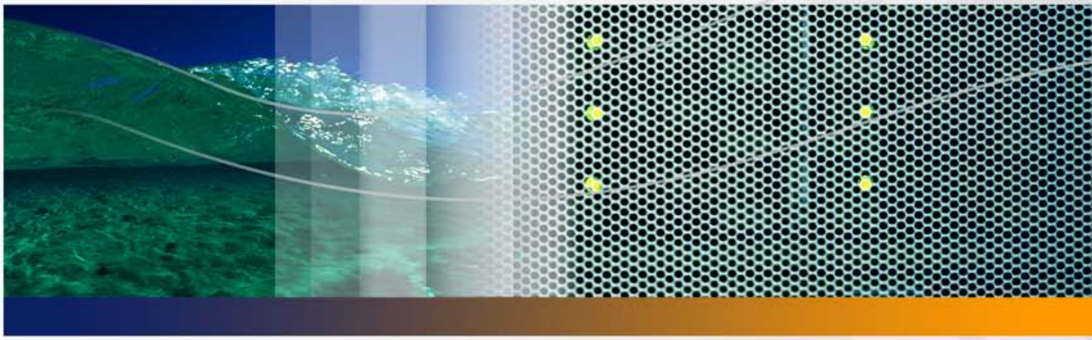
Product Documentation

Peregrine Systems, Inc.

Enterprise Discovery™ 2.0



Reference

Part No. PED-2-EN06

Peregrine SYSTEMS®

The information in this document is subject to change without notice and does not represent a commitment on the part of Peregrine Systems, Inc. Contact Peregrine Systems, Inc., Customer Support to verify the date of the latest version of this document. The names of companies and individuals used in the sample database and in examples in the manuals are fictitious and are intended to illustrate the use of the software. Any resemblance to actual companies or individuals, whether past or present, is purely coincidental. If you need technical support for this product, or would like to request documentation for a product for which you are licensed, contact Peregrine Systems, Inc. Customer Support by email at support@peregrine.com. If you have comments or suggestions about this documentation, please contact Peregrine Systems, Inc. Technical Publications by email at doc_comments@peregrine.com. This edition applies to version 2.0 of the licensed program.

For more copyright information, see the Copyright chapter of the Enterprise Discovery Reference Guide.

Peregrine Systems, Inc.
3611 Valley Centre Drive San Diego, CA 92130
858.481.5000
Fax 858.481.1751
www.peregrine.com

Contents

PEREGRINE

1 CHAPTER Introduction

Welcome to the Enterprise Discovery™ *Reference Guide*.

This guide contains several chapters that you may find useful as you use Enterprise Discovery. Included are many definitions of terms and concepts used in Enterprise Discovery, as well as other reference materials that might help you work with the product.

2 CHAPTER Terms and Concepts

This chapter is divided into two basic categories:

- Network Terms and Concepts on page 13 (to review terms and concepts common to network management)
- Enterprise Discovery Terms and Concepts on page 20 (to learn terms and concepts unique to this product)

Network Terms and Concepts

These terms and concepts are common to networks and network management. They are not unique to Enterprise Discovery.

SNMP

Defined by the Internet Engineering Task Force (IETF) in RFC 1157, Simple Network Management Protocol (SNMP) is the protocol that governs network management, and network device monitoring.

MIB

Management Information Base. This database of network management information is used by SNMP. The information contained in this database helps define each device by giving specific information about the device and manufacturer.

Domain names

Example: website.example.com

A domain name such as “website.example.com” is easier to remember than an IP address such as “192.168.96.1”. This ease of remembering is the chief reason for the existence of domain names.

The term “domain name” and “host name” are sometimes used interchangeably. A domain name is a name in the Domain Name System (DNS) format as registered with a DNS server. A host name is purely an internal name, used by a device to refer to itself.

Address types

The two main types of numeric address are the IP address and the MAC address.

IP address

An IP address was intended to be a unique number identifying a unique device or port of a device.

When you see the term “IP address” with no qualifiers in Enterprise Discovery, it means that either an IPv4 address or an IPv6 address is acceptable. The 32-bit address space of IPv4 addresses puts severe limits on the number of unique addresses available, and the supply is fast running out. The IPv6 128-bit address space was created to address this problem.

IPv4 address

An IPv4 address contains four sections separated by periods (or “dots”). Each section, called an octet, contains 8 bits expressed in decimal (0–255).

Example: 192.168.96.1

IPv6 address

An IPv6 address contains eight sections separated by colons. Each section contains 16 bits expressed in hexadecimal (0000–FFFF).

Example: 1234:5678:9ABC:DEF0:1234:5678:9ABC:DEF0

To make it easier to remember and type an IPv6 address, you can use a double colon (::) to indicate multiple contiguous sections of zeros. You can also omit leading zeroes. For example, you can simplify address 0123:0000:0000:0000:0004:0056:789A:BCDE to 123::4:56:789A:BCDE.

MAC address

A MAC (Media Access Control) address is a unique number identifying a unique device or port of a device.

When you see the term “MAC address”, it means a numeric MAC address.

Numeric MAC address

A MAC address contains six sections. Each section contains 8 bits expressed as a hexadecimal number (00–FF).

Sometimes the first three sections and last three sections are separated by one space; sometimes all sections are presented as one, without spaces; sometimes each section is separated by a colon or a space.

Examples: 010203 FDFEFF, 010203FDFEFF, 01:02:03:FD:FE:FF

MAC address including OUI

This type of MAC address is sometimes (inaccurately) referred to simply as an OUI. In fact, the Organization Unique Identifier (OUI) comprises the first three sections of a MAC address. If Enterprise Discovery recognizes the numeric form of the OUI, it replaces the numbers with a short form of the organization name. This makes it easier to identify a device. If Enterprise Discovery uses an alphabetic short form for a device’s OUI, the device is said to have a recognized OUI. Having a recognized OUI is sometimes abbreviated to “having” an OUI.

Example: DELL 59FC91

Netmask notation

Network masks, often referred to as netmasks, can usually be expressed in two formats in IPv4—either the familiar octet notation (also called dotted decimal notation) or CIDR notation.

Example of octet notation: 255.255.255.248

Example of CIDR notation: 29

The shorter CIDR notation is based on the binary equivalent of the octet notation, and refers to the numbers of contiguous 1's. Below are examples of netmask notation:

255.255.255.255	11111111.11111111.11111111.11111111	32 1s
255.255.255.248	11111111.11111111.11111111.11111000	29 1s
255.255.0.0	11111111.11111111.00000000.00000000	16 1s

In IPv6, netmasks can only be written in CIDR notation.

Community strings

A community string is a kind of device-based password that controls access to the SNMP MIB of a device. A device controls its own community strings, but you must tell Enterprise Discovery about them.

If Enterprise Discovery is not given the correct community strings and access to devices on your network, Enterprise Discovery will be unable to read device MIBs. Enterprise Discovery will then assume that each device it cannot read has no SNMP management available.

Multiple Strings

For each device that it discovers, Enterprise Discovery will try all the community strings you have provided for that device and use the first string that receives a positive acknowledgement to read or write to the system MIB. This means that Enterprise Discovery may try several community strings before it finds one that will cause the device to respond.

The fact that Enterprise Discovery may try several community strings has implications for any devices that issue SNMP traps (also known as security traps and authentication traps).

SNMP Traps

Some devices may issue an SNMP trap when Enterprise Discovery attempts to explore them. Even if Enterprise Discovery has the correct community string in its list, Enterprise Discovery may still "trip" the trap if Enterprise Discovery tries multiple community strings before finding the right one.

For example, Enterprise Discovery might try two invalid community strings before reaching the valid community string. Any invalid community string will “trip” a security trap.

Once a trap has been tripped, the trap may be re-issued periodically until the trap is reset. Enterprise Discovery does not reset traps. Therefore, you should either disable all such traps or use only a single correct community string for each device that issues a trap.

Note: If another network management system is used in the same network with Enterprise Discovery, this other system may generate alarms due to these traps.

Directed Community Strings

If a device is programmed with a directed community string (sometimes known as a direct access list), it will reject the attempt by Enterprise Discovery to SNMP QUERY it, even if Enterprise Discovery has been given the correct community string. With a directed community string, each device checks not only the “password,” but also to see if the Enterprise Discovery server is on the list of “trusted” devices.

You can allow Enterprise Discovery to communicate with a device with a directed community string, but you cannot do so merely by configuring Enterprise Discovery. You must also give the device itself an entry for a directed community string associated with the IP address of the Enterprise Discovery server.

Bridge aging

To obtain the best results with Enterprise Discovery, turn bridge aging on. Also, set the aging interval for 2–6 hours, although some circumstances may call for an aging interval as long as 12 or even 24 hours. (Longer aging intervals are not always possible. A common maximum aging interval is 32767 seconds, or just over 9 hours.)

Bridges, routers, and switches generally have tables in which they store the addresses of devices on the network. The tables are periodically purged and relearned in order to keep the list of devices current. The aging interval defines the frequency with which tables are purged and relearned.

When there is no table entry for the address of an incoming packet, the bridge, router, or switch must learn the location of the address. To learn the location, the device sends the incoming packet to all its own ports. (This is often referred to as “flooding” or “leakage”.) When the destination device with the corresponding address responds, the bridge, router, or switch learns the location and makes an entry in the address table.

If the table is full and a new entry must be made, the “oldest” entry is usually replaced by the new entry. Device manufacturers commonly strive to include a table large enough to hold the addresses of all active sessions, but space in a table is always finite.

Enterprise Discovery reads the tables of bridges, routers, and switches to learn the addresses of all the connected devices. Many bridge, router, and switch vendors use a standard aging interval of 300 seconds (5 minutes), which is too short.

If the bridge aging interval is too short:

- Enterprise Discovery may never discover devices that are connected to the network for short periods—for example, laptops.
- Tables will be purged so frequently that flooding will occur regularly, using bandwidth unnecessarily.

If bridge aging is not turned on for a device, or if the bridge aging interval is too long:

- Tables will contain old addresses of devices that may been removed from the network or devices that are broken. As a result, Enterprise Discovery will work from an outdated and possibly confused representation of what is in your network.

OSI model layers

The Open Systems Interconnection (OSI) model has seven layers. Layers 2 and 3 are the most important to Enterprise Discovery:

- Layer 2 is the Data Link layer, at which level MAC addresses are used. Bridges and some switches are layer 2 devices.
- Layer 3 is the Network layer, at which level IP addresses are used. Routers are layer 3 devices.

Some switches are both layer 2 and layer 3.

The seven layers are:

Layer number	Layer
7	Application
6	Presentation
5	Session
4	Transport
3	Network
2	Data Link
1	Physical

Management workstation

Any workstation or personal computer capable of running a supported web browser. There is more detail on requirements for a management workstation in the *Installation and Initial Setup Guide*.

Enterprise Discovery Terms and Concepts

These terms and concepts are either unique to Enterprise Discovery, or have a special meaning in this context.

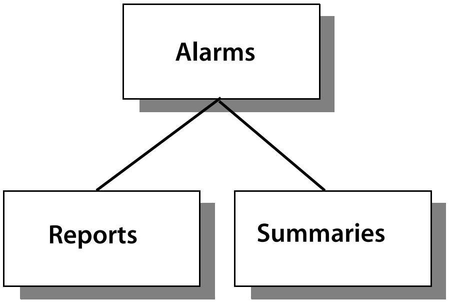
Events and Alarms

To see what alarms are raised by report data or attribute data, see **Help** > **Classifications** > **Alarms**.

There are 5 event types:

Event Type	Alarms Generated	Example
Add	info	new device added to network
Delete	info	the device is hidden or has been deactivated
Property Change	info	changing a device icon, priority

This diagram shows the Enterprise Discovery Alarm hierarchy.



Here is a list of the alarm indicators visible in the Health Panel and elsewhere in the user interface:

Alarm Type	Indicator	
n/a (not an alarm state, this indicates that the attribute is not being monitored)		blank
OK	—	dash
Info	✱	green asterisk
Minor Alarm	▲	gold triangle
Major Alarm	◆	orange diamond
Critical Alarm	🟥	red square

Panel Elements

Certain elements are common to all Device Manager or Port Manager panels:

- When data in a table has a gray background, the data shown is considered stale, because it was obtained before the beginning of your selected time period. (In some cases, data may be shown in parentheses rather than with a gray background.) To change the time before data is considered stale, see the section on Account Properties in the *Configuration and Customization Guide*.
- A blank space indicates that data is not available for a device or port.
- The final line on each panel is the date and time that the panel was refreshed. (This refers to rendering the panel itself, not when the data shown in the panel was last read.) This date can be useful when you print a panel. To change the format of this date, see the section on Account Properties in the *Configuration and Customization Guide*.

Banner

The banner that appears at the top of all Device Manager or Port Manager panels consists of several elements.

Element	Example	Notes
Device title and IP address	website.example.com / 192.168.96.1	see Device Title on page 22 if the device title is the IP address, the IP address is shown once if there is no IP address, only the device title is shown
Manager name	Device Manager	—
System name of Enterprise Discovery server	ExampleCorp	see the *Installation and Initial Setup Guide*
Web browser name	Netscape | Internet Explorer	—

Device Title

The title displayed in the banner of Device Manager or Port Manager (and in some panels of those managers) will be the first available of:

- a device title chosen by the Enterprise Discovery Administrator in **Administration** > **System preferences** > **Display preferences**. The Enterprise Discovery Administrator can choose one or several of the following and choose their order too:
  - Asset Tag
  - BIOS Asset Tag
  - NetBIOS Name (scan)
  - Last Name
  - First name
  - Device-specific title
  - Domain name
  - NetBIOS name (network)
  - Operating system
  - Family
  - Model

- Network function
- System description
- System name
- System location
- System contact
- IPv6 address
- IPv4 address
- MAC address including OUI
- MAC address (all-numeric)

Note: Only Administrator or IT Manager accounts can change device titles (**Device Properties** button on the Device Manager). The device titles are global. To determine the default title for a device, see the Diagnosis panel on the Device Manager.

CHAPTER 3

Recorded Events

The Health Panel summarizes changes to the network. Each device should contribute only a single alarm. (If there is more than one alarm per device or per port, they will be displayed in the Device Manager or Port Manager.)

To see what alarms are raised by report data or attribute data, see **Help** > **Classifications** > **Alarms**.

Port Add/Deletes

Identifies ports recently added to or deleted from a device. (An added port may or may not be recently discovered.)

Port Changes

Interface rate, duplex.

Device Adds/Deletes

Identifies devices recently added to or deleted from the database. (An added device may or may not be recently discovered.)

Device Changes

Changing icon, priority, title, or tag of a device.

Exceptions

Devices with exceptions. See **Help** > **Classifications** > **Exceptions**.

Not Recently Seen

There are two types of “not recently seen” events:

- Network Not Recently Seen
- Scan Not Recently Seen

“Network Not Recently Seen” devices are those with which Enterprise Discovery has lost contact and which may soon disappear from the database.

Note: Once Enterprise Discovery has not had contact with a device for a period greater than the threshold (by default, 6 hours), it will appear as “Not Recently Seen.”

“Scan Not Recently Seen” devices are devices for which Enterprise Discovery has not received an updated scan file (by default, 4 weeks and 2 days).

Note: You can change these defaults at **Administration** > **System preferences** > **Report time periods**.

CHAPTER 4 Scanners

The Scanner used to scan each computer can capture any or all of the following types of information, depending on the options selected when the Scanner was configured:

- Information about the hardware configuration.
- Information about the system configuration.
- Information about the software on the drives scanned.
- Information about the physical assets and user details that are recorded using the asset questionnaire.

The information collected by a Scanner is stored in a Fingerprint Save File (FSF) or Compressed XML File (XSF) file. This information can be viewed immediately with Enterprise Discovery Analysis Workbench or Viewer, or can be migrated to a third party application.

The key criteria for collecting data is to keep the amount of time spent at each computer to a minimum. It is not realistic to expect users to stop work for more than a few minutes. The process will run a lot more smoothly if users are notified in advance, preferably with an explanation of the purpose of the inventory.

Accuracy is a very important issue. It is cheaper to fix problems at the time of data collection rather than at a later date. Correct Asset Numbers are vital to ensure that the scan files are unique. Data consistency for product names and descriptions are important for reporting purposes.

Select the Scanner that is most appropriate for the computer to be inventoried.

The Scanner Types

Scanners can be generated for the following operating systems:

Scanner	Scanner default name	Platform
DOS Scanner	ScanDos.exe	Dos 16-bit
Windows 16-bit Scanner	ScanW16.exe	Windows 3.1x
Windows 32-bit Scanner	ScanW32.exe	■ Windows 95 ■ Windows 98 (includes Windows 98 SE) ■ Windows NT 4.0 (includes Windows NT Server) ■ Windows ME ■ Windows 2000 (includes Windows 2000 Server) ■ Windows XP (includes 64-bit version) ■ Windows 2003 Server (includes 64-bit version) ■ XP Media Centre ■ Windows for Tablet PCs
The Remote (Windows 32-bit) Scanner	ScanR32.exe	■ Windows 95 ■ Windows 98 (includes Windows 98 SE) ■ Windows NT 4.0 (includes Windows NT Server) ■ Windows ME ■ Windows 2000 (includes Windows 2000 Server) ■ Windows XP (includes 64-bit version) ■ Windows 2003 Server (includes 64-bit version) ■ XP Media Centre ■ Windows for Tablet PCs
OS/2 Scanner	ScanOs2.exe	OS/2 2.1 and OS/2 Warp

Scanner	Scanner default name	Platform
Solaris Scanner	scansp2	Solaris 2.5, 2.6, 7, 8 and 9 on SPARC
HP-UX Scanner	scanhpx	HP-UX 10.2 and 11.0 on HPPA
AIX Scanner	scanaix	AIX 4.3, 5.0, 5.1, 5.2, 5.3 on IBM R6000
Linux Scanner	scanLnx	Any distribution with a 2.2x, 2.4x or 2.6x kernel on i386
MSI Scanner	msiscanner.exe This Scanner is located in C:\Program Files\Peregrine\Enterprise Discovery\2.0.0\Common\bin	■ Windows 95 ■ Windows 98 (includes Windows 98 SE) ■ Windows NT 4.0 (includes Windows NT Server) ■ Windows ME ■ Windows 2000 (includes Windows 2000 Server) ■ Windows XP (includes 64-bit version) ■ Windows 2003 Server (includes 64-bit version) ■ XP Media Centre ■ Windows for Tablet PCs

The procedure for starting a Scanner depends on the native operating system environment for the computer being scanned.

