field, the NetCensus data item is appended to the data already in that field. Any data mapped to ServiceCenter's *logical.name* field is checked to assure that it is valid: it must not be an empty string or null. Invalid entries are discarded.

In addition to supporting scalar event fields, array event fields are supported in a similar fashion: identify the name of an array field, such as *feature.size*, and the NetCensus data item will be mapped to the next available slot at the end of the array.

Mapping Tips

There are two mappings for NetCensus items. One is defined inside ServiceCenter, and is extracted and written to the EventMap subfolder of the SCAuto for Tally NetCensus folder. The other is defined in the SCINFO table of the *sctally.mdb* database, which is stored in the Data subfolder.

SCAuto for Tally NetCensus is delivered with pre-defined maps. Without changing anything, starting SCAuto for Tally NetCensus will create events based upon these mapping instructions. However, the mappings are designed to be easily adapted to your business environment.

For most installations, simple mapping instructions will be sufficient. Additional fields already defined in ServiceCenter's Inventory repository can be associated to NetCensus items in the SCINFO table.

You may also need to extend the ServiceCenter database definition to accommodate more NetCensus data. To do this:

1. Use the Database Dictionary Tool to add new fields to the appropriate table definitions (the attribute table, such as *pc*, and the corresponding join table, such as *devicepc*). The table definitions for each NetCensus item are defined in the script, and is currently limited to *pc* and *pcfiles*.

2. Rename the current Event Registration for the device type (e.g., *ICMpc* for this example).

3. Delete or rename the corresponding Event Map definitions. The Event Map name is defined in the Event Registration.

4. From the Event Services menu, select Build New ICM Event Map. Identify the device type and a new mapping will be created.

Important: If you make any change to Event Mapping inside ServiceCenter, you must delete the corresponding **.input* file (for example, *ICMpc.input*) in the EventMap subfolder of the SCAuto for Tally NetCensus folder so that it can be recreated from the new Event Services Mapping. Remember that ServiceCenter must be active for SCAuto to obtain new mapping information.

Note: There is a 32K record size limit in ServiceCenter.

Debugging

Should you encounter problems in your Mapping Instructions, check the following:

- Is there a ServiceCenter device type for the pc? If not, create a new Device Type (see ServiceCenter 3 Application Administrator Guide, 4-29).

- Is there a ServiceCenter Event Registration for *ICMpc* and *DBApcfiles*? If not, add a new Registration record (see the ServiceCenter Event Services Guide.)

- Are there ServiceCenter Event Maps associated with the Registration? Create the Event Map records for the device type, and delete the *ICMpc.input* file from the EventMap subfolder—a new one will be created when the SCAuto for Tally NetCensus task is started *as long as* ServiceCenter is also active.

- Is there an *ICMpc.input* and *DBApcfiles* file in the EventMap subfolder? If not, make sure ServiceCenter is active before the SCAuto for Tally NetCensus task is started.

- Does the data in the *ICMpc.input* and *DBApcfiles* file correspond to the information in the ServiceCenter Event Maps? If not, delete the **.input* file and make sure ServiceCenter is active before restarting the SCAuto task.

You can also enable a debugging statement by setting the *debugscautoevents* parameter in the *sctally.ini* file to *1*.

```
debugscautoevents = 1
```

Then restart the script and check the *sctally.log* file. It should contain messages like these to help you debug your mapping instructions:

```
12/09/1999 18:07:05   pid (264)(197) =========== SCAuto for Tally
NetCensus: Script sctally.js started ===========
12/09/1999 18:07:05   pid (264)(197) Home directory is
D:\PROGRA~1\PEREGR~1\SCAUTO~4
12/09/1999 18:07:05   pid (264)(197) Using .ini file
D:\PROGRA~1\PEREGR~1\SCAUTO~4\sctally.ini
12/09/1999 18:07:05   pid (264)(197) Successfully created the SCEvent
object
12/09/1999 18:07:05   pid (264)(197) scauto_init: WScript.exe linkedited
on Mon Sep 29 15:05:24 1997)
12/09/1999 18:07:05   pid (264)(197) scauto_init: Using directory
D:\PROGRA~1\PEREGR~1\SCAUTO~4
12/09/1999 18:07:05   pid (264)(197) Using NetCensus database connection
string Driver={Microsoft DBase Driver (*.dbf)}; DBQ=C:\\SCAuto for
Tally NetCensus\\data; DefaultDir=C:\\SCAuto for Tally NetCensus\\
12/09/1999 18:07:05   pid (264)(197) Using SC Mapping Database
connection string Driver={Microsoft Access Driver (*.mdb)};
DBQ=SCTALLY.MDB; DefaultDir=D:\PROGRA~1\PEREGR~1\SCAUTO~4\Data
```

```
12/09/1999 18:07:05   pid (264)(197) Successfully created the ADO
database objects
12/09/1999 18:07:05   pid (264)(197) Successfully connected to the
NetCensus database
12/09/1999 18:07:05   pid (264)(197) Successfully connected to the
SCNetCensus mapping database
12/09/1999 18:07:06   pid (264)(197) Executing SQL against SCINFO:
Select * from SCINFO where TYPE = 'H' and SCNAME is not NULL and SCNAME
not Like ' ' order by SCNAME, TYPE, FIELD_ORDER
12/09/1999 18:07:06   pid (264)(197) Executing SQL against SCINFO:
Select * from SCINFO where TYPE = 'L' and SCNAME is not NULL and SCNAME
not Like ' ' order by SCNAME, TYPE, FIELD_ORDER
12/09/1999 18:07:06   pid (264)(197) Executing SQL against SCINFO:
Select * from SCINFO where TYPE = 'S' and SCNAME is not NULL and SCNAME
not Like ' ' order by SCNAME, TYPE, FIELD_ORDER
12/09/1999 18:07:06   pid (264)(197) Executing SQL against SCINFO:
Select * from SCINFO where TYPE = 'A' and SCNAME is not NULL and SCNAME
not Like ' ' order by SCNAME, TYPE, FIELD_ORDER
12/09/1999 18:07:08   pid (264)(197) Selecting NetCensus information
using SELECT DISTINCTROW [CONFIG].[CONFIG_ID],
[CONFIG].[FIRST_NAME],[CONFIG].[MID_NAME], [CONFIG].[LAST_NAME],
[CONFIG].[DEPT_NAME],[CONFIG].[LOCATION_1], [CONFIG].[LOCATION_2],
[CONFIG].[PHONE], [CONFIG].[LAN_ADDR], [CONFIG].[NODE],
[CONFIG].[COMMENT_1], [CONFIG].[COMMENT_2], [CONFIG].[COMMENT_3],
[CONFIG].[SCANDATE], [COMPON].[PROD_ID], [COMPON].[INSTANCE],
[COMPON].[STATUS], [COMPON].[COMMENT], [COMPON].[SERIAL_NUM],
[COMPON].[SIZE], [COMPON].[UNIT], [PRODUCT].[MANU_NAME],
[PRODUCT].[PROD_NAME], [PRODUCT].[MODEL],[PRODUCT].[TYPE],
[PRODUCT].[CLASS], [PRODUCT].[SUB_CLASS], [IOMAP].[BASE_ADDR],
[IOMAP].[TOP_ADDR], [ASSETTAG].[TAG_NUM] FROM (((CONFIG LEFT JOIN
COMPON ON CONFIG.CONFIG_ID = COMPON.CONFIG_ID) LEFT JOIN IOMAP ON
(COMPON.PROD_ID = IOMAP.PROD_ID) AND (COMPON.CONFIG_ID =
IOMAP.CONFIG_ID)) LEFT JOIN PRODUCT ON COMPON.PROD_ID =
PRODUCT.PROD_ID) LEFT JOIN ASSETTAG ON (COMPON.PROD_ID =
ASSETTAG.PROD_ID) AND (COMPON.CONFIG_ID = ASSETTAG.CONFIG_ID) ORDER BY
COMPON.CONFIG_ID, TYPE, COMPON.PROD_ID
12/09/1999 18:07:08   pid (264)(197) Setting SCEvent field type to "pc"
12/09/1999 18:07:08   pid (264)(197) CONFIG_ID,logical.name has been set
to 36279120 in the scValues array
12/09/1999 18:07:08   pid (264)(197) FIRST_NAME,ky.tally.contact.name
has been set to Norris in the scValues array
12/09/1999 18:07:08   pid (264)(197) LAST_NAME,ky.tally.contact.name
has been set to Norris Merritt in the scValues array
12/09/1999 18:07:08   pid (264)(197) LAN_ADDR,ipx.address has been set
to 0020AF3DBD16 in the scValues array
12/09/1999 18:07:08   pid (264)(197) COMMENT_3,comments has been set to
Computername: NMERRITT_NT40S, Domain: SAN_CLEMENTE in the scValues
array
12/09/1999 18:07:08   pid (264)(197) PROD_ID,feature.id has been set to
H0002362 in the scValues array
12/09/1999 18:07:08   pid (264)(197) MANU_NAME,feature.vendor has been
set to Unknown in the scValues array
12/09/1999 18:07:08   pid (264)(197) PROD_NAME,feature.description has