Viewing the Results of the Scan

Peregrine Enterprise Discovery comes with the Viewer program, which allows you to look at the results of your scans. Refer to the *Viewer* Chapter for more information about how to use this application.

For XSF scan files, a tool such as gzip or Winzip can be used to extract the XML data contained in them. The XML file contained inside the XSF file can be viewed with any text editor or XML viewer such as Internet Explorer.

Command Line Options and Switches

Although the options for the Scanner are normally set using the Scanner Generator, it may be necessary to change some settings to allow better operation on some machines. The operation of a Scanner can be modified with the use of the various command line options.

Reasons for Overriding the Options in a Configured Scanner

- The Scanner may encounter a problem with a particular hardware. Using command line options, the problem hardware can be circumvented.
- Command line options can change the configured options such as save path. This allows the scan results to be saved to a local machine without a full network path having to be defined.

How to Use a Command Line Option or Switch

You can specify command line options and switches by:

- Typing the command from a command line (for example, the Windows command prompt, or the UNIX command line). In any case, make sure you supply the path to the Scanner.

  For example:

  ```
  /tmp/scanLnx -?
  ```

  launches the Linux Scanner from the /tmp directory.

- Creating a Windows shortcut. Type the command line option or switch after the quotation marks.

  For example:

  ```
  "C:\Program Files\Peregrine\Enterprise Discovery\2.0.0\Scanner
  Generator\ScanW32.exe" /?
  ```

  launches the Win32 Scanner and displays a list of valid command line options.

- Typing the command in the Windows Run command in the Start menu. Type in or navigate to the location where the Scanner executable is located. Type the command line option or switch after the quotation marks.

  For example:

  ```
  "C:\Program Files\Peregrine\Enterprise Discovery\2.0.0\Scanner
  Generator\ScanW16.exe" /?
  ```

Command Line Switch Format

When specifying command line switches use the following command format:

ScannerName [switch]

In the following example, the -10 switch causes the Scanner to scan the C drive and excludes the BIOS from the hardware tests:

```
ScanW32 C: -10
```

or

```
ScanW32 C: /10
```

Where:

- `ScanW32` is the Windows-based 32-bit [Scanner]
- `-10` or `/10` is the [switch]

Important: UNIX is case-sensitive. Use the exact name of the Scanner and switches (all switches are lower case).

Command Line Options for Scanners

Valid command line options for the Windows, DOS, UNIX and OS/2 Scanners are shown in the following table:

Note: Only the - prefix is used for the UNIX Scanners. You can use the – or / prefix for all of the other Scanners.

Command Line Option	Function	Scanners
-force or /force	Do not check disk space saving offsite Scan File. This may be useful in situations where the operating system reports insufficient space, but this is actually due to access rights.	Windows DOS OS/2 UNIX
-noarc or /noarc	Disables archive processing.	Windows DOS OS/2
-noplug or /noplug	Do not use plug-ins.	Windows DOS OS/2
-noclassic or /noclassic	Do not use classic software scanning method.	Windows DOS OS/2
-noshortcut or /noshortcut	Do not use shortcuts for software scanning.	Windows DOS OS/2
The noclassic and noshortcut switches are only useful when the Combined software scanning mode is used. In Combined mode both Classic and Targeted scans are enabled. Then by using one of these switches you can disable one of these two methods. If you use are using a Targeted scan for example, and try to disable it by using the -noshortcut switch the Scanner will not do any software scanning.		
-noenv or /noenv	Do not use environment variables for software scanning.	Windows DOS OS/2
-nosvc or /nosvc	Do not use services for software scanning.	Windows DOS OS/2
-nofileassoc	Do not use file associations for software scanning.	Windows DOS OS/2

Command Line Option	Function	Scanners

-p:<path> or /p:<path>	Override default offsite save path.
	For example:
	`ScanDos /p:r:\results`
	A UNC path can also be entered as the argument to this option. The format for a UNC path is:
	`\\servername\sharename\path\`
	For example:
	`ScanW32 -p:\\Peregrine\DI\scanfiles\`
	The Scanner must have Write permissions to the specified UNC path.
		Windows
		DOS
		OS/2
		UNIX

-r:<path> or /r:<path>	Override default path for refilling.
	For example:
	`ScanDos /r:r:\results`
	A UNC path can also be entered as the argument to this option. The format for a UNC path is:
	`\\servername\sharename\path\`
	For example:
	`ScanW32 -r:\\Peregrine\DI\scanfiles\`
	The Scanner must have read permissions to the specified UNC path.
		Windows
		DOS
		OS/2
		UNIX

Command Line Option	Function	Scanners

-scandays:<Count> or
/scandays:<Count>
	Scan only if previous scan was more than Count days ago.
	Forces the Scanner to perform the scan only if the previous scan was <N> or more days ago. For example:
	`-scandays:7`
	For example, if the Scanner is launched from a login script every day, it will only perform the scan every week.
	When the scandays:<N> parameter is specified, the Scanner attempts to check when the last scan was run. If no previous scan file is found, no messages are displayed and the scan runs.
	If a scan file is found, the following message is added to the log file:
	`"Checking the age of Scan File "%s"`
	Where %s is the full name of the scan file it uses to check it.
	If there is a problem determining the age of the scan file (for example, if it is a newer version or it is corrupt), it then outputs:
	The age of the Scan File cannot be determined.
	If it does manage to obtain the date, it outputs:
	`Last scan was %d days ago`
	Where %d is substituted for an integer number.
		Windows
		DOS
		OS/2
		UNIX

Command Line Option	Function	Scanners

-scandayofweek:<Number> or
/scandayofweek:<Number>
	Scan only on specified day of week(0-Sun,1-Mon, etc).
	<N> can be one of the following:
	`0-Sunday`
	`1-Monday`
	`2-Tuesday`
	`3-Wednesday`
	`4-Thursday`
	`5-Friday`
	`6-Saturday`
	For example:
	`-scandayofweek:5`
	This will cause the scan to be performed on Fridays only.
	The scandays: and scandayofweek: options can be combined. For example:
	`ScanW32 -scandays:14 -scandayofweek:3`
	This causes the scan to be performed every other Wednesday.
		Windows
		DOS
		OS/2
		UNIX

-noems or /noems	Prevents EMS memory from being used.	DOS
-mono or /mono	Forces monochrome mode (UI is shown without colors).	DOS OS/2
-displayassets	Print the asset questionnaire and exit	UNIX
-incl:<switch>	Switches for re-enabling individual hardware tests that were disabled in the Scanner Generator. To include tests 10, 20 and 50, you would run: `-incl:10 -incl:20 -incl:50` See Using Command Line Switches to Enable and Disable Specific Hardware Tests (PC Scanners) on page 41 for a table of hardware tests.	UNIX

Command Line Option	Function	Scanners
-excl:< switch >	Switches for disabling individual hardware tests. To To exclude tests 10, 20 and 50, you would run: `-excl:10 -excl:20 -excl:50` See Using Command Line Switches to Enable and Disable Specific Hardware Tests (PC Scanners) on page 41 for a table of hardware tests.	UNIX

-assets	Specifies a file that can contain the asset information and will automatically populate the fields.
	In Scanner Generator you add the asset fields that you want to populate. For example:
	`Asset Tag`
	`Last Name`
	`First Name`
	Then in a text file in UNIX you enter the hardware field names:
	`hwAssetTag=Solaris`
	`hwAssetUserLastName=Smith`
	`hwAssetUserFirstName=John`
	Run the following command:
	`./scansp2 -assets:<filename>`
	For every asset field in Scanner Generator you have to have a matching Hardware field in the text file on the UNIX machine or it will not work.
		UNIX

-exit_vpc	Terminates the Scanner if it was launched inside Virtual PC.	Windows 32-bit
-exit_ts	Terminates the Scanner if it was launched from within a Windows Terminal Services session.	Windows 32-bit
-exit_vm	Terminates the Scanner if it was launched inside a VMWare virtual machine.	Windows 32-bit Linux

Command Line Option	Function	Scanners
-paths	Using this switch, it is possible to define exactly which directories to scan; the parameter can be repeated as many times as necessary. For example: `scan -paths:/etc -paths:/var -paths:/bin` will scan just /etc, /var and /bin and their subdirectories.	UNIX
-unhide	Run the hidden Scanner normally.	Windows 32-bit Linux
-hidden	Run the Scanner hidden.	Windows 32-bit Linux
-hiddennoerrors	Run the Scanner hidden with no errors reported.	Windows 32-bit Linux
<paths>	If one or more paths are specified only these paths are scanned. For example: `Scan C:\ D:\Test` will scan the C: drive and D:\Test and subdirectories only. This switch is equivalent to the UNIX -paths: switch	Windows 32-bit
-o:<filename> or /o:<filename>	Takes the offsite scan file name from the command line. For example (non UNIX): `ScanW32 /o:r:\results\SC002154` Where `r:\results\SC002154` is the path to the file `SC002154`. If a file name is not entered, the file is named Default.fsf or Default.xsf. If the path is not specified, the file is placed in the directory configured for offsite scan files in the Scanner Generator (see the *Customization Guide*)	Windows DOS OS/2 UNIX
-<switch> or +<switch>	Disable or Enable hardware tests. See Using Command Line Switches to Enable and Disable Specific Hardware Tests (PC Scanners) on page 41.	Windows DOS OS/2

Command Line Option	Function	Scanners
-reaudit or /reaudit	Performs an automatic re-scan with the Remote Scanner. An optional scan file name can be specified in this command line option. This gives the name of the scan file from which the Scanner reads the hardware data. The full syntax is: /Reaudit[HardwareScanFileName] For more information about the Remote Scanner see The Remote Scanner on page 64.	Windows Remote Scanner
-? or /?	The full list of command line options can be obtained by running the Scanners with the -? or /? command line option.	Windows DOS OS/2 UNIX

Command Line Software Data Override Option

An option is available which can override the software selection configured in the Scanner Generator by specifying a list of drives and/or directories to scan on the command line.

Note: You must ensure that the Allow Command Line Override option is checked in the Scanner Generator Software Data tab for this to work.

Use the following command format:

```
Scanner {scan location} {option}
```

Scan locations can either be drives (such as C: or F:) or can be directories (C:\Windows)

Multiple scan locations can be specified by separating them with a space.

An example of a command line override is:

```
ScanW32 C: N: Z: -noplug
```

or

```
ScanW32 C:\Windows C:\Programs D: -noplug
```

Where:

- `ScanW32` is the Windows-based 32-bit [Scanner].
- `C:\Windows C:\Programs D:` are the [scan locations].
- `-noplug` is the [option].

This will override the software scan selection made in the Scanner Generator and will scan the locations specified. No plug-ins will be used.

Note: When the software scan selection is changed in this way, the local scan file (local$.fsf or local$.xsf) is only created if the offsite scan file is disabled.

Using Command Line Options to Scan Special Hard Drives

Drives that come under this category are:

- Drives that are not supported by the current operating system (such as FAT32 drives in Windows NT Version 4 or below or NTFS drives in DOS).
- FAT partitions not readable by DOS. Normally all FAT drives are accessible by DOS. The exception is multiple primary FAT partitions on the same physical hard disk.
- HPFS drives when not running OS/2.
- NTFS drives when not running Windows NT/2000/XP/2003.
- Drives where access to the drives may be restricted (such as Boot Manager).
- Partitions that are valid, but are not assigned a drive letter in Windows NT/2000/XP/2003.

Because these drives are not assigned drive letters by the operating system, the Scanner assigns “virtual” drive letters to them. Such virtual drive letters are in lower case, making it easy to distinguish them from normal drive letters.

When running the Scanner, any drive letters specified on the command line are case-sensitive. To scan your C: drive, use an upper case C.

In some cases, it may be desirable to force a scan of one or more “virtual” partitions. To do this, specify the virtual drive letter, followed by a colon.

Example

'scan a:' will scan the first virtual partition, if it exists. If both an A: and a: drive exist then because the command line is case-sensitive, drive a: will be scanned.

Example:

```
ScanW16 c: b: -scandayofweek:4
```

or

```
ScanW16 c: b: /scandayofweek:4
```

Where:

- ScanW16 is the Windows-based 16-bit [Scanner].
- c: and b: are the drives to scan [drives].
- -scandayofweek:4 or /scandayofweek:4 is the [option].

Occasionally, it may be useful to run a Scanner without scanning any software. Because it is unlikely that a virtual drive letter z will be used (because this would require 26 unsupported or unmounted partitions to exist), this can normally be achieved by running the Scanner with z: as the parameter:

```
scan z:
```

Viewing Command Line Options in Viewer or Analysis Workbench

If a command line option or switch has been used, it can be viewed in Analysis Workbench or Viewer.

This can be very useful when you want to check if the scan results were obtained from a Scanner that had been run with any special command line options.

For example, if the Scanner had been run with the -paths command:

```
scan -paths:/etc -paths:/var -paths:/bin
```

The -paths command line option will be displayed in Viewer (System Data folder in the Hardware and Configuration tab page).

Using Command Line Switches to Enable and Disable Specific Hardware Tests (PC Scanners)

Switches for disabling single hardware tests in the PC Scanners are shown in the following table:

Command Line Switch	Command Line Switch
-10 : BIOS Data	-11 : BIOS Extension
-12 : SMBIOS Information	-13 : Compaq Asset Tag
-14 : Plug and Play Version	-30 : Video data
-31 : Monitors	-40 : Port data
-50 : Keyboard and Mouse data	-51 : Hardware Mouse detection
-60 : Disk data	-62 : Partition Table scan
-70 : Memory Data	-71 : XMS freespace detection
-72 : Swap Files	-80 : CPU Data
-81 : Enhanced CPU identification	-82 : Fast CPU speed detection
-90 : Operating System Data	-91 : Device driver files
-92 : Cluster Data	-93 : Services
-94 : Virtual Machine Data	-95 : User profiles
-100 : Storage Data	-101 : Devices
-102 : SCSI/IDE serial numbers	-110 : Network data
-111 : TCP/IP data	-112 : IPX data
-113 : Netbios Data	-114 : Network Shares
-120 : Bus Data	-121 : PCI Cards
-122 : PCMCIA Cards	-123 : MCA Cards
-124 : EISA Cards	-125 : ISA PnP Card detection
-126 : USB Data	-130 : Peripherals
-140 : DMI Information	-141 : DMI 1.x Version detection
-142 : DMI 2.x Version detection	

Using Command Line Switches to Enable and Disable Specific Hardware Tests (UNIX Scanners)

Switches for disabling single hardware tests in the UNIX Scanners are shown in the following table:

Command Line Switch	Command Line Switch
10 : BIOS detection	30 : Video detection
40 : I/O Port detection	50 : Mouse & Keyboard detection
60 : Disk detection	70 : Memory detection
80 : CPU identification	90 : Operating System detection
93: Services	94: Virtual Machine Data
95 : User profiles	100 : SCSI/ASPI detection
110 : Network detection	120 : Bus/Card detection
126 : USB Data detection	130 : Peripherals detection
150: System Configuration	

The Windows Scanners

Supported Platforms for Windows-Based Scanners

Windows-based Scanners can be generated for the following platforms:

Scanner	Platform
Windows 16-bit Scanner	Windows 3.1x
Windows 32-bit Scanner	Windows 95 Windows 98 (includes Win98 SE) Windows NT 4.0 (includes Windows NT Server) Windows ME Windows 2000 (includes Windows 2000 Server) Windows XP (includes 64-bit version) Windows 2003 Server (includes 64-bit version) XP Media Centre Windows for Tablet PCs

Note: The best results can be obtained when running the ‘native’ Scanner for a given platform. On 16-bit Windows (Windows 3.x), use the Win16 Scanner. For all other Windows versions, use the Win32 Scanner.

Information Collected by the Windows-Based Scanners

See the document entitled ‘Data collected by the Scanners’.

Starting the Windows-Based Scanner

To start the Windows-based Scanners:

1. Locate the Scanner using File Manager (Windows 3.x) or Windows Explorer (Windows 95, Windows 98, Windows NT, Windows ME, Windows 2000, Windows XP, Windows 2003).

2. Double-click on the Scanner icon or executable file.

3. The Windows-based Scanners can be started from a command prompt inside 32-bit Windows, in the same way as the DOS Scanner. Use the Program Manager or File Manager to start Windows-based Scanners in Windows 3.x.

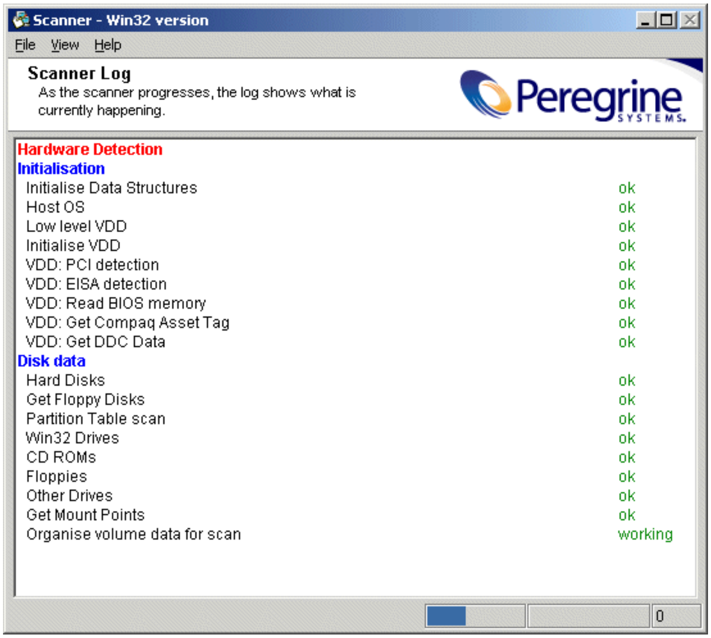
The Scanning Sequence for Windows-Based Scanners

The Windows-based Scanners display a series of screens as they run to provide real-time information about the data being collected:

- The Windows-based Scanner Log
- The Windows-based Scanner Asset Questionnaire
- The Windows-based Scanner Software page
- The Windows-based Scanner Information page

The Windows-Based Scanner Log

The first page displayed is the Scanner Log page which shows details of what is currently happening as the Scanner progresses. Initially, it shows each major hardware component being scanned as well as each specific test performed as each hardware item is scanned.