been set to Pentium System in the scValues array
12/09/1999 18:07:08   pid (264)(197) TYPE,subtype has been set to H in
the scValues array
```

```
12/09/1999 18:07:08   pid (264)(197) CLASS,feature.description has been
set to Pentium System System in the scValues array
12/09/1999 18:07:08   pid (264)(197) SUB_CLASS,feature.description has
been set to Pentium System System Pentium in the scValues array
12/09/1999 18:07:08   pid (264)(197) Setting SCEvent field logical.name
to "36279120"
12/09/1999 18:07:08   pid (264)(197) Setting logical.name to 36279120
12/09/1999 18:07:08   pid (264)(197) Setting SCEvent field primary.phone
to ""
12/09/1999 18:07:08   pid (264)(197) Setting primary.phone to
12/09/1999 18:07:08   pid (264)(197) Setting SCEvent field location to
""
12/09/1999 18:07:08   pid (264)(197) Setting location to
12/09/1999 18:07:08   pid (264)(197) Setting SCEvent field feature.id to
"H0002362│"
12/09/1999 18:07:08   pid (264)(197) Appending H0002362 to array
feature.id
12/09/1999 18:07:08   pid (264)(197) Setting SCEvent field
feature.description to "Pentium System System Pentium│"
12/09/1999 18:07:08   pid (264)(197) Appending Pentium System System
Pentium to array feature.description
12/09/1999 18:07:08   pid (264)(197) Setting SCEvent field
feature.vendor to "Unknown│"
12/09/1999 18:07:08   pid (264)(197) Appending Unknown to array
feature.vendor
12/09/1999 18:07:08   pid (264)(197) Setting SCEvent field comments to
"Computername: NMERRITT_NT40S, Domain: SAN_CLEMENTE│"
12/09/1999 18:07:08   pid (264)(197) Appending Computername:
NMERRITT_NT40S, Domain: SAN_CLEMENTE to array comments
12/09/1999 18:07:08   pid (264)(197) Setting SCEvent field ipx.address
to "0020AF3DBD16"
12/09/1999 18:07:08   pid (264)(197) Setting ipx.address to 0020AF3DBD16
12/09/1999 18:07:08   pid (264)(197) Setting SCEvent field domain to ""
12/09/1999 18:07:08   pid (264)(197) Setting domain to
12/09/1999 18:07:08   pid (264)(197) Setting SCEvent field feature.size
to "│"
12/09/1999 18:07:08   pid (264)(197) Appending  to array feature.size
12/09/1999 18:07:08   pid (264)(197) Setting SCEvent field last.update
to ""
12/09/1999 18:07:08   pid (264)(197) Setting last.update to
12/09/1999 18:07:08   pid (264)(197) Setting SCEvent field subtype to
"H"
12/09/1999 18:07:08   pid (264)(197) Setting subtype to H
12/09/1999 18:07:08   pid (264)(197) Setting SCEvent field type to "pc"
12/09/1999 18:07:08   pid (264)(197) CONFIG_ID,logical.name has been set
to 36279120 in the scValues array
12/09/1999 18:07:08   pid (264)(197) FIRST_NAME,ky.tally.contact.name
has been set to Norris Merritt Norris in the scValues array
12/09/1999 18:07:08   pid (264)(197) LAST_NAME,ky.tally.contact.name
has been set to Norris Merritt Norris Merritt in the scValues array
12/09/1999 18:07:08   pid (264)(197) LAN_ADDR,ipx.address has been set
to 0020AF3DBD16 in the scValues array
12/09/1999 18:07:08   pid (264)(197) COMMENT_3,comments has been set to
Computername: NMERRITT_NT40S, Domain: SAN_CLEMENTE in the scValues
array
```

```
12/09/1999 18:07:08  pid (264)(197) PROD_ID,feature.id has been set to
H0551002 in the scValues array
12/09/1999 18:07:08  pid (264)(197) COMMENT,feature.description has
been set to 1 CPU(s) in the scValues array
12/09/1999 18:07:08  pid (264)(197) SIZE,feature.size has been set to
166 in the scValues array
12/09/1999 18:07:08  pid (264)(197) UNIT,feature.size has been set to
166 MH in the scValues array
12/09/1999 18:07:08  pid (264)(197) MANU_NAME,feature.vendor has been
set to Intel in the scValues array
12/09/1999 18:07:08  pid (264)(197) PROD_NAME,feature.description has
been set to 1 CPU(s) Pentium in the scValues array
12/09/1999 18:07:08  pid (264)(197) TYPE,subtype has been set to H in
the scValues array
12/09/1999 18:07:08  pid (264)(197) CLASS,feature.description has been
set to 1 CPU(s) Pentium Processor in the scValues array
12/09/1999 18:07:08  pid (264)(197) SUB_CLASS,feature.description has
been set to 1 CPU(s) Pentium Processor Pentium in the scValues array
12/09/1999 18:07:08  pid (264)(197) Setting SCEvent field logical.name
to "36279120"
```

Warning! This setting generates many messages, and can cause your *log* file
to grow quickly (to the size established in the **logmaxlen**
parameter), so try to limit your test by setting the lastrundate
parameter to a value you think will select a very small number of
records. After you have corrected your mapping instructions, be
sure to reassign the debugging parameter to *0*:

```
debugscautoevents = 0
```

to disable debugging and improve performance.

Chapter 6   Scripting

Overview

This chapter explains the how Microsoft JScript 5.0 is used in the SCAuto for Tally NetCensus gateway, and how other ActiveX scripting languages such as VBScript 5.0 may be used in addition to, or instead of JScript. It also discusses some of the logic of the provided script called *sctally.js*, which is the main program of the gateway.

Windows Scripting Architecture and Terminology

The Microsoft architecture for ActiveX scripting languages in Windows is still fairly new, so it may help to briefly explain a few concepts and terminology.

Until very recently, the only native scripting language supported by the Windows operating system was the old MS-DOS® operating system command language used for batch files (*.bat*). This was a serious limitation of Windows. With the new Windows scripting architecture, users can take advantage of powerful scripting languages such as Visual Basic Scripting Edition and JScript. These scripting languages support object-oriented programming using OLE Automation objects (COM).

The Windows scripting architecture has three components:

ActiveX scripting hosts

These are programs such as the Microsoft-provided WSCRIPT.EXE and CSCRIPT.EXE. A scripting host is a program which can load an ActiveX scripting engine and a text file containing a script, and *feed* the script to the engine. A scripting host may also implement various objects and make these available to the script programmer.

ActiveX scripting engines

A scripting engine is a .DLL file which implements a particular scripting language. Microsoft currently provides two different scripting engines, one for JScript and another for VBScript. These are included in the Windows Script 5.0 distribution described in Chapter 2.

Scripts written in one of the supported scripting languages

Using Windows scripting is as simple as writing a script in a supported language and invoking a scripting host such as *wscript.exe* to process the script. The script can use any OLE Automation objects which have been installed on the Windows platform where the script is running. You may be surprised at the number of such objects which can be found on any given Windows system.

Microsoft Scripting Engine Considerations

The SCAuto for Tally NetCensus scripts use Microsoft JScript 5.0, which is an ECMA262 compliant implementation of ECMA script.

Because it makes use of Microsoft extensions not yet in the ECMA specification, such as exception handling using *try...catch* notation, it is important to be sure that Microsoft JScript version 5.0 or later is installed on the NT platform where SCAuto NetCensus is running. Versions of JScript prior to 5.0 do not contain support for exception handling.

Although we chose to use JScript to implement SCAuto for Tally NetCensus, the OLE Automation objects provided by Peregrine can be used just as easily from VBscript than from JScript. So if you have a major project to tackle, such as developing an additional script to integrate some other Tally NetCensus application with ServiceCenter, you could do it in VBScript if you prefer.

Windows Scripting Host Considerations

No particular use is made of the special *wsh* objects or other Windows Scripting Host specific functionality in this release. *wscript.exe* is simply used to invoke and run the sctally.js script. The command shell *cscript.exe* could just as easily be used instead.

The first release of SCAuto for Tally NetCensus happens to use Windows Scripting Host 1.0, because WSH 2.0 was not yet generally available.

If you plan to do significant customization, you should investigate Windows Scripting Host 2.0, which supports multiple scripting languages in the same file, inclusion of scripts from other files, and other useful things.

Recommended Tools

There are a variety to tools available for free download from the Microsoft scripting web site referenced earlier in this book. These include a simple script debugger. However, it is recommended that you instead install the much more capable script debugger included in the Visual Studio Professional or Enterprise edition product.

No script customization should be attempted without first installing some sort of script debugger. If you do not have a version of Visual Studio which includes the enhanced script debugger, download the free script debugger from the Microsoft scripting web site.

The SCTALLY.JS Script

This script functions as a main program of the SCAuto NetCensus gateway. It collects information about about hardware devices, maps it to ServiceCenter fields, and writes it to an external file. It is executed by the command **wscript.exe sctally.js** which appears in the *sctally.cfg* file. The latter file is processed by the SCAuto NetCensus service control program, *sctallysrv.exe*, at startup, and commands in the file are executed. You can make the gateway execute additional scripts by adding additional lines to the *sctally.cfg* file to invoke wscript.exe for the new script.

The *sctally.js* script is essentially an endless loop which runs until the SCAutoNetCensus service is shut down, which causes *sctallysrv.exe* to terminate the wscript process(es) that it started at startup. After some initialization code is executed to read parameters from sctally.ini and create some OLE Automation objects, the script executes continuously until service shutdown.

If the SCAutomate services are running, the events written to *scevents.to.<host.port>* will be sent to the ServiceCenter *eventin* queue, where they will be processed by Event Services and inserted into ServiceCenter's Inventory repository.

The flow diagram on the next page provides an overview of the *sctally.js* script.
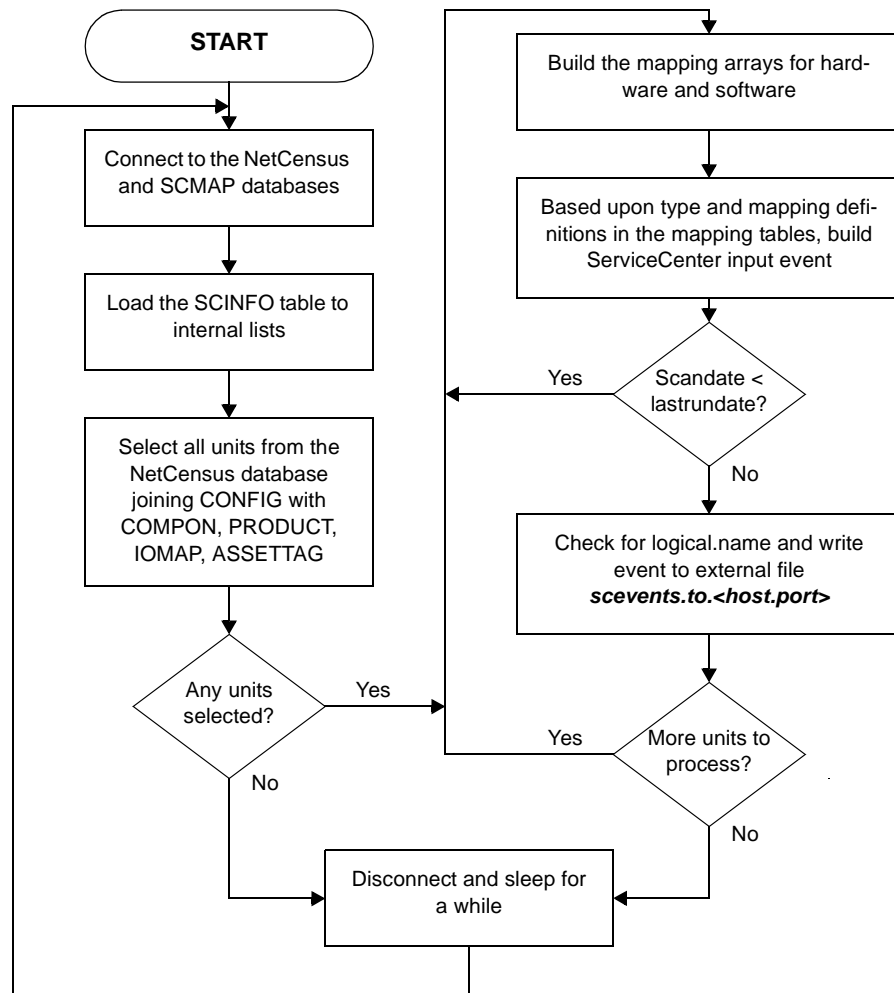
Figure 6-1. sctally.js script flow diagram