Scanner Log
As the scanner progresses, the log shows what is currently happening.

```
Hardware Detection
Initialisation
  Initialise Data Structures                          ok
  Host OS                                             ok
  Low level VDD                                       ok
  Initialise VDD                                      ok
  VDD: PCI detection                                  ok
  VDD: EISA detection                                 ok
  VDD: Read BIOS memory                               ok
  VDD: Get Compaq Asset Tag                           ok
  VDD: Get DDC Data                                   ok
Disk data
  Hard Disks                                          ok
  Get Floppy Disks                                    ok
  Partition Table scan                                ok
  Win32 Drives                                        ok
  CD ROMs                                             ok
  Floppies                                            ok
  Other Drives                                        ok
  Get Mount Points                                    ok
  Organise volume data for scan                       working
```

Displaying the Scanner Log During the Scan

To display the Scanner Log during the scan do one of the following:

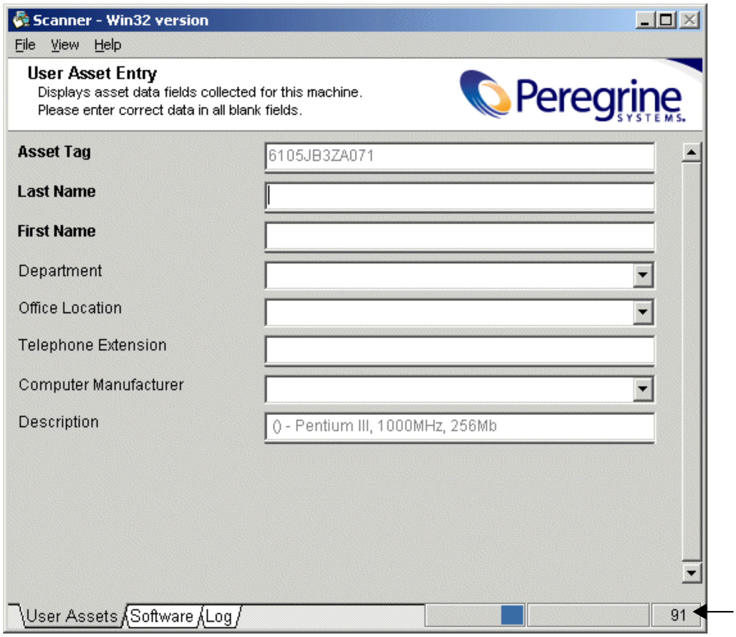
1. Click the Log tab.

2. Select the Log option from the View menu.

3. Use the Ctrl+L keyboard shortcut.

The Windows-Based Scanner Asset Questionnaire

As soon as the hardware scanning is completed, the software scanning commences (this takes place in the background) and the User Asset Entry page (asset questionnaire) is presented on-screen.



User Asset Entry

Displays asset data fields collected for this machine.
Please enter correct data in all blank fields.

Asset Tag	6105JB3ZA071
Last Name	
First Name	
Department	
Office Location	
Telephone Extension	
Computer Manufacturer	
Description	() - Pentium III, 1000MHz, 256Mb

91

Seconds left of the Scanner active for a least' setting.

Entering Asset Information for the Windows-Based Scanners

After the hardware scan is completed, software scanning (if specified) commences. This usually takes place in the background.

Note: For manual scans the Asset Tag entry field is required by default.

The Scanner presents an asset questionnaire for entry of the asset details.

The asset questionnaire consists of a list of prompts on the left side and a list of corresponding data entry fields on the right side. The special field types, which expect particular type of data to be entered, are indicated as follows:

- Required fields - These are denoted in Bold font. These fields must be completed before the scan can complete.
- Pick List/Type Pick List - These fields have a drop-down list of permitted answers which is available by clicking the arrow symbol at the end of the box.

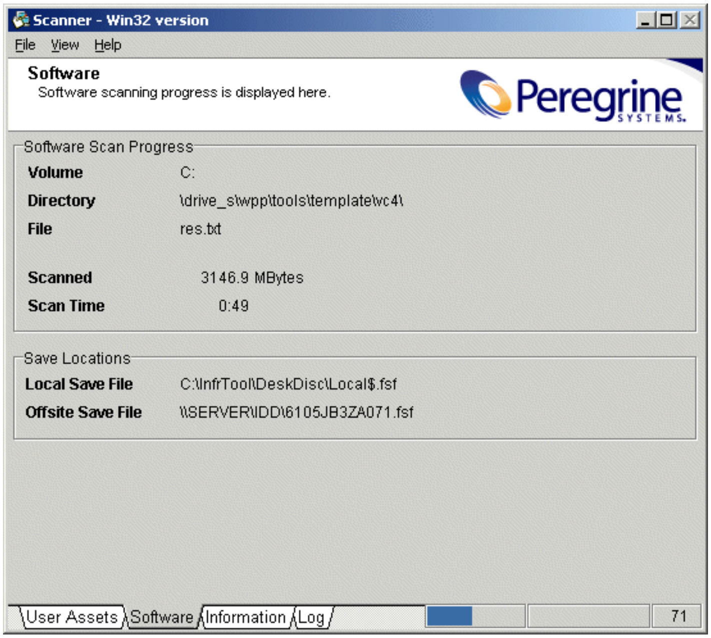
To pause the software scan at any time, select the Pause scan command in the File menu.

Note: If the scan is paused for more than 30 minutes, it automatically resumes and cannot be paused again.

The Scanner is usually active for the period specified in the User Interaction tab of the Scanner Generator – Scanner Options page. See the *Customization Guide* for more information.

The Windows-Based Scanner Software Page

If the default Scanner configuration allows the user to view all pages, the Software page can be displayed as the software information is being scanned. This page shows details of the software being scanned.



Software

Software scanning progress is displayed here.

Software Scan Progress

Volume	C:
Directory	\drive_s\wpp\tools\template\wc4\
File	res.txt
Scanned	3146.9 MBytes
Scan Time	0:49

Save Locations

Local Save File	C:\InfrTool\DeskDisc\Local$.fsf
Offsite Save File	\\SERVER\IDD\6105JB3ZA071.fsf

In addition to showing the progress of the software scan, the names of the scan file(s) to be saved when the scan is complete is shown in the Save locations box.

Displaying the Software Tab Page During the Scan

To display the Software tab page during the scan, do one of the following:

- Click the Software tab.
- Select the Software option from the View menu.
- Use the Ctrl+S keyboard shortcut.

The Windows-Based Scanner Information Page

This tab is only displayed if the Scanner was configured to do so in the Scanner Generator – Scanner Options page. It displays a user-defined bitmap and text while the Scanner is being run.

To display the Information tab page during the scan do one of the following:

- Click the Information tab.
- Select the Information option from the View menu.
- Use the Ctrl+I keyboard shortcut.

Saving, Aborting and Exiting from a Windows Scan

Saving the Scan Data

After the scan is completed, the Scanner automatically saves the scan data to one or two scans files as set up on the Scanner Options page of the Scanner Generator.

If the offsite scan file cannot be written to the location selected when the Scanner was configured, the user will be prompted to select another drive for the file. In this case, the scan file will be saved to the root directory of the drive selected.

The paths to where the Local and Offsite scan files are saved are shown in the Save Locations group in the Software page.

Note: If the local scan file was not saved, the Scanner sometimes may report that there is insufficient room to save an offsite scan file, even though there is enough room. This is due to the fact the Scanner makes an estimate of the resulting file size. The exact file size is not known until the scan file is saved. Save the file to a drive that has enough free space and copy it to the destination location manually.

Aborting the Scan

A Windows-based scan can only be aborted when the scan has not completed, if the Allow Users To Abort Audit option was enabled in the Scanner Generator when the Scanner was configured.

If this option has not been selected, the commands for exiting the scan are not available until the scan is completed.

Exiting from the Scan

To exit from the Scanner, after the time delay following completion of the asset details has expired:

- Select the Exit option from the File menu.
- Use the Alt-X keyboard shortcut.

These options can be used to stop the scan before the process is completed, if the Scanner has been configured to provide this facility.

You can also exit from the Scanner by:

- Selecting the Save & Exit option from the File menu.
- Using the Alt-S keyboard shortcut.

This allows the user to exit the scan as soon as the scan file has been saved (without having to wait for the time delay following completion of the data entry). It therefore provides a quick way of exiting the scan by revoking the time delay set for the minimum time the Scanner is active (selected as a Timers setting in the Scanner Options|User Interaction page of the Scanner Generator).

Note: Alt-S causes the Scanner to verify the validity of the asset data questionnaire. If any required fields are blank or other field restrictions are violated, an error message is displayed and the relevant field is highlighted in the asset questionnaire. Only when the asset questionnaire is complete and valid will the scan file be saved.

The No-UI Version of the Win32 Scanner

This Scanner is a command line application. By default it is called scanN32.exe.

This Scanner can only refill from an existing scan file or use automatic fields for asset data. There is no manual asset entry in this type of Scanner.

Starting the No-UI version of the Win32 Scanner

To start the No-UI Scanner:

1 Exit Windows, and at the DOS prompt, change to the directory where the Scanner program is located.

2 Type the name of the Scanner program, for example, ScanN32, to start the Scanner.

Important: As the scanner has no way to interact with the user, if an error is encountered during its operation, it will fail with the appropriate error level. The reason for failing can usually be found in the error log file.

Windows Scanner Error Level Codes

The Scanners produce error level codes which can be used to handle situations if the Scanner terminates without producing a scan file.

These error codes can, for example, be used in a batch file so that specified actions can be carried out in the event that particular error codes are returned.

These can be used to control re-scan activities when a scan has not completed successfully.

Error Level	Description
20	Scanner terminated because virtual machine was detected (Windows 32-bit only)
6	Another Scanner instance is already running.
5	Too Early – It is earlier than the scan days variable.
4	Fatal Error – Scanner encountered a fatal error.
3	Help Screen – Command line help screen has been requested.
2	User Abort – User aborted the Scanner.
1	Exception – Scanner terminated because of an exception

See Using Error Level codes on page 73 for further information about how to use Error Level codes.

The DOS and OS/2 Scanners

Supported Platforms for DOS and OS/2 Scanners

DOS and OS/2 Scanners can be generated for the following platforms:

Scanner	Platform
DOS Scanner	DOS 16-bit
OS/2 Scanner	OS/2 2.1 and OS/2 Warp

Note: The best results can be obtained when running the ‘native’ Scanner for a given platform.

Information Collected by the DOS and OS/2 Scanners

See the document entitled ‘Data collected by the Scanners’.

Starting the DOS-Based Scanner

To start the DOS-based Scanners:

1. Check that there is enough base memory to run the Scanner (approximately 525K bytes of base memory is needed).

   If this is a problem, try removing TSR programs to release memory.

2. Exit Windows, and at the DOS prompt, change to the directory where the Scanner program is located.

   For a walkround inventory, this will usually be executed from the A: drive; if you are testing the Scanners locally, it will be the directory to which you generated the Scanner in the Scanner Generator.

   The DOS Scanner can also be executed from inside Windows or OS/2, although hardware results collected may differ from those collected under ‘real’ DOS.

3. Type the name of the Scanner program, for example, ScanDos, to start the Scanner.

Note: The DOS Scanner may not pick up all the hardware details that a Windows 32-bit Scanner can pick up. Use the Scanner Generator to check the configuration of the Scanner and that the hardware details and other settings meet the needs of the project. It is highly advisable to use native Scanners in preference to the DOS Scanner wherever possible as they do not have the memory constraints and collect more data.

Starting the OS/2 Scanner

To start the OS/2 Scanner:

1. Open an OS/2 window, and change to the directory containing the Scanner executable file.

2. Type the name of the Scanner program, for example, ScanOs2, to start the Scanner.

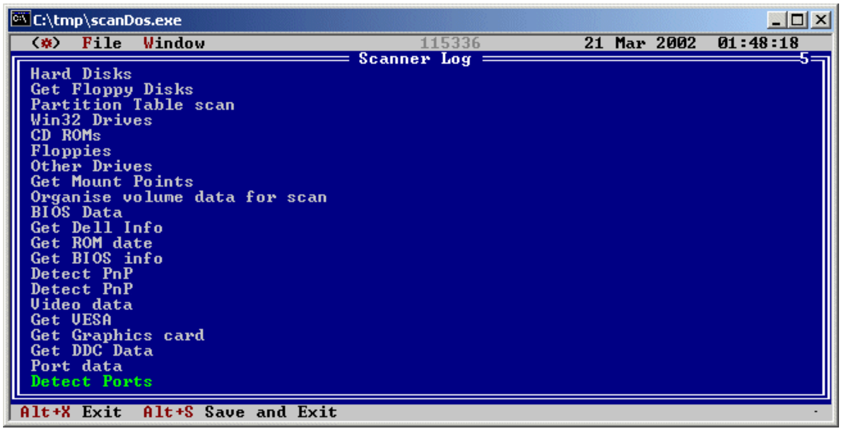
The Scanning Sequence for DOS and OS/2 Scanners

A series of screens is displayed as they run to provide real-time information about the data being collected.

- The DOS and OS/2 Scanners Log page
- The DOS and OS/2 Scanner Asset questionnaire
- The DOS and OS/2 Scanner Software page

The DOS and OS/2 Scanners Log Page

This is the first Scanner page to be displayed. It displays the status of the hardware scanning. After the hardware scan has completed, the asset questionnaire will be displayed.



```
Hard Disks
Get Floppy Disks
Partition Table scan
Win32 Drives
CD ROMs
Floppies
Other Drives
Get Mount Points
Organise volume data for scan
BIOS Data
Get Dell Info
Get ROM date
Get BIOS info
Detect PnP
Detect PnP
Video data
Get VESA
Get Graphics card
Get DDC Data
Port data
Detect Ports
```

Displaying the Hardware Page During the Scan

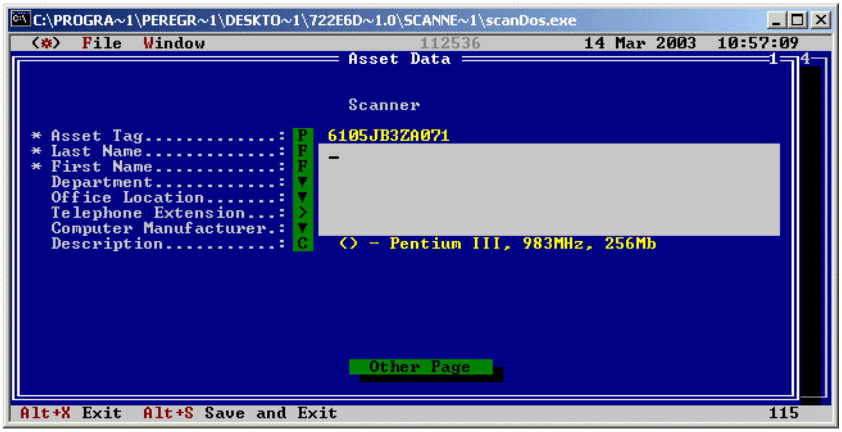
To display the Log page during the scan:

- Press Alt-5.

The DOS and OS/2 Scanner Asset Questionnaire

After the hardware scan is completed, software scanning commences (which usually takes place in the background). The Scanner then presents an asset questionnaire for entry of the asset details.

The first asset data screen which forms the first part of the asset questionnaire (depending on the number of data entry items defined) is displayed.



```
C:\PROGRA~1\PEREGR~1\DESKTO~1\722E6D~1.0\SCANNE~1\scanDos.exe
  (*)  File  Window              112536          14 Mar 2003  10:57:09
                              Asset Data                          1  4

                               Scanner

 * Asset Tag.............: P  6105JB3ZA071
 * Last Name.............: F  _
 * First Name............: F
   Department............: ▼
   Office Location.......: ▼
   Telephone Extension...: >
   Computer Manufacturer.: ▼
   Description...........: C   <> - Pentium III, 983MHz, 256Mb

                             Other Page

 Alt+X Exit  Alt+S Save and Exit                                     115
```

To get to the other page of the asset questionnaire:

- Select the Asset Data Screen 2 (Asset Data Screen 3) option from the Window menu or press Alt-2 (Alt-3) to display the next screen of the asset questionnaire.

The asset data screens provide a screen-based questionnaire, consisting of a list of user prompts on the left side and a list of corresponding data entry fields on the right side. The field type, which specifies the type of data to be entered, is indicated by the symbol to the left side of each data entry field.

The following symbols are used to distinguish the different field types on the asset data screens:

Note: Required fields are displayed with an asterisk '*' to the left of the field name.

Field	Meaning
X	File Extract field.
F	Formatted field which accepts data entered in a predefined format.
E	Environment variable field.
C	Combination field
D	DMI Extract field
P	Sequence field

Field	Meaning
>	Above (Num) field, which accepts numeric data above a specified number.
<	Below (Num) field, which accepts numeric data below a specified number.
|	Range (Num) field, which accepts numeric data input within a defined range.
▼	Pick List (or predefined drop-down list) used to select an entry.

Entering Asset Information for DOS and OS/2 Scanners

To enter the asset information for DOS and OS/2 Scanners:

1. Enter the user and asset information as instructed on-screen. Some of the boxes have a drop-down list of permitted answers which is available by clicking the arrow symbol at the end of the box.

2. Use the Tab key to move to the next field. To move to the previous field, use Shift and Tab.

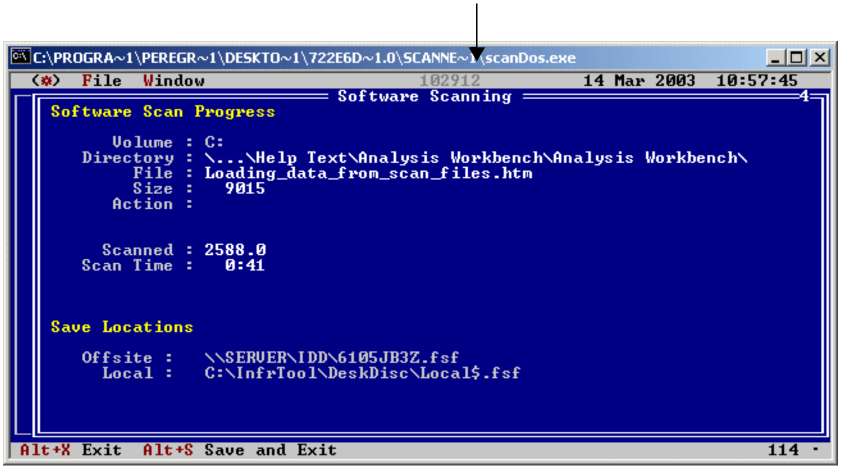
3. To display the drop-down list for a Pick List or Type/Pick List, use the down arrow key to move to the required entry and press Enter to select it.