Chapter 7   Operation

Overview

This chapter describes the procedures for operating SCAuto for Tally NetCensus.

Starting SCAuto for Tally NetCensus

There are three ways to start SCAuto for Tally NetCensus:

- Select **Start->Programs->Peregrine Systems->Start SCAuto for Tally NetCensus** from the menu.

  This executes the command **sctallysrv -start** from the *\SCAuto for Tally NetCensus\Bin* directory.

- You can also issue the following command from a Command window (the current directory in the command window is unimportant):

  net start scautotally

- You can also execute the Control panel services applet to start SCAuto for Tally NetCensus.

All three of these methods cause the same code, and thus are completely identical in function.

Starting the SCAUTOTALLY service normally takes from two to five seconds. A normal startup of the service generates messages in the *sctally.log* file like this:

```
12/10/1999 12:11:04   pid (220)(191) =================== SCAuto for
Tally NetCensus started ===================
12/10/1999 12:11:04   pid (220)(191) sctallysrv.exe (Linked: Fri Dec 03
21:46:40 1999) started
12/10/1999 12:11:04   pid (220)(191) sctallysrv.exe: Created stop event
SCTNC.StopEvent
12/10/1999 12:11:04   pid (220)(191) Changing desktop for process
wscript.exe from Default to winsta0\default
12/10/1999 12:11:05   pid (155)(201) scevmon.exe (Linked: Fri Nov 19
14:28:56 1999) started
```

```
12/10/1999 12:11:05   pid (155)(201) scevmon: Using stop event
SCTNC.StopEvent
12/10/1999 12:11:05   pid (155)(201) scauto_init: scevmon.exe linkedited
on Fri Nov 19 14:28:56 1999)
12/10/1999 12:11:05   pid (155)(201) scauto_init: Using directory
D:\PROGRA~1\PEREGR~1\SCAUTO~4
12/10/1999 12:11:05   pid (155)(201) scevmon: "noeventsfromsc" parm
recognized. No events will be retrieved from ServiceCenter EventOut
queue
12/10/1999 12:11:05   pid (201)(74) Using NetCensus database connection
string Driver={Microsoft DBase Driver (*.dbf)}; DBQ=C:\\SCAuto for
Tally NetCensus\\data; DefaultDir=C:\\SCAuto for Tally NetCensus\\
12/10/1999 12:11:05   pid (201)(74) Using SC Mapping Database connection
string Driver={Microsoft Access Driver (*.mdb)}; DBQ=SCTALLY.MDB;
DefaultDir=D:\PROGRA~1\PEREGR~1\SCAUTO~4\Data
12/10/1999 12:11:05   pid (201)(74) Successfully created the ADO
database objects
12/10/1999 12:11:08   pid (201)(74) Successfully connected to the
NetCensus database
12/10/1999 12:11:09   pid (201)(74) Successfully connected to the
SCNetCensus mapping database
```

If the service is unable to start for any reason, a message will be logged in the Windows NT system log, and detailed error messages will appear in the *sctally.log* file.

What happens when SCAuto for Tally NetCensus is Started

When SCAuto for Tally NetCensus is started, the main service executable *sctallysrv.exe* is invoked. It then locates the file *sctally.cfg* which is found in the *\SCAuto for Tally NetCensus* directory. If this file cannot be found, the service will not start.

Next, the *sctally.cfg* file is processed. Each line in the file which is not blank or commented is interpreted as a process to run. Processes are typically either transient commands, i.e., invocations of *cmd.exe*, or one of the following background processes installed with SCAuto for Tally NetCensus:

scevmon.exe

> This process makes a TCP connection with the SCAUTOD component of ServiceCenter, using the hostname and port number information supplied to the configuration dialog program, *sctallycfg.exe*. Whenever a connection is established, scevmon attempts to exchange events with ServiceCenter. *scevmon.exe* manages the local queue of outbound events contained in the SCAuto for Tally NetCensus installation directory.