4. If a field is a required field and the user attempts to move past the field without making or selecting an entry, a message is displayed. The inventory cannot continue until a valid entry is made for the field.

The Scanner is usually active for the period specified in the User Interaction tab of the Scanner Generator – Scanner Options page.

The DOS and OS/2 Scanner Software Page

After the asset details have been entered at run-time, the Software page can be displayed. This page shows details of the software being scanned.



Available memory for scanning

```
C:\PROGRA~1\PEREGR~1\DESKTO~1\722E6D~1.0\SCANNE~1\scanDos.exe
 (*)  File  Window               102912          14 Mar 2003  10:57:45
                                    Software Scanning                  4
   Software Scan Progress

         Volume : C:
      Directory : \...\Help Text\Analysis Workbench\Analysis Workbench\
           File : Loading_data_from_scan_files.htm
           Size :   9015
         Action :


        Scanned : 2588.0
      Scan Time :   0:41



   Save Locations

      Offsite :    \\SERVER\IDD\6105JB3Z.fsf
        Local :    C:\InfrTool\DeskDisc\Local$.fsf


 Alt+X Exit  Alt+S Save and Exit                                      114
```

Displaying the Software Tab Page During the Scan

To display the Software tab page during the scan:

- Select the Software Data Screen option from the Window menu (or press Alt-4) to display the Software Detected screen.

  This screen shows the progress of the scan.

Saving, Aborting and Exiting from a DOS or OS/2 Scan

Saving the Scan Data

After the scan is completed, the Scanner automatically saves the scan data to one or two scan files as defined on the Scanner Options page of the Scanner Generator.

If the offsite scan file cannot be written to the location selected when the Scanner was configured, the user will be prompted to select another drive for the file. In this case, the scan file will be saved to the root directory of the drive selected.

The locations to where the Local and Offsite scan files are saved are shown in the Software page.

Monitoring Memory Usage in a DOS Scan

In DOS, only a limited amount of memory is available for scanning and the Scanner may run out of memory during the software scan on machines with very large numbers of files.

The memory available for scanning is shown at the top of the screen. If this figure gets very low, the scan may fail.

Aborting the Scan

A DOS or OS/2 scan can only be aborted when the scan has not completed, if the Allow Users To Abort Audit option was enabled in the Scanner Generator when the Scanner was configured.

If this option has not been selected, the File|Exit (or Alt-X) and commands for exiting the scan are not available until the scan is completed.

Exiting the Scan

To exit from the Scanner, after the time delay following completion of the asset details has expired:

- Select the Exit option from the File menu.
- Press Alt-X.

These options can be used to stop the scan before the process is completed, if the Scanner has been configured to provide this facility.

You can also exit from the Scanner by

- Selecting the Save & Exit option from the File menu.
- Pressing Alt-S.

This allows the user to exit the scan as soon as the scan file has been saved (without having to wait for the time delay following completion of the data entry).

It therefore provides a quick way of exiting the scan by revoking the time delay set for the minimum time the Scanner is active (selected as a Timers setting in the Scanner Options|User Interaction page of the Scanner Generator).

DOS and OS/2 Scanner Error Level Codes

The Scanners produce error level codes which can be used to handle situations if the Scanner terminates without producing a scan file.

These error codes can, for example, be used in a batch file so that specified actions can be carried out in the event that particular error codes are returned.

These can be used to control re-scan activities when a scan has not completed successfully.

Error Level	Description
6	A plug-in caused problems with the Scanner.
5	Too Early – It is earlier than the scan days variable.
4	Fatal Error – Scanner encountered a fatal error.
3	Help Screen – Command line help screen has been requested.
2	User Abort – User aborted the Scanner.
1	Exception – Scanner terminated because of an exception

See Using Error Level codes on page 73 for further information about how to use Scanner Error Level codes.

The UNIX Scanners

Supported Platforms for the UNIX Scanner

UNIX Scanners can be generated for the following platforms:

Scanner	Platforms
Solaris Scanner	Solaris 2.5, 2.6, 7, 8 and 9 on SPARC
HP-UX Scanner	HP-UX 10.2 and 11.0 on HPPA
AIX Scanner	AIX 4.3, 5.0, 5.1, 5.2, 5.3 on IBM R6000
Linux Scanner	Any distribution with a 2.2x, 2.4x or 2.6x kernel on i386

Information Collected by the UNIX Scanners

See the HTML document entitled *Data collected by the Scanners* which is available from the Documents folder in the Enterprise Discovery Start menu item.

Starting the UNIX Scanner

Note: The methods for starting the various UNIX Scanners (HP-UX, Solaris, Linux and AIX) are identical.

We recommend that you use the UNIX agent and Enterprise Mode to schedule scans regularly.

To run a UNIX Scanner:

1. Copy the Scanner executable to the machine to be scanned.

2. Make sure that executable bit has been set (for example, for the Solaris Scanner run chmod +x scansp2 to ensure this)

3. Type the name of the Scanner, for example, scansp2, followed by any desired Scanner command line options, to run it.

   You will have to type ./ in front of scansp2 if the current directory (.) is not in the PATH: ./scansp2

The Scanning Sequence for UNIX Scanners

After the scan has been executed, the following events take place:

- Hardware scan (also contains system configuration scan)
- Asset Entry
- Software scan

Hardware Scan

Initially a copyright message is displayed, after which, hardware is detected (this too, is indicated as text mode messages).

Asset Entry

Asset entry is only supported in Manual Deployment mode. Automatic deployment of Unix Scanners will not expect any user asset entry. If a required asset field that is empty is encountered, the Scanner will fail.

It is still possible to provide asset questionnaire input to the Scanner running in Enterprise Mode by placing the file called **response.txt** into the agent data directory (this file is usually located in $home/.discagnt).

If a Scanner running in Enterprise Mode finds such a file it will be used as if it was specified in the **-assets** command line parameter to provide input for the asset questionnaire.

After hardware detection has completed the Scanner prompts for the asset data to be entered. You will be presented with each question in turn.

A prompt is symbolized by the ? character.

Entering Data

There are several possible ways in which to enter data into the questionnaire depending on how the field was configured in the Scanner Generator:

- To retain a value that was previously used, just hit Enter as the response.
- To enter a new value, just type it and hit Enter.
- To pick an item from a pick list, type the number of the list entry. If you type anything other than a valid number in the list, the entry is taken literally and stored as the value.
- To enter a blank response, press the enter key. This will blank the field value. If this is not valid (the field is required), the question will be asked again.
- If a question has a value the user is not permitted to change (because the field was configured as ‘editable if blank, read-only otherwise’), the value is displayed but you will not be prompted for a value.

Examples of Asset Questionnaire Entries

Examples of questionnaire entries may look as follows:

Example 1 - Overriding a Previous Value

```
+ get asset data
--------------------------------
Asset Tag; Required field
Max Width: 32
Current Value: MyMachine
? Enter new value: Frodo
--------------------------------
```

This field is the asset tag field, which has overridden the previous value of MyMachine with Frodo.

Example 2 - Required Fields That Cannot Be Edited

```
Last Name; Required field
Max Width: 30
Current Value: Mertner
--------------------------------
First Name; Required field
Max Width: 30
Current Value: Allan
--------------------------------
```

These two fields are required fields and have been set up not to be editable if they have a value, so no prompt is displayed, only the field description, attributes and current value are displayed.

Example 3 - Overriding a Pick List Entry

```
Department
Max Width: 25
Current Value: Accounts

1 Accounts
2 Corporate Finance
3 Customer Support
4 Development
5 Facilities Management
6 Human Resources
? Enter new value: 4
---------------------------------------
```

This is a Pick List, Department, with a value of Accounts. To override it to have a value Development, type 4 which selects the 4th item - Development as the value

Example 4 - Retaining a Pick List Entry

```
Office Location
Max Width: 30
Current Value: Canada

1 Belgium
2 Canada
3 Denmark
4 Finland
5 France
6 Germany
? Enter new value: 2
```

This field is office location and has a value Canada. To keep this value, type 2 as the reply.

After the asset questionnaire has completed, the software scan will start.

Software Scan

The software scan commences after the asset data has been entered. It shows a list of directories as they are being scanned.

The Scanner by default is configured to perform a ‘Targeted Directory Scan’, which means that the Scanner will not scan the entire machine but only those directories pointed to by the PATH and LIBPATH environment variables (and subdirectories). This can be changed in the Scanner Generator Software Data page.

Saving, Aborting and Exiting from a UNIX Scan

Saving the Scan Data

After the scan is completed, the Scanner automatically saves the scan data to one or two scan files as defined on the Scanner Options page of the Scanner Generator.

Aborting the Scan

To exit from the UNIX scan prematurely press Crtl C. The results of the scan will not be saved.

Exiting from the Scan

After the UNIX Scanner has finished, it automatically returns to the command line prompt.

UNIX Scanner Error Level Codes

The Scanners produce error level codes which can be used to handle situations if the Scanner terminates without producing a scan file.

These error codes can, for example, be used in a batch file so that specified actions can be carried out in the event that particular error codes are returned.

These can be used to control re-scan activities when a scan has not completed successfully.

The following table shows a list of error level codes found in the UNIX Scanners.

Return code	Meaning
0	Normal exit
1	This code is returned if the user does not input asset data in the specified time.
20	Scanner terminated because Virtual Machine was detected (Linux only).
-1	Gets the asset data from the file specified on the command line, but the required field does not exist in the file.
-54	In the specified asset file on the command line, each line must contain something similar to hwxxxxxx=yyyyyy: If the total length of the line is less than 5, it is in the wrong format. If the line does not contains "=" to separate ID and value, it is in the wrong format. Asset data is specified more than once. The user specified two much asset data.
-56	Insufficient access rights There is not enough disk space to create the offsite fsf file. The local save path exists, but it is not a directory, it is a normal file. No write permission when creating a file.

Return code	Meaning
-57	Object not found In the specified asset file on the command line, the symbolic name of the field ID (left to "=") is incorrect. The user specified an invalid refill path on the command line. The save path specified, but does not exist. The specified asset file in the command line does not exist.
-65	A generic error at the OS level Unable to load string resources (Memory shortage).
-2010	In the specified asset file, a value is invalid for the asset field.

The Remote Scanner

Remote Scanner Overview

The Remote Scanner is generated from the Scanner Generator along with the other Scanners and is used to perform a software scan of a remote computer.

Important: The XSF scan file format is not supported by the Remote Scanner

The Remote Scanner provides the ability to remotely scan a machine which is accessible across a network. This means that the Scanner software does not run on the target computer, but obtains its hardware detail separately.

Providing the person performing the inventory has access to the local drives, this could be for example:

- A server, if there are network shares available.
- A machine performing a non-interruptible task.

Note: Automatic asset fields, such as File, Environment, Registry, WMI Extract and DMI Extract do not work with the Remote Scanner as it does not run on the computer being scanned.

Even if certain parameters in the Scanner Generator have been set, the Remote Scanner enforces the following behavior:

- The local scan file is never saved, but the offsite scan file is always saved.
- The asset number batch file (Asset.bat by default) is never created.
- Refilling from the local scan file is never performed.

Executing a Remote Scanner

The triggering of a Remote Scanner is best done from a command prompt. Here the Remote Scanner can be called and the save path and name of the scan file can be defined.

A sample of the syntax is as follows:

```
ScanR32.exe /o:M:\path\filename
```

Where:

- ScanR32.exe is the Scanner defined with just software collection.
- /o: is the save offsite command line option.
- M:\path\filename is the destination and name for the resulting scan file.

Note: There must be no space between /o: and path.

If there is a space between /o: and the path, the file will be named default.fsf and will be put in the default save location configured in the Scanner Generator.

If the .fsf extension is not supplied, it is assumed. If any other extension is specified, it will be disregarded and replaced with .fsf.

Important: The XSF scan file format is not supported by the Remote Scanner

Example

```
Remtlr32 /o:G:\Scan\Test
```

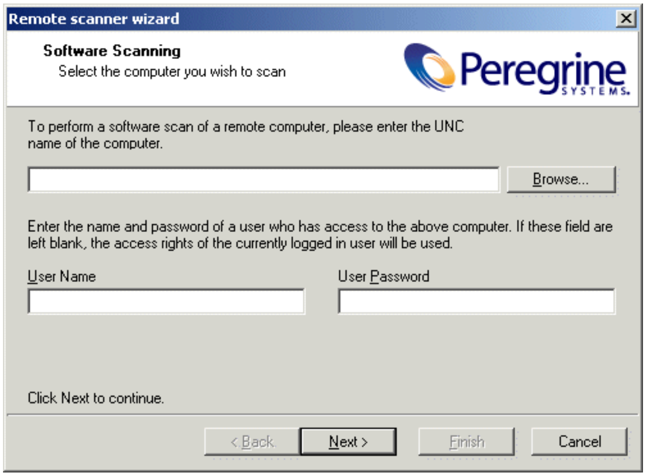
The example will run a Remote Scanner named Remtlr32 from the command line, save it to a mapped drive letter G: and the directory Scan. The scan file will be named Test.fsf.

The user who performs the scan must have sufficient privileges to access files on the volumes that they wish to scan.

If software information about all files on the remote volume is required, make sure to use the network connection that is mapped to the root directory of the target volume.

The Remote Scanner - Software Scanning Page

On starting the Remote Scanner, the Software Scanning page is displayed, which asks for the name of the remote computer to be scanned.



Remote scanner wizard

Software Scanning

Select the computer you wish to scan

To perform a software scan of a remote computer, please enter the UNC name of the computer.

Enter the name and password of a user who has access to the above computer. If these field are left blank, the access rights of the currently logged in user will be used.

User Name

User Password

Click Next to continue.

1 Enter the UNC network name of the computer or use the Browse… button to select the network share from the Network Neighborhood style dialog box.

   If a computer name is not entered, no software scan is performed and the Remote Scanner can only be used to enter the asset and hardware details. The Scanner Generator will skip the second page and go straight to the Hardware Data page when the Next button is clicked.

2 Enter the name and password of a user who has access to the shares on the remote computer specified. The User Name and User Password boxes are only available under Windows NT/2000/XP/2003. If these boxes are left

blank, the access rights of the user currently logged into the client machine will be used.

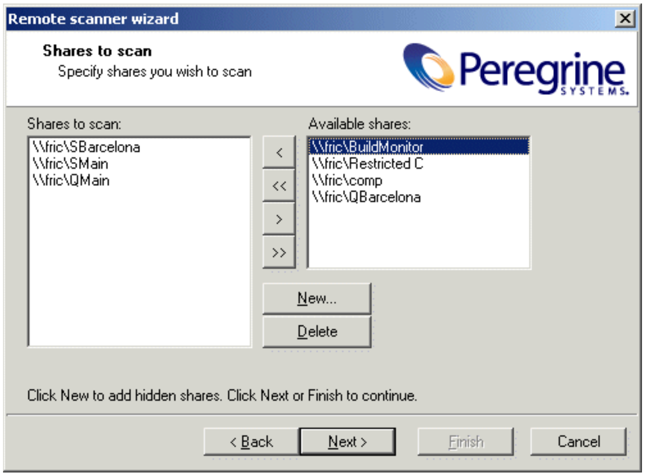
Note: If automatic re-scans are planned, do not use the user name and password for security reasons. The /Reaudit command line option which is used for re-scans does not allow the user and password to be specified.

3 Click the Next button to continue.

The Remote Scanner - Specifying the Shares to Scan

The second page is the Shares to scan page.

This page is shown when the proper UNC name of a computer is entered and allows you to enter the list of shares that need to be scanned on that computer.



Remote scanner wizard

Shares to scan

Specify shares you wish to scan

Shares to scan:

\\fric\SBarcelona
\\fric\SMain
\\fric\QMain

Available shares:

\\fric\BuildMonitor
\\fric\Restricted C
\\fric\comp
\\fric\QBarcelona

Click New to add hidden shares. Click Next or Finish to continue.

To specify the shares that need to be scanned on the remote computer:

1 Drag the available shares into the Shares to scan list box or use the buttons in the centre of the dialog box.

- < Move a single item from the Available shares list into the Shares to scan list.
- << Move all items from the Available shares list into the Shares to scan list.

- > Remove a single item from the Shares to scan list to the Available shares list.
- >> Remove all items from the Shares to scan list to the Available shares list.

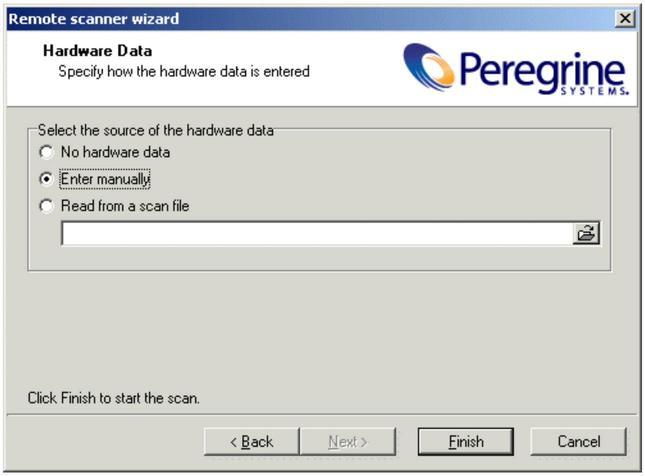
2 Some shares are hidden and therefore are not visible when browsing the network (for example, shares ending with $ are hidden in Windows NT/2000/XP/2003). To specify such shares, use the New… button. A prompt is displayed, which will ask you to enter the name of the specific share to scan.

3 You can use the Delete button to removes hidden shares from the list, if they were entered by mistake.

4 Click the Next button to continue.

The Remote Scanner Hardware Data Page

The next page is the Hardware Data page, which asks where the hardware data is to be taken from.



Remote scanner wizard

Hardware Data

Specify how the hardware data is entered

Select the source of the hardware data

○ No hardware data

● Enter manually

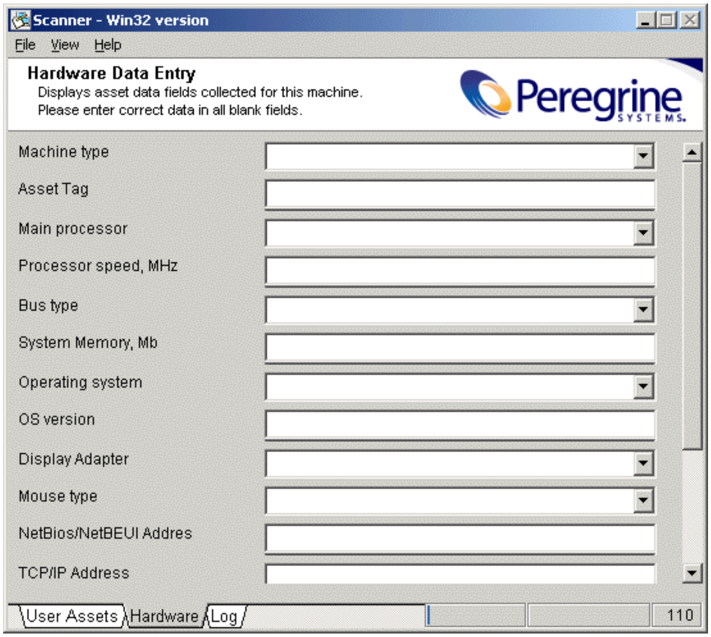
○ Read from a scan file

Click Finish to start the scan.

The hardware data can be read from another scan file. This is useful when performing separate hardware and software scans, for example, when scanning Novell NetWare. Otherwise, the key hardware items can be entered by hand.

Select the source of the hardware data

- No Hardware Data - No hardware data is included.
- Enter manually - Hardware data is manually entered.



Scanner - Win32 version

Hardware Data Entry

Displays asset data fields collected for this machine.
Please enter correct data in all blank fields.

- Read from a scan file - The hardware data is read from an existing scan file. For example, this could be a scan file captured when a machine was locally scanned. For example, a NetWare server can be scanned for hardware data using the DOS Scanner, when booted from a bootable floppy disk.

  Enter the name and path of the scan file or click the browse button to navigate to where the file is located.

  Click the Finish button to start the scan.

  The scan progresses and saves the results to Default.fsf or Default.xsf. This default name can be overridden by using the /o option. For example,

  ```
  ScanR32 /o:<filename>
  ```

  will remotely scan and save data to the specified file name.

The Remote Scanner cannot be generated if the following asset fields have been configured in Scanner Generator:

- Registry Extract
- File Extract
- DMI Extract
- Environment Extract

Performing an Automatic Re-scan with the Remote Scanner

If the asset data collection was enabled and the Scanner was configured to do the refilling from an offsite scan file, the asset and the hardware data will be automatically refilled from the offsite scan file.

The /Reaudit option is used to facilitate a completely automatic re-scan. This command line option causes the Remote Scanner to re-scan the shares that were originally scanned and stored in the offsite scan file. If the Scanner cannot find the offsite scan file to read the list of shares from, it terminates.

An optional scan file name can be specified in this command line option. This gives the name of the scan file from which the Scanner reads the hardware data.

The full syntax is:

```
/Reaudit[HardwareScanFileName]
```

The MSI Scanner

In This Section...

Overview of the MSI Scanner

The MSI Scanner is a command line utility used to scan an MSI based installer, extract all required file information and write an XML file describing the installer and its contents. This XML file can then be sent to the central office where the person maintaining the application library can load it into the SAI Editor exactly as if it was the original MSI based Installer.

Important: The MSI Scanner (msiscanner.exe) is not generated by the Scanner Generator. It is supplied with the software in the following location by default: C:\Program Files\Peregrine\Enterprise Discovery\2.0.0\Common\bin

Starting the MSI Scanner

To start the MSI Scanner:

◆ From the command prompt, type the following:

```
msiscanner <setup_package> <output_file>
```

Where:

- <setup_package> is the path and file name of the MSI-based installer.
- <output_file> is the path and file name of the output XML file. Note that if the specified file name does not end in .xml, the MSI Scanner will append an .xml extension to it.

Opening the MSI Scanner Output File in the MSI Importer

The output from the MSI Scanner is usable in the MSI Importer so that you can browse the MSI and teach from it based on the XML file only.

To open the MSI Scanner output file:

1. In the SAI Editor, select the Import MSI based Installer option from the Tools menu.

   The File Open dialog box is displayed.

2. In the Files of Type drop-down box, select the MSI Scanner output file.

3. Navigate to the file to be opened.

4. Click OK

MSI Scanner Error Level Codes

The MSI Scanner produce error level codes which can be used to handle situations if the Scanner terminates without producing a scan file.

These error codes can, for example, be used in a batch file so that specified actions can be carried out in the event that particular error codes are returned.

Error Level	Description
6	Unexpected error
5	Unable to open the output file
4	Insufficient space available in the Temp directory.
3	Unable to open input MSI
2	Unrecognized package
1	Incorrect parameters
0	Success

Troubleshooting

Using Error Level codes

Windows, DOS and OS/2 Scanners produce Error Level codes that can be used to handle situations if the Scanners terminate without producing an audit file.

These can be used to control re-scan activities when a scan has not completed for some reason.

Because the Error Level is available as an environment variable when the Scanner finishes this can be incorporated in a log file.

For example, a Windows NT/200x/XP Scanner with ComputerName and UserName available, a simple batch file could include:

```
echo %computername%, %errorlevel%, %username% to a flag file %
%computername%.flg
```

Note: If the Scanner is terminated using the Windows Task Manager, then it is reported as successful.

Files Affected by Scanning

File	Location
ASSET.BAT	C:\if selected
Local$.FSF	PDI version 7.x and Windows 2000 C:\Documents and Settings\All Users\application Data\Peregrine
FP_000x.TMP Where x=0,1,2,3 etc.	C:\Documents and Settings\username\Local Settings\Temp Created while the Scanner is running and removed on successful completion. If a scan fails and is subsequently rerun, the temporary files allow the asset data to be filled into the asset fields to save retyping.
Scanner.ini	In the local save directory. This is used to keep track of whether migration from PDI was done or not.
Output.FSF	Defined by Scanner ot/p: or /o:
OVERRIDE.INI	Same directory as Scanner

Scanner Generator Errors

The most usual problems encountered when the Scanner is generated result in an error message:

ERROR {value} Generating Scanner

The causes can usually be identified as follows:

- The path defined for the executable Scanner executable file does not exist.
- The Scanner file already exists and is currently being used by another application.
- The file name chosen for the Scanner executable file is invalid (that is, does not follow the MS DOS file naming conventions and may include illegal characters).
- Some virus protection software may prevent the Scanner Generator from creating and writing to Scanner executable files.

To Resolve the Problem

- Try generating the Scanner using the default settings, path and file name.
- If the previous step fails, try again selecting a file name which you have checked does not exist.

Hardware Scanning Errors

If a hardware scanning error occurs, the screen will appear to stop responding, or ‘hang’ during hardware scanning.

Note the test which is failing. This will be displayed in the Log page.

The Scanner provides several command line parameter switches for disabling specific hardware detection tests. Use these command line parameter switches to disable the specific test which has failed.

Note: Typing /? following the Scanner file name at the command line displays a list of the available parameter switches, for example, `ScanW32 /?` (or -? for UNIX Scanners)

To use a parameter switch to disable a specific hardware test, enter it on the command line after the Scanner file name when the Scanner is started.

For example:

```
ScanW32 -60
scanlnx -excl:60
```

Asset Data Entry Errors

The following messages may be displayed during entry of asset information using the asset questionnaire:

The Value ‘x’ Is Not a Number Between <value> and <value>

This message is displayed if an illegal value is entered for a numeric field specifying a minimum value, a maximum value or a numeric range. Check that the value entered falls within that valid range defined for the field.

Entry Required at Line <field identifier>

This message is displayed in DOS and OS/2 Scanners for fields that are required to be completed by the user and an entry has not been made. It occurs when the user attempts to bypass the field to make the next entry. The Scanner cannot continue until a legal entry has been made.

In the Windows Scanners, the same message is displayed when the asset timer expires.

Input Does Not Conform to Picture <field identifier>

This message is displayed on formatted fields, where the data input does not conform to the input picture defined for the field. Check the format for the data entered for the field.

Software Scanning Errors

The following errors may occur during the examination of files and collection of software information.

Out of Swap Space – Cannot Store More Files in Scan File

This message is displayed if there is not enough room on the hard disk drive or in EMS for storing a file that has been marked for collection in the Scanner Generator.

Very Low on Memory – Scan May Fail Any Time

This message is applicable to the DOS Scanner only. This message is displayed if the memory is too low for the Scanner to continue scanning the software files. If the scan fails, try to free more DOS memory then re-run the Scanner.

Could Not read File <file name> - File Not Saved

This message is displayed if a file marked for collection in the Scanner Generator cannot be stored. Check to see if the file is being locked (used) by another process.

Scan File Saving Errors

The following messages may be displayed when the Scanner tries to save a scan file:

Error {value} Saving Local Scan File

There may be insufficient space on the local drive that the Scanner is attempting to save the scan file to. Check the available space on the local disk drive.

Another cause of this error message appearing might be that sufficient privileges do not exist to write the file or the drive cannot be accessed.

Error {value} Saving Offsite Scan File

There may be insufficient space on the offsite drive (for example the floppy disk or network drive) when the Scanner is attempting to save the scan file. Check the available space on the offsite disk drive.

The Scanner Cannot Create the Scan File in A:\<filename>.

Please correct the error, or specify a new drive. The Scanner will then attempt to save to the root of the drive.

This message is displayed, if there is not enough room for saving an offsite scan file, either because the disk selected to save the scan file to, cannot be accessed, or there is insufficient space available.

- Check the Use New Drive option to specify a new drive to save to.
- Select the new drive letter from the drop-down list.

Additional Errors

Additional errors the user may encounter running the Scanner include:

- Large Directory Found While Memory Low. Directory Split
- Corrupt Disk: 775 Error
- Not Enough Temp Space
- Hardware Security Partition Table Entry Problems
- Compression on Netware Servers
- Slow Scanning
- Virus Warning

Large Directory Found While Memory Low. Directory Split

This message is a notification. It is displayed when the Scanner reaches a directory that contains more than 16100 files. When this happens, the directory is “split” into two or more logical ones, although this is normally hidden from the user.

Corrupt Disk: 775 Error

If Windows is not shut down properly, the cached buffers may not be written to the disk, resulting in corrupt disk structures.

Run chkdsk or similar to check for corrupt sectors.

Not Enough Temp Space

Check that the TEMP variable points to a valid directory with enough disk space available. If it is missing or points to an incorrect directory, set it up accordingly (for example: SET TEMP=C:\TEMP).

Hardware Security Partition Table Entry Problems

It is necessary to process inventory disks with identification. There may be problems with partition table detection.

Use the –62 command line switch to circumvent the partition table scan test.

Compression on Netware Servers

All signatures should be off or override.ini must be set to ignore all files. This ensures that files are not opened and stored files are not collected. Netware compression is not dynamic such as NTFS compression in Windows NT/2000/XP/2003. Running Scanners could have detrimental effects in Netware Servers if compression is being used. This is because to signature a file, the file must be decompressed and then opened by the Scanner. Netware will not recompress the file, thus a capacity problem could result if the compressed volume is greater than the actual disk space available.

Slow Scanning

This may be due to real-time antivirus software being run. Any file that is opened will be checked for virus infection. Although this can be tedious, it is not advisable to disable the antivirus software for the reasons discussed in the next section.

Virus Warning

Because the Scanner opens files on the computer, if there is real-time antivirus software in operation, it may detect a virus being present in a file. Depending on the virus product being used, they will have an action defined to deal with the virus. Some will try to deal with the problem and immediately disinfect the file. Others will try to move the infected file to a quarantine directory and rename its file extension.

In this case, the quarantine directory may be scanned by the Scanner later during its scan.

To prevent this happening, use the override.ini file with *.vir (where .vir is a typical quarantine file extension). Check the specific product to find the extension for this type of file.

Enterprise Discovery and Compaq Machines

On some Compaq machines, a BIOS problem would sometimes cause the Scanner to fail or crash the machine when run.

This version of Enterprise Discovery implements a workaround where the Scanner refrains from running those parts of the Compaq code that have been shown to be problematic.

In this case the Compaq Asset tag may not be detected.

To detect the Compaq Asset Tag, the Scanner first locates and calls a standard 32-bit interface to the Compaq Intelligent Manageability layer. If this call fails or is not implemented, the Scanner has the ability to fall back on 16-bit code stored in the Compaq BIOS; this is the code that is fragile in some BIOSes.

Tests show that newer Compaq BIOSes correct the problem. On machines where the BIOS has been upgraded, the code works correctly.

Because it is difficult to ensure that all BIOSes have been upgraded, the default behavior of the Scanner has been changed to only fall back on the 16-bit code on machines not running Server operating systems.

A command line switch (COMPAQ16) has been introduced, allowing this default behavior to be overridden:

Parameter Switch	Description
Scanw32 /COMPAQ16=0	When this switch is used, the Scanner will never use the 16-bit BIOS code.
Scanw32 /COMPAQ16=1	In this case, the 16-bit BIOS code will only be called on systems where the 32-bit interface is not present or does not work, and only if the machine being scanned is not running Windows NT/2000/XP/2003 Server or Advanced Server. This is the default.
Scanw32 /COMPAQ16=2	In this case, the Scanner will always fall back on the 16-bit code as necessary to retrieve the Compaq Asset Tag.

Using the Scanners for Manual Inventories

Enterprise Discovery Scanners can be generated as stand-alone executables that can be run in a number of ways.

Once you have configured and generated the correct type of Scanners for your computer population, the next issue you will face is how to execute them.

Walkround Inventory

When starting your inventory project it may be necessary to initially conduct a walkround inventory. There may be machines that are not connected to the network, or there may be a closet full of older or broken machines which may only be discovered by physically finding then.

All of these machines need to be accounted for as part of a sound asset management program. Additionally, there is user asset information such as user first name, last name and location which must initially be manually entered.

With a walkround inventory, you can execute the Scanner from a floppy disk or connect to a network share and run it from there.

Using a Distribution Tool Such As SMS

The advantage of a distribution tool, such as Microsoft's SMS is, it allows an administrator to determine at their discretion when an inventory needs to take place. An administrator has the power from their System Management Console to push a command onto a remote machine at their will. This could include the execution of a Scanner. The disadvantage of a product such as SMS is that an agent must be present on each desktop that the administrator would like to control. this requires time and expertise. Enterprise Discovery comes with a the capability to produce MIF files. These basically allow all SMS clients who are scanned to have their scan files converted to a standard MIF format which SMS can store, read and process.

Command Line Execution

Although the options for the Scanner are normally set using the Scanner Generator, it may be necessary to change some settings to allow better operation on some machines being scanned. This may be to accommodate a 'quirky' machine or to simply change the name given to the scan file. The advantage of running a Scanner from the command line is that there are numerous switches available to override options configured in the Scanner Generator. In addition, new features become available such as the option to run a scan on a scheduled basis.

For more information about command line options for the Scanners, see Command Line Options and Switches on page 30.

Login Scripts

Important: Peregrine cannot provide support in the writing of login scripts. Sample login scripts have been included in this section to provide you with a feel for what can be achieved.

There are several sample scripts provided in Example Network Inventory Scripts on page 92 to assist you. This section provides a more detailed description of these scripts. The network clients supported are Windows 9x (95, 98, Millennium) and Windows NT (NT 4, 2000, XP).

Using these scripts to set up an inventory requires that you to do the following:

- Create the directories that the scripts and Scanner will write to and read from. All of these directories have the same parent.
  - Scanners: this is the location where the Scanner executables reside.
  - Scans: this is the collection point.
  - Refill: This is the offsite refill location for the Scanner. Users only require read permission on this directory.
  - Log: An optional location that the scripts will write any system messages that occur during scanning to. This is useful for troubleshooting any problems.
  - Excluded: If a scan does not succeed on a client, the script will write a file in the form “<ASSETTAG>.fal” to this location. This will prevent the

Scanner from running the next time, and will also indicate that something went wrong during scanning.

- Invoke the script "audit.bat" from a workstation's login script.

The Directories

This example assumes that the directories are organized with user permissions in mind.

The five directories used during the network inventory in this example are:

- Scanners, containing all batch and executable files that are run during the network inventory. A trigger to launch the main script, Audit.bat, is typically be placed with the NT domain login script. Users need Read and Execute permissions over files inside this directory.
- Scans Directory that the Scanner is configured to save back to. Users need Write permissions to this directory.
- Refill Directory that the Scanner is configured to refill from. Users need Read permission to this directory.
- Log Contains an optional Audit.log file that maintains a log of any system messages that occur during the inventory. This file can be useful for tracking any errors that occur during the inventory process. Users need Write permissions to this directory.
- Excluded. This is used if an inventory has not been successful on a client. In some cases, it may be desirable to exclude that particular client machine from being repeatedly inventoried, until the problem has been located and corrected. A .fal file is created (a flag file for a given asset number). The specified client is excluded from being inventoried until Technical Support clear the fault. It may be desirable to not create flags in cases of 'User Termination'. This directory is optional but is useful for system administration purposes.

Overview of the Process

The following steps are carried out in this example:

- The remote access clients are detected
- Audit.bat is started
- The client operating system is detected

- Under Windows 95/ME environment space needs to be allocated, so a .pif file is created. This is so that there is enough space to create environment variables. The 95audit.bat file is then called.
- In Windows NT/200x/XP, a local environment is created inside the NTaudit.cmd file, therefore a .pif is not needed. The NTaudit.cmd file is called.
- The main scripts (95audit.bat for Windows 95/98/Me and NTaudit.cmd for Windows NT/200x/XP), carry out the following:
    - Set up variable locations being used and the frequency of inventory.
    - Activate the workstation identifier. This example uses the computer name as the identifier.
    - Check for flag files for any excluded machines.
    - Execute the Scanner with the scandays switch which allows the Scanner to check the age of the fingerprint. If the age of the scan file to be refilled from is older that the scan date, then you will get a full scan, otherwise the Scanner terminates.
    - Check the error codes
    - Check that the scan file has been created
    - Log various system messages.

The Scripts

The high level of what actually happens from login script to audit script termination is as follows:

- The user logs in, and their login script is executed.
- The login script calls “audit.bat”.
- “audit.bat” detects the operating system that is running and either launches the appropriate audit script (either “95audit.bat” or “NTaudit.bat”) or exits after displaying a brief “Operating system not supported” message.
- The audit script checks that no applicable “.fal” file appears in the “excluded” directory and launches the Scanner.
- The Scanner executes, and exits when it is finished.