The name of the outbound queue is *scevents.to.<host>.<port number>.* Outgoing events are logged to the *scevents.to.* file by processes such as the transient message action command *scevent.exe* or the long-running background process and consumed by *scevmon.exe*, which sends them to the ServiceCenter SCAUTOD process indicated in the file name of the *scevents.to* queue. When SCAUTOD accepts the events, they are written to the ServiceCenter EVENTIN file.

After start-up, SCAuto for Tally NetCensus generates one ICM event for each NetCensus *configuration* (a term similar to the SMS term *machine*), which includes all of the hardware components contained in the unit, and one DBA event for each software component. For hardware, some of the NetCensus configurations are personal computers, some are Microsoft NT servers and some are Novell servers.

SCAuto for Tally NetCensus decides what information to put in each sub field of the events by consulting the SCINFO table. Upon startup, without doing any customization, you should see ICM and DBA events being generated. Look in the sctally.log file for a number of messages such as these:

```
12/07/1999 21:55:47   pid (253)(329) Successfully logged event
36279120^bebe Brown^^^^^
^^^^^pc^^^^^^^^^^H0002362|H0551002|H1001001|H1101L01|H1300001 24.1.152.
2|H1400001|H1400002|H1501001|H2041143|H3000001|H3000004|H3000005|H30000
05|H3000005|H3000005|H3001001|H3001001|H3001055|H3002006|H3011001|H3501
001|H3504001|H3504004|H3511001|H3599031|H4021293|H4031L01|H6011002|H990
4050|^Pentium System System Pentium|1 CPU(s) Pentium ProcessorPentium|B
ASE MEMORY Base Memory Memory Base|AWARD MODULAR BIOS V4.50PG ROM BIOS
01/26/96 BIOS BIOS|Winsock Vers:2.2, MaxSockets:32767 IP Address
Network Address IP Address|Total Space Properties Fixed Disk|Free Space
Properties Fixed Disk|Monitor Color Monitor Monitor Color|Video Adapter
ViRGE PCI Video Adapter VGA|Hard Drive 0 Boot Manager Partition Mass
Storage Disk Partition|Hard Drive 2 IFS Partition Mass Storage Disk
Partition|Hard Drive 0 FAT-16 Partition - Big DOS Mass Storage Disk
Partition|Hard Drive 1 FAT-16 Partition - Big DOS Mass Storage Disk
Partition|Hard Drive 1 FAT-16 Partition - Big DOS Mass Storage Disk
Partition|Hard Drive 0 FAT-16 Partition - Big DOS Mass Storage Disk
Partition|Logical Drives: DG Hard Drive Mass Storage Hard Drive|Logical
Drives: E Hard Drive Mass Storage Hard Drive|Logical Drives: CF Type 46
Hard Drive Mass Storage Hard Drive|CD-ROM #1, Logical Drive G: CD-ROM
Drive Mass Storage CD-ROM|FLOPPY DRIVE #1 Diskette Drive Mass Storage
Diskette|SERIAL PORT Serial Ports I/O Serial|Primary Bus ISA Bus I/O
Bus|Secondary Bus PCI Bus I/O Bus|PARALLEL PORT Parallel Ports I/O
Parallel|Mouse I/O Other|EtherLink III Comm Hardware LAN|COM2
VoiceSURFR 56K External Comm Hardware Modem/Async|Keyboard 101/102
keyboard Keyboard 101/102 key|SCSI Controller C825 PCI SCSI Controller
Other Hardware SCSI|^Unknown|Intel||Award Software|||||S3,
Inc.||||||||||||||||Microsoft
compatible|3Com|Motorola||NCR|^Computername: BBROWN_NT40S, Domain:
SAN_CLEMENTE|Computername: BBROWN_NT40S, Domain:
SAN_CLEMENTE|Computername: BBROWN_NT40S, Domain:
SAN_CLEMENTE|Computername: BBROWN_NT40S, Domain:
```

```
SAN_CLEMENTE|Computername: BBROWN_NT40S, Domain:
SAN_CLEMENTE|Computername: BBROWN_NT40S, Domain:
SAN_CLEMENTE|Computername: BBROWN_NT40S, Domain:
SAN_CLEMENTE|Computername: BBROWN_NT40S, Domain:
SAN_CLEMENTE|Computername: BBROWN_NT40S, Domain:
SAN_CLEMENTE|Computername: BBROWN_NT40S, Domain:
SAN_CLEMENTE|Computername: BBROWN_NT40S, Domain:
SAN_CLEMENTE|Computername: BBROWN_NT40S, Domain:
SAN_CLEMENTE|Computername: BBROWN_NT40S, Domain:
SAN_CLEMENTE|Computername: BBROWN_NT40S, Domain:
SAN_CLEMENTE|Computername: BBROWN_NT40S, Domain:
SAN_CLEMENTE|Computername: BBROWN_NT40S, Domain:
SAN_CLEMENTE|Computername: BBROWN_NT40S, Domain:
SAN_CLEMENTE|Computername: BBROWN_NT40S, Domain:
SAN_CLEMENTE|Computername: BBROWN_NT40S, Domain:
SAN_CLEMENTE|Computername: BBROWN_NT40S, Domain:
SAN_CLEMENTE|Computername: BBROWN_NT40S, Domain:
SAN_CLEMENTE|Computername: BBROWN_NT40S, Domain:
SAN_CLEMENTE|Computername: BBROWN_NT40S, Domain:
SAN_CLEMENTE|Computername: BBROWN_NT40S, Domain:
SAN_CLEMENTE|Computername: BBROWN_NT40S, Domain:
SAN_CLEMENTE|Computername: BBROWN_NT40S, Domain:
SAN_CLEMENTE|Computername: BBROWN_NT40S, Domain:
SAN_CLEMENTE|Computername: BBROWN_NT40S, Domain:
SAN_CLEMENTE|Computername: BBROWN_NT40S, Domain:
SAN_CLEMENTE|^0020AF3DBD16^^^^^^^^^^^^|166 MH|64 MB|||6.5 GB|1.47
GB|||2.06 MB|4.33 GB|534.64 MB|534.61 MB|551.06 MB|551.16 MB|1.09
GB|4.33 GB|1.09 GB|NA|1.44 MB|2 EA|||1
EA||||||^^^^^^^^^^^^^^^^^^^^^^^^^H^
12/07/1999 21:55:47   pid (253)(329) Successfully logged event
^36279120^Build 1381 Service Pack 3^^S0004L01^^^^Op System Windows
NT^Full^4.00^^^^50382438633948201198^^Microsoft^Windows NT Server
12/07/1999 21:55:48   pid (253)(329) Successfully logged event
^36279120^C:\WIN95^^S0202020^^^^Games
Misc^Full^1.0^^^^^^Microsoft^Hearts Network for Win95
12/07/1999 21:55:48   pid (253)(329) Successfully logged event
^36279120^C:\WIN95^^S0202022^^^^Games
Misc^Full^4.0^^^^^^Microsoft^FreeCell for Win32
12/07/1999 21:55:48   pid (253)(329) Successfully logged event
^36279120^N:\NETCWIN^^S0301267^^^^Asset Mgmt PC
Inventory^Full^2.93^^^^^^Tally Systems^NetCensus
12/07/1999 21:55:48   pid (253)(329) Successfully logged event
^36279120^N:\NETCWIN^^S0301268^^^^Asset Mgmt PC
Inventory^Full^2.93^^^^^^Tally Systems^NetCensus Config Detail Report
12/07/1999 21:55:48   pid (253)(329) Successfully logged event
^36279120^N:\NETCWIN^^S0301269^^^^Asset Mgmt PC
Inventory^Full^2.93^^^^^^Tally Systems^NetCensus DOS Collector
12/07/1999 21:55:48   pid (253)(329) Successfully logged event
^36279120^N:\NETCWIN^^S0301270^^^^Asset Mgmt PC
Inventory^Full^2.93^^^^^^Tally Systems^NetCensus OS/2 Collector
12/07/1999 21:55:48   pid (253)(329) Successfully logged event
^36279120^N:\NETCWIN^^S0301271^^^^Asset Mgmt PC
Inventory^Full^2.93^^^^^^Tally Systems^NetCensus Server Census
12/07/1999 21:55:48   pid (253)(329) Successfully logged event
^36279120^N:\NETCWIN^^S0301272^^^^Asset Mgmt PC
Inventory^Full^2.93^^^^^^Tally Systems^NetCensus Win32 Collector
```

```
12/07/1999 21:55:48   pid (253)(329) Successfully logged event
^36279120^N:\NetCensus^^S0301272^^^^Asset Mgmt PC
Inventory^Full^2.93^^^^^^Tally Systems^NetCensus Win32 Collector
12/07/1999 21:55:48   pid (253)(329) Successfully logged event
^36279120^N:\MSOffice\Winword^^S0501224^^^^Word Processor Word
Processors^Full^7.0^^^^^^Microsoft^Word for Win95
12/07/1999 21:55:49   pid (253)(329) Successfully logged event
^36279120^N:\ACC2SE\PROGS^^S0502011^^^^Word Processor
Translators^Full^1.3^^^^^^Microlytics^Berlitz Interpreter for Windows
12/07/1999 21:55:49   pid (253)(329) Successfully logged event
^36279120^N:\SMS\SITE.SRV\X86.BIN^^S0706159^^^^Network Network
Mgmt^Full^1.2^^^^^^Microsoft^Systems Management Server
12/07/1999 21:55:49   pid (253)(329) Successfully logged event
^36279120^N:\SMSADMIN\SITE.SRV\X86.BIN^^S0706159^^^^Network Network
Mgmt^Full^1.2^^^^^^Microsoft^Systems Management Server
12/07/1999 21:55:49   pid (253)(329) Successfully logged event
^36279120^N:\WINNT40S\system32^^S0803010^^^^Multimedia
Video^Full^2.1^^^^^^Apple^QuickTime for Win32
12/07/1999 21:55:49   pid (253)(329) Successfully logged event
^36279120^N:\WINNT40\system32^^S0803010^^^^Multimedia
Video^Full^2.1^^^^^^Apple^QuickTime for Win32
```

The rate at which the events are generated depends on several factors:

- The number of milliseconds of wait time introduced into the script. If you have a fast server, or if the database is on another machine, you should set the wait interval to zero. Locate the line of code toward the top of the *\Bin\sctally.js* file that says:

```
perItemSleepInterval = 1;
```

and change it to set the value to 0, instead of 1.