The Audit.bat File

The Audit.bat files uses an executable called IDDKind.exe (which is located in the \Peregrine\Enterprise Discovery\2.0.0\Scanner Generator directory) to detect the client operating system. It returns the following return codes (error levels) corresponding to the client operating system detected:

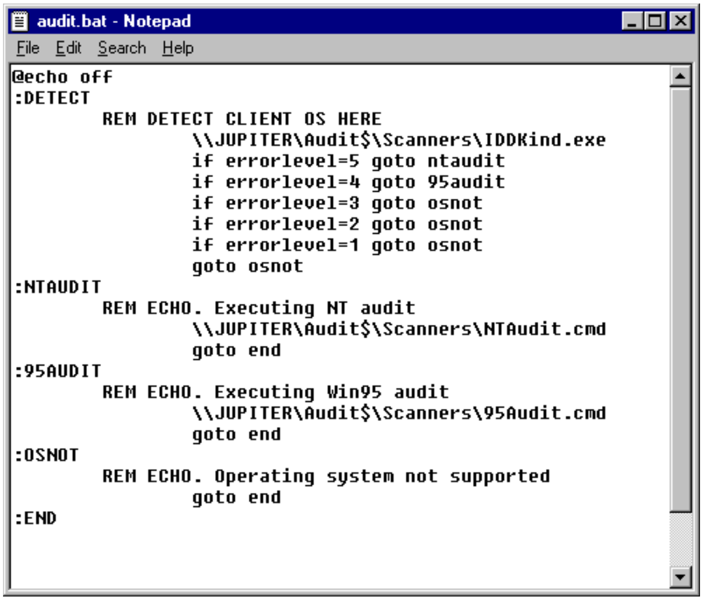
- 1. Dos
- 2. Win 3.x
- 3. OS/2
- 4. Win95/98/ME
- 5. WinNT/2000/XP

If the client is running Windows NT, it starts a Windows NT inventory by executing the NTaudit.cmd file.

If the client is running Windows 95/98/ME, it starts a Windows 95 inventory by executing the 95audit.pif file.

If the client operating system is not supported the following message is displayed:

```
Operating system not supported
```



```
@echo off
:DETECT
        REM DETECT CLIENT OS HERE
                \\JUPITER\Audit$\Scanners\IDDKind.exe
                if errorlevel=5 goto ntaudit
                if errorlevel=4 goto 95audit
                if errorlevel=3 goto osnot
                if errorlevel=2 goto osnot
                if errorlevel=1 goto osnot
                goto osnot
:NTAUDIT
        REM ECHO. Executing NT audit
                \\JUPITER\Audit$\Scanners\NTAudit.cmd
                goto end
:95AUDIT
        REM ECHO. Executing Win95 audit
                \\JUPITER\Audit$\Scanners\95Audit.cmd
                goto end
:OSNOT
        REM ECHO. Operating system not supported
                goto end
:END
```

Refer to Example Network Inventory Scripts on page 92 for an example Audit.bat file.

The NTaudit.cmd File

- The first part of the NTaudit.cmd file sets environments locally.
- Windows 95 does not allow environments to be set locally.
- A Drive Variable is set to \\Jupiter\audit$

  This is the collection point.

- Hidden (that is, $ shares) are not visible to users when they browse the network and therefore are a little more secure.
- A variable is set for the Scanner repeat frequency.
- The computer name is set as the unique identifier.
- A check is made to see if any file flags have been raised for clients to be excluded from the inventory.
- A check is made to see if the Infrtool.ini file exists. If not, a new one is created.
- The Scanner is launched and a message is displayed on-screen.
- A check is made for any errors that may have occurred during the inventory. The errors are as follows:

Error Level	Description
ErrorLevel 6 is PLUGERR	The Scanner was terminated by a plug-in. An error message is displayed and an entry is added to the log file and the .fal flag file.
ErrorLevel 5 is TOOEARLY	The scan is not required yet. That is, it is earlier than the scandays variable set earlier in the file. A message is displayed.
ErrorLevel 4 is FATAL	A fatal exception occurred. An error message is displayed and an entry is added to the log file and the .fal flag file.
ErrorLevel 2 is USERBAD	A user terminated the Scanner. An error message is displayed and an entry is added to the log file and the .fal flag file.
ErrorLevel 1 is EXECPT	An unspecified exception occurred. An error message is displayed and an entry is added to the log file and the .fal flag file.

ErrorLevel 3 is not used. This is reserved for use when the Scanner has been executed with the /? (help) switch. This situation would never arise in an automated Scanner execution. If no errors are reported, the error level is set to 0.

- The scan is given passed status if an offsite scan file exists and the ErrorLevel was 0. An entry is written to the log.
- The scan is given failed status if an offsite scan file does not exist or an ErrorLevel other than 0 was detected. An entry is written to the log and the .fal flag file is created for that asset number.
- The environment variables (ASSETNO and EDDRV) are reset if necessary.

The 95audit.bat File

- A Drive Variable is set to \\Jupiter\audit$

  This is the collection point.

  Hidden (that is, $ shares) are not visible to users when they browse the network and therefore are a little more secure.

- It sets the asset number as an environment variable. If the asset number (taken from asset.bat) does not exist, a scan is triggered.
- A check is made to see if any file flags have been raised for clients to be excluded from the inventory.
- A check is made to see if the Infrtool.ini file exists. If not, a new one is created.
- The Scanner is launched and a message is displayed on-screen. In Windows 95, the default behavior for launching batch jobs is to have them running in the background. This differs between Windows NT/200x/XP and Windows 95. In Windows 95, it is necessary to use the /w switch to instruct it to wait.

- A check is made for any errors that may have occurred during the inventory. The errors are as follows:

Error Level	Description
ErrorLevel 6 is PLUGERR	The Scanner was terminated by a plug-in. An error message is displayed and an entry is added to the log file and the .fal flag file.
ErrorLevel 5 is TOOEARLY	The scan is not required yet. That is, it is earlier than the scandays variable set earlier in the file. A message is displayed.
ErrorLevel 4 is FATAL	A fatal exception occurred. An error message is displayed and an entry is added to the log file and the .fal flag file.
ErrorLevel 2 is USERBAD	A user terminated the Scanner. An error message is displayed and an entry is added to the log file and the .fal flag file.
ErrorLevel 1 is EXECPT	An unspecified exception occurred. An error message is displayed and an entry is added to the log file and the .fal flag file.

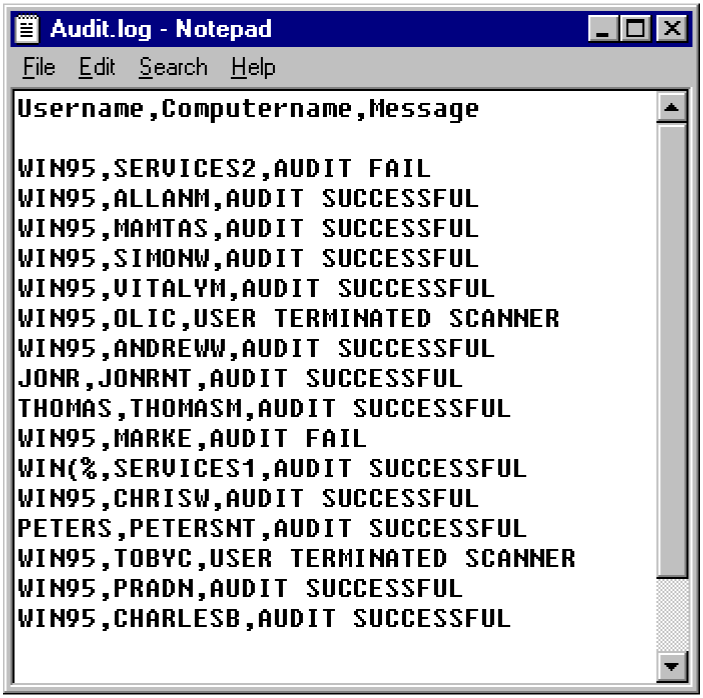
ErrorLevel 3 is not used. This is reserved for use when the Scanner has been executed with the /? (help) switch. This situation would never arise in an automated Scanner execution. If no errors are reported, the error level is set to 0.

- The scan is given passed status if an offsite scan file exists and the ErrorLevel was 0. An entry is written to the log.
- The scan is given failed status if an offsite scan file does not exist or an ErrorLevel other than 0 was detected. An entry is written to the log and the .fal flag file is created for that asset number.
- The environment variables (ASSETNO and EDDRV) are reset if necessary.

Refer to Example Network Inventory Scripts on page 92 for an example 95audit.bat file.

The Inventory Log

Creating a log during a network inventory can be a useful tool in controlling the resulting data. It is a simple way to find out if someone has aborted an attempted scan or to locate failed scans. The inventory log can be created in Comma Separated Variable (CSV) format so it can easily be imported into another application such as Microsoft Excel to sort, filter and analyze the logged data.



```
Username,Computername,Message

WIN95,SERVICES2,AUDIT FAIL
WIN95,ALLANM,AUDIT SUCCESSFUL
WIN95,MAMTAS,AUDIT SUCCESSFUL
WIN95,SIMONW,AUDIT SUCCESSFUL
WIN95,VITALYM,AUDIT SUCCESSFUL
WIN95,OLIC,USER TERMINATED SCANNER
WIN95,ANDREWW,AUDIT SUCCESSFUL
JONR,JONRNT,AUDIT SUCCESSFUL
THOMAS,THOMASM,AUDIT SUCCESSFUL
WIN95,MARKE,AUDIT FAIL
WIN(%,SERVICES1,AUDIT SUCCESSFUL
WIN95,CHRISW,AUDIT SUCCESSFUL
PETERS,PETERSNT,AUDIT SUCCESSFUL
WIN95,TOBYC,USER TERMINATED SCANNER
WIN95,PRADN,AUDIT SUCCESSFUL
WIN95,CHARLESB,AUDIT SUCCESSFUL
```

Sample Code

The following code shows how to add a line containing “Audit Successful” to the log file. The code is placed after the “SCANPASS” label, assuming that the full script contains a GOTO SCANPASS when the scan is successful.

```
:SCANPASS
REM OFFSITE SCANFILE EXISTS, AUDIT SUCCESSFUL, WRITING TO LOG
ECHO.WIN95,%Computername%,Audit Successful>> %EDDRV%\AUDIT.LOG
```

Server Inventories

Scanning servers is an integral part of an overall inventory. Because most important company data is kept on a server it is important to know the process and some of the considerations involved in server scanning. This sections takes a close look at two types of server inventories:

- Windows NT/2000 servers
- Novell Netware servers

Scanning NT Servers

An NT server can be inventoried in very much the same way as an NT workstation. The inventory can take place directly on the server with the Scanner executed from the Windows interface. A Enterprise Discovery Scanner can be run while an NT server is serving users. To avoid a performance impact during the scan, use Scanner Generator to configure the server Scanners to run at low priority.

It can be difficult to locate asset information (for example, serial number, make) from the system units themselves because the location of servers in server rooms makes access to items such as serial numbers difficult. To make asset collection easier, check whether a server list with the asset details already exists. Many network teams compile this type of information for server maintenance contracts (but normally do not bother for workstations).

Identify when the server can be accessed. Usually a server is only available out of normal working hours. Always ensure that there has been a recent system backup before carrying out a scan of the server. This will ease the disaster recovery process if the server does develop a fault either while scanning or while performing a startup/shutdown.

When electronically scanning NT servers use a 'local administrator' account. With administrator access rights, the Scanner will be able to collect a detailed inventory of the hardware on the machine.

If administrator level access is not available then use a 'backup operator' account instead. The Scanner may be unable to collect certain pieces of information in this case.

Avoid using a standard user account. The Scanner will be able to collect only a minimum of hardware information with user rights.

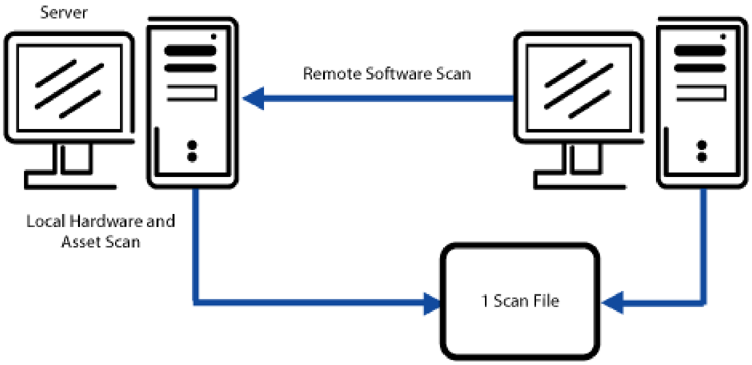
Scanning Novell Netware Servers

Scanning a Netware server is a two-stage process:

- Scanning hardware from DOS while it is down.
- Scanning software while the server is running.

Because of the Netware architecture, native processes running on the OS are able to bring down the server easier than in other operating systems that are designed to also function as application servers. The two stages are necessary because Enterprise Discovery does not include a native Scanner that can run as a process on a running Netware server.

Perform the hardware scan first, followed by a remote scan using the Remote Scanner. The Remote Scanner is able to read the hardware scan collected from the DOS scan, and incorporate it into a single scan file containing the hardware data as well as the remotely collected software data.



DOS Hardware Scan

This stage must be performed after the Netware server has been brought down and exited to a DOS environment. When bringing down the server, ensure that a recent system backup is available, any active users have been advised of the shutdown, and that any active NLM modules can be correctly closed.

If shutting down the server is not an option, it is still possible to store information about your Netware servers; refer to the next section.

When DOS is available, a Enterprise Discovery DOS Scanner can be run on the server. This Scanner can be used to capture asset details and hardware information.

Because the DOS partition is often limited to a size which has just sufficient space to store the support files needed to load Netware, there may not be enough space for the Scanner to create a temporary file. This needed to hold the scan information and may result in a 770 error. If this happens, set the TEMP environment variable to point to the inventory floppy disk, “SET TEMP=A:\”.

Another problem is that Autoexec.bat on the server may try and load Netware immediately. It is necessary to break into this boot sequence to get to DOS to run the Scanner. If this is not possible, then boot from a bootable floppy disk with the TEMP variable pointing to the floppy disk.

While the server is in DOS and the scan is in progress, the server will not be available to users. However, because the scan is limited to asset information, hardware test and any small DOS partitions, the time required to scan is very short. Usually the time consuming element is ensuring that all users are logged out before the server is taken down.

Because it is necessary to shut down a Novell server before a hardware inventory, it is good practice to try and restart the server before attempting an inventory. Because servers are rarely turned off, there may be problems reloading the server and its services. If a problem is found at this point, the inventory process can be eliminated as the source of the problem.

Scanning Software Using the Remote Scanner

Because Enterprise Discovery does not include a native Netware Scanner, it is not possible to scan the Netware volumes from the server itself. Fortunately, the Remote Scanner provides a good solution that allows the relevant information to be captured without risking a problem that could bring down the server.

The Remote Scanner can scan all of the network shares published by the server, and integrate the hardware scan collected with the DOS Scanner earlier. Because the hardware on your Netware server is unlikely to change, running the Remote Scanner regularly is sufficient to maintain an up-to-date inventory of your Netware servers.

If the server could not be brought down or a hardware scan of the machine is unavailable for other reasons, the Remote Scanner allows the hardware data to be typed in, similar to the asset questionnaire. The supplied hardware information is then available for use in other Enterprise Discovery tools, just as if it had been detected by the Scanner itself.

For more information refer to The Remote Scanner on page 64.

A Potential Problem Running a Scan on Netware 4 or 5

To scan a file, the Scanner opens it. Netware 4 or 5 decompresses files when they are opened, if compression has been applied. This means that the disk may fill rapidly and slow down the server when a software scan is performed. Generally, printing is the first service impacted. Netware compression is not dynamic, so uncompressed files will not recompress after being read.

To overcome this problem, do one of the following:

- Create a file in the same directory as the Scanner. Name the file override.ini. This is an ini file that allows Scanner Generator settings for the software scan to be changed without regenerating the Scanner. Add the following lines to the file:

```
[exclude]
file=*.*
```

- This excludes all files from being opened, although file details may still be recorded in the scan file. Because the file is not opened or a signature calculated, this speeds up the scan (although the data collected is less complete than without the exclusion).
- Create a Scanner version with Ignore file signatures for specific files.

These are a number of issues to consider with these approaches. No header information is captured when files are not opened by the Scanner, which makes file identification difficult. Any stored file included in the exclusion will not be opened for the contents to be saved as part of the scan file. No plug-in data will be captured, if the file is not opened.

This technique can also be used on the rare occasion that the software scan fails on a specific file. This file name can be included in the override.ini file and the scan re-run. Again the file name will be recorded but no detail will be stored. This facility can be used with NT servers but may not be necessary, as the file decompression and recompression is dynamic.

Example Network Inventory Scripts

The scripts shown in this appendix are examples only. They are relevant to Windows 95/98/Me and Windows NT/2000/XP.

Note: You will need to write your own scripts that are customized for use with your network.
The scripts assume that the name of the server used is Jupiter.

Audit.bat

```
Audit.bat
@echo off
:DETECT
       REM DETECT CLIENT OS HERE
```

```
                \\JUPITER\audit$\scanners\IDDkind.exe
                if errorlevel=5 goto ntaudit
                if errorlevel=4 goto 95audit
                if errorlevel=3 goto osnot
                if errorlevel=2 goto osnot
                if errorlevel=1 goto osnot
                goto OSnot
:NTAUDIT
        rem echo. executing NT audit
                \\JUPITER\AUDIT$\SCANNERS\NTAUDIT.CMD
                goto end
:95audit
        rem echo. executing Win95 audit
                START \\JUPITER\audit$\scanners\95AUDIT.PIF
                goto end
:osnot
        rem echo. operating system not supported
        goto end
:end
```

NTaudit.cmd

```
@ECHO OFF
color 17
CLS
SETLOCAL
REM Sample login script for network inventory using Enterprise
Discovery v2.0.0
REM Script for use in Windows NT/2000/XP

:EDDDRV
REM Set EDDDRV variable to make script revisions easier
SET EDDDRV=\\JUPITER\AUDIT$

:DAYS
REM Set DAYS variable for scanner repeat frequency
SET DAYS=14

:FLGCHK
REM Set the asset number as an environment variable before
checking for flag files
SET ASSETNO=%COMPUTERNAME%

:FALCHK
REM Check dummy Fail file for asset number %ASSETNO% on%EDDDRV%
IF NOT EXIST %EDDDRV%\EXCLUDED\%ASSETNO%.FAL GOTO SCANRUN
ECHO Fail flag exists - terminate
GOTO END
```

```
:SCANRUN
REM Now launching the scanner
CLS
ECHO.
ECHO. ***********************************************************
ECHO. * This Windows NT PC may now be scanned with -            *
ECHO. * Enterprise Discovery V8.0 (Win32 Scanner)      *
ECHO. * Please be patient, if required the scan will only take  *
ECHO. *  a few minutes, and will then repeat after %days% days  *
ECHO. ***********************************************************
ECHO.
ECHO. +++  Thank you for your co-operation %USERNAME%  +++
ECHO.
%EDDDRV%\Scanners\ScanW32.Exe -SCANDAYS:%DAYS%
GOTO ERRCHECK

:ERRCHECK
REM Check the scanner return code (available in the errorlevel)
IF ERRORLEVEL 6 GOTO PLUGERR
IF ERRORLEVEL 5 GOTO TOOEARLY
IF ERRORLEVEL 4 GOTO FATAL
IF ERRORLEVEL 2 GOTO USERBAD
IF ERRORLEVEL 1 GOTO EXCEPT
GOTO SCANCHK

:PLUGERR
REM The scanner returned with an internal plug-in error
CLS
ECHO.
ECHO. ***********************************************************
ECHO. *       !!!  SCANNER TERMINATED BY PLUG-IN  !!!           *
ECHO. *            =============================                *
ECHO. *                                                         *
ECHO. *      PLEASE CONTACT IT SUPPORT AS SOON AS POSSIBLE      *
ECHO. *                                                         *
ECHO. ***********************************************************

REM Add failure to audit log and to the failure file for this
asset
ECHO.%USERNAME%,%ASSETNO%,PLUGIN FAILURE>> %EDDDRV%\LOG\AUDIT.LOG
ECHO.PLUGIN FAILURE>>%EDDDRV%\EXCLUDED\%ASSETNO%.FAL
GOTO END

:TOOEARLY
REM The scanner does not yet need to be run as %DAYS% have not
elapsed since the last scan
CLS
ECHO.
ECHO. ***********************************************************
ECHO. *                  SCAN NOT REQUIRED                      *
ECHO. *                  =================                      *
ECHO. *                                                         *
ECHO. * Result has not yet reached threshold of %days% days    *
ECHO. *                                                         *
ECHO. ***********************************************************
GOTO END

:FATAL
REM The scanner has terminated with a fatal error.
REM Add this information to the log and failure file
ECHO.%USERNAME%,%ASSETNO%,FATAL ERROR IN SCANNER>>
%EDDDRV%\LOG\AUDIT.LOG
ECHO.FATAL ERROR IN SCANNER>>%EDDDRV%\EXCLUDED\%ASSETNO%.FAL
GOTO END

:USERBAD
REM The user terminated the scan without saving a scan file
ECHO.%USERNAME%,%ASSETNO%,USER TERMINATED SCANNER>>
%EDDDRV%\LOG\AUDIT.LOG
ECHO.USER TERMINATED SCANNER>>%EDDDRV%\EXCLUDED\%ASSETNO%.FAL
GOTO END

:EXCEPT
REM The scanner has terminated with an exception.
REM Add this information to the log and failure file
ECHO.%USERNAME%,%ASSETNO%,EXCEPTION IN SCANNER
>>%EDDDRV%\LOG\AUDIT.LOG
ECHO.EXCEPTION IN SCANNER>>%EDDDRV%\EXCLUDED\%ASSETNO%.FAL
GOTO END

:SCANCHK
REM If no errorlevel is reported (errorlevel 0), check the result
ECHO. Checking results...
IF NOT EXIST %EDDDRV%\FSFS\%ASSETNO%.FSF GOTO INICHECK
GOTO SCANPASS

:SCANPASS
REM The scan was successful; the offsite scan file exists. Add
this to the log.
ECHO.%USERNAME%,%ASSETNO%,Audit Successful>>
%EDDDRV%\LOG\AUDIT.LOG
GOTO END

:SCANFAIL
REM The scan was not successful: no offsite scan file exists. Add
this to the log.
ECHO.%USERNAME%,%ASSETNO%,UNEXPECTED FAILURE>>
%EDDDRV%\LOG\AUDIT.LOG
```

```
ECHO.UNEXPECTED FAILURE>>%EDDDRV%\EXCLUDED\%ASSETNO%.FAL
GOTO END

:END
REM Reset environment variables if necessary (ASSETNO and EDDDRV)
REM Change the ASSET.BAT file to be Hidden and Read/Only.
ATTRIB C:\ASSET.BAT +H +R
ENDLOCAL
```

95audit.bat

```
@ECHO OFF
REM Sample login script for network inventory using Enterprise
Discovery v8.0
REM Script for use in Windows 95/98/Me

:EDDDRV
REM Set EDDDRV variable to make script revisions easier
SET EDDDRV=\\JUPITER\AUDIT$

:DAYS
REM Set DAYS variable for scanner repeat frequency
SET DAYS=14

:FLGCHK
REM Set the asset number as an environment variable before
checking for flag files
REM Get the asset number from ASSET.BAT. If that does not exist,
just run the scanner.
IF NOT EXIST C:\ASSET.BAT GOTO SCANRUN
CALL C:\ASSET.BAT

:FALCHK
REM Check dummy Fail file for asset number %ASSETNO% on%EDDDRV%
IF NOT EXIST %EDDDRV%\EXCLUDED\%ASSETNO%.FAL GOTO SCANRUN
ECHO Fail flag exists - terminate
GOTO END


:SCANRUN
REM Now launching the scanner
CLS
ECHO.
ECHO. ***********************************************************
ECHO. * This Windows 9x PC may now be scanned with -            *
ECHO. * Enterprise Discovery V8.0 (Win32 Scanner)      *
ECHO. * Please be patient, if required the scan will only take  *
ECHO. *  a few minutes, and will then repeat after %days% days  *
ECHO. ***********************************************************
ECHO.
ECHO. +++  Thank you for your co-operation +++
ECHO.
        %WINDIR%\Command\Start /W %EDDDRV%\Scanners\ScanW32.Exe
-SCANDAYS:%DAYS%

REM Call Asset.Bat to make sure the asset tag is in the ASSETNO
environment variable
CALL C:\ASSET.BAT
GOTO ERRCHECK

:ERRCHECK
REM Check the scanner return code (available in the errorlevel)
IF ERRORLEVEL 6 GOTO PLUGERR
IF ERRORLEVEL 5 GOTO TOOEARLY
IF ERRORLEVEL 4 GOTO FATAL
IF ERRORLEVEL 2 GOTO USERBAD
IF ERRORLEVEL 1 GOTO EXCEPT
GOTO SCANCHK


:PLUGERR
REM The scanner returned with an internal plug-in error
CLS
ECHO.
ECHO. ***********************************************************
ECHO. *       !!!  SCANNER TERMINATED BY PLUG-IN  !!!           *
ECHO. *            =============================                *
ECHO. *                                                         *
ECHO. *      PLEASE CONTACT IT SUPPORT AS SOON AS POSSIBLE      *
ECHO. *                                                         *
ECHO. ***********************************************************

REM Add failure to audit log and to the failure file for this
asset
ECHO.WIN9X,%ASSETNO%,PLUGIN FAILURE>> %EDDDRV%\LOG\AUDIT.LOG
ECHO.PLUGIN FAILURE>>%EDDDRV%\EXCLUDED\%ASSETNO%.FAL
GOTO END

:TOOEARLY
REM The scanner does not yet need to be run as %DAYS% have not
elapsed since the last scan
CLS
ECHO.
ECHO. ***********************************************************
ECHO. *                  SCAN NOT REQUIRED                      *
ECHO. *                  =================                      *
ECHO. *                                                         *
ECHO. * Result has not yet reached threshold of %days% days     *
ECHO. *                                                         *
ECHO. ***********************************************************
GOTO END

:FATAL
REM The scanner has terminated with a fatal error.
REM Add this information to the log and failure file
ECHO.WIN9X,%ASSETNO%,FATAL ERROR IN SCANNER>>
%EDDDRV%\LOG\AUDIT.LOG
ECHO.FATAL ERROR IN SCANNER>>%EDDDRV%\EXCLUDED\%ASSETNO%.FAL
GOTO END

:USERBAD
REM The user terminated the scan without saving a scan file
ECHO.WIN9X,%ASSETNO%,USER TERMINATED SCANNER>>
%EDDDRV%\LOG\AUDIT.LOG
ECHO.USER TERMINATED SCANNER>>%EDDDRV%\EXCLUDED\%ASSETNO%.FAL
GOTO END

:EXCEPT
REM The scanner has terminated with an exception.
REM Add this information to the log and failure file
ECHO.WIN9X,%ASSETNO%,EXCEPTION IN SCANNER >>%EDDDRV%\LOG\AUDIT.LOG
ECHO.EXCEPTION IN SCANNER>>%EDDDRV%\EXCLUDED\%ASSETNO%.FAL
GOTO END

:SCANCHK
REM If no errorlevel is reported (errorlevel 0), check the result
ECHO. Checking results...
IF NOT EXIST %EDDDRV%\FSFS\%ASSETNO%.FSF GOTO SCANFAIL
GOTO SCANPASS

:SCANPASS
REM The scan was successful; the offsite scan file exists. Add
this to the log.
ECHO.WIN9X,%ASSETNO%,Audit Successful>> %EDDDRV%\LOG\AUDIT.LOG
GOTO END

:SCANFAIL
REM The scan was not successful: no offsite scan file exists. Add
this to the log.
ECHO.WIN9X,%ASSETNO%,UNEXPECTED FAILURE>> %EDDDRV%\LOG\AUDIT.LOG
ECHO.UNEXPECTED FAILURE>>%EDDDRV%\EXCLUDED\%ASSETNO%.FAL
GOTO END

:END
REM Reset environment variables if necessary (ASSETNO and EDDDRV)
SET ASSETNO=
SET EDDDRV=
SET DAYS=
REM Change the ASSET.BAT file to be Hidden and Read/Only.
ATTRIB C:\ASSET.BAT +H +R
```

5

CHAPTER

Scanner Plugin-SDK

The Enterprise Discovery Scanners provide an interface for “plug-in” modules. Organizations that require additional information to be collected during the scan, or wish to develop plug-ins for reselling, can use the information in this document to assist in writing a plug-in module.

The current plug-in interface allows two different kinds of data to be collected:

- Using the *Data File Recognition* method, information can be collected from each file the scanner accesses.
- Using the *Archive Processing* method, a plug-in can implement scanning of a new archive type, recognize a file as an archive, and send data (filename, size, CRC, etc.) on each file in the archive back to the scanner.

The possibilities for collecting data for individual files on a machine are limited only by imagination – with the appropriate plug-in, it would be possible to extract keywords from documents, highlight Year 2000 problems in spreadsheets or databases, scan for possible virus infections, etc.

The archive processing can be used to implement support for one or more of the less common archive types not natively supported by the Scanners, such as SQZ. In addition, it would be possible to implement support for disk image files, which also can be considered archive files as they are files that themselves contain directories and files. With the right code available, it is also feasible to use this feature to scan message databases or other file stores using this feature.

Implementation and Distribution

A plug-in consists of a number of dynamic link libraries, along with a configuration file that identifies the plug-in and the files that it comprises. Due to the fact that DOS has no support for DLLs, the DOS scanners do not support plug-ins.

Note: The plug-in is not supported by UNIX scanners.

A Scanner plug-in is made up of at least one but optionally several executable components:

- Scanner Generator DLL:

  This DLL is required if the plug-in needs advanced configuration not covered using the standard plug-in options of the Scanner Generator. If present, the DLL should display a dialog box displaying relevant options, initialize its controls from data obtained from the Scanner Generator, and send the modified configuration information back to the Scanner Generator.

- Scanner DLLs:

  This is the core of the plug-in, which implements the information gathering functionality of the plug-in. The DLL is initialized at scan-time with the configuration obtained from the Scanner Generator, and is then passed a reference to each file that the scanner examines. The DLL is free to do its own examination of the file, is allowed full read access to the file through scanner interface functions, and can store any information (or none) that it gathers into the generated fingerprint.
  For full platform-support, separate scanner DLLs for Win32, Win16 and OS/2 should be produced.

- Analysis DLL:

  Provides an interface to the plug-in data for those Enterprise Discovery tools parsing the scan result. The interface allows the plug-in to structure the data efficiently, while allowing the data to be displayed and queried like any other data collected by the Scanners. An analysis DLL is not required for plug-ins using the *Archive Recognition* method.

All of the DLLs in question, except for the Win16 and OS/2 Scanner DLLs, must be Win32 PE DLLs.

Each of the files must be given a file name of the format "pg*NNNTTT.EXT*", where

- *NNN* is a unique plug-in ID, which is a number in the range [100 <= ID <= 999]. The plug-in ID must be unique across all installed plug-ins; to obtain a unique ID for a new plug-in, please contact Peregrine Systems, Inc. Technical Support. All files that make up the plug-in must have the same ID in the filename.

Note: Compiled DLL files have an internally hard-coded library name, which is used by the operating system to determine the module handle of the DLL. The actual name of the DLL must be the same as its internal name.

Plug-in IDs below 100 are reserved for the use of Peregrine Systems, Inc.

- *TTT* is the type of the file; any names can be used. The convention used for Enterprise Discovery plug-ins is:

  cfg: Indicates that this is a Scanner Generator DLL.

  w16, w32 or os2: Identifies that this is a Scanner DLL, and further identifies the platform that this DLL is intended for. The platforms are Win32, Win16 and OS/2 respectively.

  anl: Indicates that this is an Analysis DLL.

  ini: Indicates that this is a plug-in configuration file.

- *EXT* is the filename extension.

Sample Distribution

A plug-in with a unique plug-in ID of 218, implementing a plug-in for collecting extra file information when using the Win32 and OS/2 scanners, and requires special setup in the Scanner Generator would consist of the following files:

- pg218cfg.dll Scanner Generator configuration DLL
- pg218w32.dll Scanner DLL for the Win32 scanner
- pg218os2.dll Scanner DLL for the OS/2 scanner
- pg218anl.dll Analysis DLL
- pg218ini.ini Plug-in configuration file used by the Scanner Generator

To make the plug-in available to the Scanner Generator these files need to be copied into the **Common\Plugins** subdirectory of the Enterprise Discovery program directory.

Using the Plug-in

When a plug-in is configured in the Scanner Generator, the Configuration DLL is invoked (using the interface discussed in the Technical Guide section of this document) when the user presses the Advanced button for the plug-in. If the configuration file indicates that the plug-in does not contain a Configuration DLL, the Advanced button will be grayed out.

When the Scanner Generator generates the scanner executables, all relevant Scanner DLLs are packaged inside the scanner executable itself and are extracted on demand. Even when using one or more plug-ins, the scanner consists of a single self-contained executable only.

When launching the Enterprise Discovery Viewer, all Analysis DLLs available are enumerated and initialized. Any scan files (FSFs or XSF) containing data collected using plug-ins are processed normally, using the relevant Analysis DLL to parse the plug-in data stored. Plug-in data for which an Analysis DLL cannot be found is ignored.

In the Enterprise Discovery Analysis Workbench, all Analysis DLLs are also enumerated, and the Options dialog allows the user to choose which plug-in generated data should be loaded, if any. As for the Viewer, all scans containing any of the plug-in data selected for loading is parsed using the Analysis DLLs and is available for detailed analysis. Plug-in data for which an Analysis DLL cannot be found is ignored.

Files Included with this SDK

This Software Development Kit includes several files demonstrating how to write a plug-in using either MS Visual Studio (C source code) or Borland Delphi (Pascal source code).

The source code included was developed and tested using Microsoft Visual Studio v5.0 and Borland Delphi v3.02.

The plug-in is given an ID of 13, and contains a Configuration DLL, a Scanner DLL for Win32 only, and an Analysis DLL. The plug-in, which uses the Data File Recognition method, extracts and stores the first 4 bytes of every file scanned.

Directory	File Name	Purpose
sdk\plugin\1.0	Plugin Interface.pdf	This document
sdk\plugin\1.0\c	fp_def.h	Header file defining Enterprise Discovery data type aliases
	fpplgint.h	Header file defining the Enterprise Discovery Plug-in Interface
	pg13cnst.h	Header file defining constants specific to this plug-in
sdk\plugin\1.0\c\pg013cfg	pg013cfg.h	Header file defining the exported functions of the Configuration DLL
	pg013cfg.cpp	C source code for the Configuration DLL
	pg013cfg.def	Linker definition file for the Configuration DLL
	pg013cfg.res	Resource file containing dialog design for the Configuration DLL
sdk\plugin\1.0\c\pg013w32	pg013w32.h	Header file defining the exported functions and Instance Data structure of the Scanner DLL

	pg013w32.cpp	C source code for the Scanner DLL
	pg013w32.def	Linker definition file for the Scanner DLL
sdk\plugin\1.0\c\pg013anl	pg013anl.h	Header file defining the exported functions and Instance Data structure of the Analysis DLL
	pg013anl.cpp	C source code for the Analysis DLL
	pg013anl.def	Linker definition file for the Analysis DLL
sdk\plugin\1.0\pascal	use32.pas	Unit redefining various Integer types based on target platform and compiler
	fpplgint.pas	Unit defining the Enterprise Discovery Plug-in Interface
sdk\plugin\1.0\pascal\pg013	pg13cnst.pas	Unit with constant definitions specific to this plug-in
	pg013cfg.dpr	Delphi source file for the Configuration DLL
	pg013cfg.r32	Resource file containing dialog design for the Configuration DLL
	pg013w32.dpr	Delphi source file for the Scanner DLL
	pg013anl.dpr	Delphi source file for the Analysis DLL
sdk\plugin\1.0\dist	pg013c.zip	Sample plug-in distribution archive, based on output from the C source code
	pg013pas.zip	Sample plug-in distribution archive, based on output from the Pascal source code

Configuration File Format

The plug-in configuration file describes the version information, components and default configuration of the plug-in. The file, which is used by the Scanner Generator, is a Windows INI file, with the following sections, of which only the Options section is mandatory:

- Options: Describes version information and specifies the default configuration of the plug-in.
- <*platform*>ScannerFiles: Indicates the name of the Scanner DLL and any data files required for the Scanner plug-in for the platform <*platform*>.
- CfgFiles: Indicates the name of the scanner generator DLL.
- AnalysisFiles: Indicates the name of the analysis DLL.