- The CPU speed and number of processors.

- The network speed and the speed of the database machine, if the database is on another machine.

Under ideal conditions, SCAuto for Tally NetCensus should be able to process about 4 components per second. The initial loading may take several hours in a large shop with 10,000 or more units in the database.

After processing all the units, the product will wait a user-defined interval before querying NetCensus to see if any of the units have been updated. This is done by selecting units with an LRUNDATE date value larger than that previously seen for any of the units previously processed.

Note: This wait interval parameter is defined in the *sctally.ini* file, and is specified as **sleep:**3600 (which is one hour) by default.

After the initial run, updates may arrive at the NetCensus database from time to time, rather than all at once.This is dependent upon the way you configure NetCensus. The performance of SCAuto for Tally NetCensus will be far better if the inventory collections done by NetCensus are periodic (i.e. each individual PC reboot or login) rather than all configurations in the entire organization at the same scheduled weekly time.

Stopping SCAuto for Tally NetCensus

There are three ways to stop SCAuto for Tally NetCensus:

- Select **Start/Programs/Peregrine Systems/Stop SCAuto for Tally NetCensus** from the menu.

    This executes the command **sctallysrv -stop** from the *\SCAuto for Tally NetCensus\Bin* directory.

- You can also issue the following command from a Command window. The current directory in the command window is unimportant:

    net stop scautotally

- You can also execute the **Control Panel Services** applet to stop SCAuto for Tally NetCensus.

All three of these methods execute the same code, and thus are completely identical in function.

Stopping the SCAUTOTALLY service normally takes five to ten seconds.

What happens when SCAuto for Tally NetCensus is Stopped

Stopping the SCAuto for Tally NetCensus service signals a Windows NT event called *SCTNC.StopEvent*. This causes the various processes that were started by the service to immediately go into termination logic, where they release whatever resources they acquired from Windows NT and then exit. You can verify that the service stopped normally by checking the *sctally.log* file, where yoy should see a series of messages like these:

```
12/10/1999 12:11:23  pid (155)(201) scevmon: Stop event recognized.
Terminating.
12/10/1999 12:11:23  pid (155)(201) scevmon.exe: scauto_term: entered
12/10/1999 12:11:23  pid (220)(191) Stop event recognized; waiting for
child processes to terminate
12/10/1999 12:11:26  pid (220)(191) One or more processes did not
terminate.
12/10/1999 12:11:26  pid (220)(191) Terminated process (215)
wscript.exe
12/10/1999 12:11:26  pid (220)(191) All processes terminated
12/10/1999 12:11:26  pid (220)(191) =================== SCAuto for
Tally NetCensus shutdown complete ===================
```

Should any process fail to exit voluntarily for any reason, the main service executable *sctallysrv.exe* will terminate it using the Windows NT *Terminate Process* command. For this reason, the service should not normally fail to stop. In this unlikely event, detailed error messages will appear in the *sctally.log* file.

Determining if Tally NetCensus is Running Normally

You can use the Windows NT Resource Kit command **SC** (Service Control) to query the SCAUTOtally service as follows:

sc query scautotally

This results in output similar to the following:

```
N:\tmp>sc query scautotally
SERVICE_NAME: scautotally
        TYPE               : 10  WIN32_OWN_PROCESS
        STATE              : 4   RUNNING
```

An alternative way to determine if SCAuto for Tally NetCensus is running is to use the Task Manager. If the service is running, the *sctallysrv.exe* process is visible in the task list displayed by Task Manager.

Day to Day Administration

SCAuto for Tally NetCensus is designed for minimal administration overhead and maximum availability and operational flexibility.

SCAuto for Tally NetCensus can tolerate indefinite outages of NetCensus and ServiceCenter. The operation of the latter products does NOT have to be synchronized with SCAuto for Tally NetCensus in any way; i.e., SCAuto for Tally NetCensus does NOT have to be stopped if ServiceCenter is being stopped or paused, nor is it necessary for those products to be running in order to start SCAuto for Tally NetCensus.

SCAuto for Tally NetCensus does not maintain a database. It produces a log file, but it automatically wraps this log when it reaches a defined maximum size. The default maximum log file size is 5 MB. This value is controlled by a setting in the *sctally.ini* file.

You should never need, therefore, to stop SCAuto for Tally NetCensus.

SCAuto for Tally NetCensus can be started and stopped at any time. However, long periods of NetCensus operation without running SCAuto for Tally NetCensus should be avoided, as this will result in out-of-date information in the ServiceCenter database.

Similarly, long periods of NetCensus and SCAuto for Tally NetCensus operation without running ServiceCenter should be avoided, as large numbers of events may accumulate in the *scevents.to...* file awaiting transmission to ServiceCenter.

Index