Options

The options section of the configuration file can contain the following key values:

- Name: The name of the plug-in.
- Description: A brief description of the plug-in.
- ID: The ID of the plug-in.
- LimitByName, LimitBySize: Default configuration information. These are boolean values. A ‘0’ indicates ‘false’, a ‘1’ indicates ‘true’.
- MinFileSize, MaxFileSize: Default configuration information. These values are only pertinent if ‘LimitBySize’ is set to true. These are numerical integer values, in bytes.
- IncludeFileMask, ExcludeFileMask: Default configuration information. These keys are only pertinent if ‘LimitByName’ is set to true. These values are unquoted, text values containing comma-delimited lists of file-masks.

<platform>ScannerFiles

These sections must contain at least one key:

```
<filenameN>= Description of file
```

Valid values for <*platform*> are Win16, Win32 and OS2. The filenames listed must start with the name of the Scanner DLL for the platform. Following this, the names and descriptions of any data files used by the Scanner DLL for the platform can be specified, if any are required.

CfgFiles

This section must contain at least one key:

```
<CfgFileName>= Description of Configuration DLL
```

If the Configuration DLL requires additional files, these should be listed in this section as well.

AnalysisFiles

This section must contain at least one key:

```
<AnalysisFileName> = Description of Analysis plug-in DLL
```

If the Analysis DLL requires additional files, these should be listed in this section as well.

Example

The content of the configuration file for the hypothetical plug-in mentioned above, pg218ini.ini, could be:

```
[Options]
Name=Sample
Description=Hypothetical Plug-in for Scanners
ID=218
LimitByName=1
IncludeFileMask=*.EXE;*.COM
ExcludeFileMask=WIN*.*
LimitBySize=1
MinFileSize=1000
MaxFileSize=100000

[Win32ScannerFiles]
pg218w32.dll=The hypothetical Win32 Scanner DLL

[Os2ScannerFiles]
pg218os2.dll=The hypothetical OS/2 Scanner DLL

[CfgFiles]
pg218cfg.dll=The hypothetical Configuration DLL
```

```
[AnalysisFiles]
pg218anl.dll=The hypothetical Analysis DLL
```

API Overview

The table below lists the names of all entry points relevant to plug-ins, where they are or should be defined, and how they can be used. Please refer to the Technical chapter for detailed information on each API, or refer to the source code files defining the interface (fp_def.h, fpPlgInt.h for C programmers and fpPlgInt.pas for Pascal programmers):

API Name	Defined in	Usage
PlgGetAPIVersion	Configuration DLL Scanner DLLs Analysis DLL	Called by Enterprise Discovery components to verify that the plug-in is compatible with the Enterprise Discovery software used.
PlgConfigure	Configuration DLL	Allows the DLL to call and interface to the Scanner Generator functions.
Int->OptSetValue	Scanner Generator	Internal Scanner Generator function, the address of which is passed to the Configuration DLL. Used to set the value of a plug-in specific option.
Int->OptGetValue	Scanner Generator Scanners Analysis DLL	Internal Scanner Generator, Scanner and Analysis function. Used to retrieve the value of a plug-in specific option.
PlgInit	Scanner DLLs	Called by the Scanner at start-up. During this call, the DLL is expected to initialize any required internal data structures, as well as return information about itself to the scanner.
PlgRecogniseDataFile	Scanner DLLs	Called by the Scanner (for Data File Recognition plug-ins) for each file the plug-in has been set up to process.

PlgIsArchive	Scanner DLLs	Called by the Scanner (for Archive Processing) for each file the plug-in has been set up to process.
PlgFindFileInArchive	Scanner DLLs	Called by the Scanner (for Archive Processing) several times for each file identified as a supported archive.
PlgStoreData	Scanner DLLs	Called by the Scanner when scanning is complete. During this call, the plug-in can call the `MemStore` Scanner function to store any data collected by the plug-in.
PlgDone	Scanner DLLs	Called by the Scanner prior to exiting. The plug-in should free all memory previously allocated.
ScanInt->FileSeek	Scanners	Seek to a given offset of a file.
ScanInt->FileRead	Scanners	Read a block of data from a file.
Int->MemGet	Scanners Analysis DLL	Can be used by the plug-in to dynamically allocate memory.
Int->MemFree	Scanners Analysis DLL	Frees memory allocated using `MemGet`
ScanInt->MemStore	Scanners	Store a block of data in the scan file.
PlgGetColumns	Analysis DLL	Called to retrieve a list of columns the plug-in defines.
PlgNewData	Analysis DLL	Called when a new scan file is about to be loaded, allowing the plug-in data block to be retrieved from the scan file.
PlgGetColumnData	Analysis DLL	Called to retrieve plug-in data for a specific column for a given file.
PlgFreeData	Analysis DLL	Called when all data from the current scan file has been retrieved; memory allocated can be freed.
AnalysisInt->MemLoad	Enterprise Discovery tool	Load a block of data previously saved by the Scanner DLL from the scan file.

Technical Reference

Returning API Version Information

Every DLL component of a plug-in must export a function similar to the following:

```
unsigned short FPCALLCONV PlgGetAPIVersion()
{
return PlgAPIMinorVer + (PlgAPIMajorVer << 8);
}
```

The various Enterprise Discovery applications will query the component in order to find out what version of the plug-in API it supports. In the current implementation, this function must return a major version of 1 and a minor version of 0.

Scanner Generator Interface – Configuration DLL

The Scanner Generator is able to store advanced options for a plug-in. The format of the advanced options data is completely arbitrary, and at the sole discretion of the author of the plug-in. The data is communicated as pairs of NULL terminated strings, one for the name of the option, and one for its value.

The Scanner Generator will only store an option if it actually contains data. The following statement has no effect:

```
_SGInt->OptSetValue("MyOption", "");
```

Similarly, if a non-existing option is queried, it will return NULL (no data).

The interface between the Configuration DLL and the Scanner Generator works as follows:

- The Scanner Generator invokes the Configuration DLL by calling its single exported function, declared as:

  ```
  void FPCALLCONV PlgConfigure(const TPlgSGInterface* _SGInt);
  ```

  The `_SGInt` parameter is a data structure containing pointers to the SG Interface functions. Refer to the file “`fpPlgInt.h`” for the full structure of this item.

- `PlgConfigure` should then display a dialogue box, initializing its controls with data obtained from the Scanner Generator via the function

  `_SGInt->OptGetValue`

  which is declared as:

  ```
  void (FPCALLCONV *OptGetValue)(const char* Name, char* Value);
  ```

- The user then configures the plug-in via this dialogue box. When the configuration is completed, the DLL can send the new configuration back to the Scanner Generator via the function:

  `_SGInt->OptSetValue`

  which is declared as:

  ```
  void (FPCALLCONV *OptSetValue)(const char* Name, char* Value);
  ```

  For plug-ins that feature a Configuration DLL, the user may not have used the Advanced button when configuring the plug-in in the Scanner Generator. In this case, no advanced configuration data is stored and the Scanner DLLs must be able to assume reasonable defaults in this case. Only settings differing from the default should be saved using the `_SGInt->OptSetValue` function.

Collecting Information on a File By File Basis (Data File Recognition)

Scanner Interface – Data File Recognition

The Scanner DLL will be given an opportunity to examine every file that the scanner examines. The DLL can extract data from these files, and store it in data structures internal to the DLL. For each file, the DLL passes a unique index back to the scanner. The scanner stores this index, along with the plug-in ID, with the file's data. At the end of the scanning process, the DLL will be given an opportunity to write the data that it has collected into the fingerprint file. The Scanner Generator treats this information as a blob, writing it 'as-is' into the scan file.

At analysis time, the information is read from the fingerprint and passed to the Analysis DLL as a blob, exactly as it was written to the scan file. When the plug-in data for a specific file is needed, the Analysis DLL is passed the indices that were generated by the Scanner DLL and is expected to extract the pertinent data from the blob and pass it back to the caller.

The interface between the scanner and the Scanner DLL works as follows:

- The scanner calls a function of the DLL, where any initialization required by the DLL can be performed:

  void FPCALLCONV PlgInit(const TPlgScannerInterface *ScanInt, r TPlgScanInfo Info)

  When this function is called, the DLL is also required to send return information to the scanner. This information is returned via the `Info` structure, defined as follows:

```
typedef struct _TplgScanInfo
{
long  Id;            /* Unique ID of the plug-in         */
long  Flags;         /* Bitmapped field                  */
char* Description;   /* Description of the plug-in       */
void* InstanceData;  /* Per-instance data of the plug-in */
long  Reserved[4];
} TPlgScanInfo;
```

  The plug-in ID is sent back via the `Id` member.

  The `Flags` member is a bitmapped value used to indicate the type of plug-in, that is, a Data File Recognition plug-in, or an Archive plug-in.

  A brief description of the plug-in is sent back via the `Description` member. This description will be displayed in the ‘Messages’ text box on the software page of the scanner.

  The `InstanceData` member is used to store a reference to any internal data that the DLL allocates. This `InstanceData` pointer is passed to every subsequent function call, so that the DLL always has access to its allocated data. Allocating data and storing the references to it in variables that are global to the DLL should be avoided, as this could lead to conflicts in a multi-threading environment, or when using a DLL instancing the Data segment only once.

  Along with the references to its own internal data, the Scanner DLL must ensure that `InstanceData` points to a structure that also stores a reference to ScanInt, as this pointer is not passed to any further functions. ScanInt is a structure that contains pointers to all of the internal Scanner functions necessary to seek and read in files, extract advanced configuration options and write information to the scan file. For the exact structure of this item, see the header file "fpPlgInt.h".

- For each file read by the scanner, matching the criteria specified in the Scanner Generator plug-in configuration (File size and name restrictions), it invokes the DLL's exported function

  ```
  FP_BYTE FPCALLCONV PlgRecogniseDataFile(void* InstData, const
  TPlgFileInfo *DataFile, TPlgFileData *Data, const void
  *_Buffer);
  ```

  The scanner DLL can now read the file using the functions provided by ScanInt. The file handle required by ScanInt->FileSeek and ScanInt->FileRead can be obtained from DataFile->Handle. Any information collected can be stored in the DLL's internal data structures, and an identifying index passed back via Data->DataID. Other useful information about the file can be found in the DataFile structure.

  If the plug-in has data to store for the file, the function should return 1. If not, it should return a value of 0, in which case the value of Data->DataID is ignored.

  The first 8192 bytes of the file have already been read by the scanner, and are cached in a buffer pointed to by the _Buffer parameter. When at all possible, this buffer should be used instead of the file reading functions, for example, for signature scanning. This could potentially save one read per scanned file, which amounts to thousands of reads over the course of the scan, a significant time saving.

- When the scan is complete, the Scanner will invoke the DLL's exported function

  ```
  void FPCALLCONV PlgStoreData(void *InstData);
  ```

  Using the function ScanInt->MemStore, the DLL can now write its data to the scan file. MemStore may be called more than once if necessary, for example, to store blocks larger than 64kb in a Win16 environment. If MemStore is called multiple times, the individual data blocks will be stored sequentially, right behind each other, so that the contiguity of the blob is preserved. It is advised that the data collected by the DLL be stored in as tight a format as possible, to minimize the size of the scan file.

- Finally, the scanner will invoke the function

  ```
  void FPCALLCONV PlgDone(void *InstData);
  ```

  During this function, the DLL should de-allocate any memory that it has allocated.

Analysis Interface – Data File Recognition

When Enterprise Discovery tools read scan files containing plug-in data, the Analysis DLLs are used to parse the data. On startup, the Analysis DLL is called to get information about the number and type of 'columns' required to display its data. This function is called only once for each Analysis DLL.

Each time a new scan file with plug-in data is read, the Analysis DLL is notified in order for it to initialize its data based on the data blob stored in the fingerprint by the plug-ins Scanner DLL. To do this, the Analysis DLL is supplied with the addresses of a set of entry points that allow the data block to be read from the scan file, plug-in options to be queried, etc.

As software data is read from the scan file, the Analysis DLL is called for each file of interest, passing the file index generated by the Scanner DLL along with a plug-in column identifier. When this occurs, the DLL should extract the pertinent data from its data structures and pass it back in the format specified. The only format currently supported by Enterprise Discovery is null-terminated string. Note that not all files or columns may be queried, and that the order in which the data is retrieved is indeterminate.

When all software data has been read, the Analysis DLL may be asked to free memory used by internal data structures. Whether this function is called or not, the Analysis DLL should be ready to for the next scan file and be able to handle the situation where plug-in data from several scan file is requested in any order.

- The columns defined by the plug-in are queried via the function

  ```
  void FPCALLCONV PlgGetColumns(FPULONG ID, FP_ULONG* Count,
                                TPlgColumnStruc** Columns);
  ```

  The DLL passes back a pointer to the first element in an array of `TplgColumnStruc` structures. For the format of this structure, see the file "`fpPlgInt.h`".

- The initialization call of the DLL takes the form

  ```
  void FPCALLCONV PlgNewData(FP_ULONG ID,
                        const TPlgAnalysisInterface* AnalysisInt,
                            TPlgAnalysisInfo* Info);
  ```

  The AnalysisInt member contains pointers to the functions used to read the blob from the scan file, as well as functions that permit the analysis DLL to obtain the plug-in configuration data from the scan file. The DLL should read the blob back from the scan file, and construct data structures that permit it to extract the correct information depending on the file index it is passed. A reference to internal data structures should again be stored in the Info->InstanceData pointer.

  This function is called for every scan file to be processed, so the Analysis DLL should allocate separate Instance data for each scan file.

- The Analysis tool requests data from the plug-in on a per file/per column basis. The function

  ```
  void FPCALLCONV PlgGetColumnData(void *InstData,
                                   FP_ULONG DataID,
                                   FP_ULONG Column,
                                   TPlgColumnData* Data);
  ```

  requests data from the DLL for the file identified by DataID, and the column identified by Column. The pertinent data should be passed back via one of the members of the TplgColumnData union.

- After all of the files have been read from a scan file, the DLL may be asked to free all of its internal data structures for that scan file via the function

  ```
  void FPCALLCONV PlgFreeData(void *InstData);
  ```

Collecting Information From the New Archive Formats

A plug-in that provides support for new archive formats requires no Analysis DLL. The files that it reports the existence of are included with the rest of the files in the generated fingerprint. This section deals only with the scanning portion of this type of plug-in, as the configuration section is identical to that of a Data File Recognition plug-in.

The general logic of the interface is as follows:

- The scanner invokes the DLL with a call to

  ```
  void FPCALLCONV PlgInit(const TPlgScannerInterface *ScanInt,
                          TPlgScanInfo *Info);
  ```

  The DLL is initialized in the same way that a Data File Recognition Scanner DLL is initialized. However, the DLL sends a flag back to the scanner to indicate that it is an archive scanning DLL (spfArchive, instead of spfDataFileRecognition).

- The Scanner DLL is given a reference to each file as the scanner examines it via a call to

  ```
  FP_BYTE FPCALLCONV PlgIsArchive(void *InstData,
                                  const TPlgFileInfo* ArcFile,
                                  const void *Buffer);
  ```

  During this call, the Scanner DLL can now examine the file to determine if it is an archive of the format(s) that the DLL is meant to recognize. In order for the check to be fast, the archive signature should be checked as this eliminates the need for reading files that are obviously not of the desired archive types. For performance reasons, the 8kb buffer pointed to by the Buffer parameter should be used in reference to reading the file when possible.

  The DLL must return a value of 1 if this is a readable archive, and 0 otherwise.

- If the Scanner DLL indicated that the file was indeed a readable archive, the DLL is repeatedly queried about the files in the archive until it indicates that there are no more files. The method is conceptually identical to a ‘Find First…Find Next’ iteration. The function that is repeatedly called is

  ```
  FP_BYTE FPCALLCONV PlgFindFileInArchive(void *InstData,
                                     const TPlgFileInfo *Archive,
                         TPlgArcFileInfo* FileInArc,
                                          FP_BYTE First);
  ```

  The DLL reports information on the files by filling in the members of the FileInArc parameter, and returning a value of 1. The DLL indicates that there are no more files to report in this archive by returning a value of 0.

When this function is called for the first time, the `First` parameter is set to `1` and `0` otherwise.

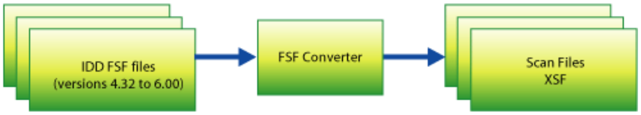
- The DLL is finalized by the scanner after the scan is complete via a call to

  ```
  void FPCALLCONV PlgDone(void *InstData);
  ```

  The DLL is required to free any memory that it has allocated for its own uses.

6 CHAPTER FSF Converter

Use the FSF converter to convert old InfraTools Desktop Discovery Fingerprint files (FSFs) to the Enterprise Discovery version 2.0.0 .xsf format.



Important!

Before converting your FSF files, read the following:

- The FSF converter accepts FSFs from InfraTools Desktop Discovery version 4.40 to version 6.03 (FSF version 4.32 to 6.00).
- In order to achieve the best quality of data we recommend that you re-scan you computer population instead of converting all your existing fingerprint files. The Enterprise Discovery Scanners collect a wealth of information not collected by the InfraTools Desktop Discovery Scanners.
- Scan files created with Enterprise Discovery software cannot be read by InfraTools Desktop Discovery software. That is, there is no backwards compatibility.

Starting the FSF Converter

To start the FSF Converter:

- Select the FSF Converter entry in the Programs|Peregrine|Enterprise Discovery 2.0.0 submenu of the Start menu.

Menus in the FSF Converter Workspace

Three menus are available:

Menu item	Function
File menu	
Convert FSFs...	Displays the FSF conversion wizard.
Exit	Exits from the FSF Converter program
Log menu	
Copy to Clipboard	Copies the entries in the log window to the clipboard.
Copy to File	Allows you to save the entries in the log window to a .tsv (tab separated) file.
Clear Log	Clear any entries from the log window.
Help menu	
About	Provides information about the software.

Converting your FSFs

There are three steps to converting your FSF files

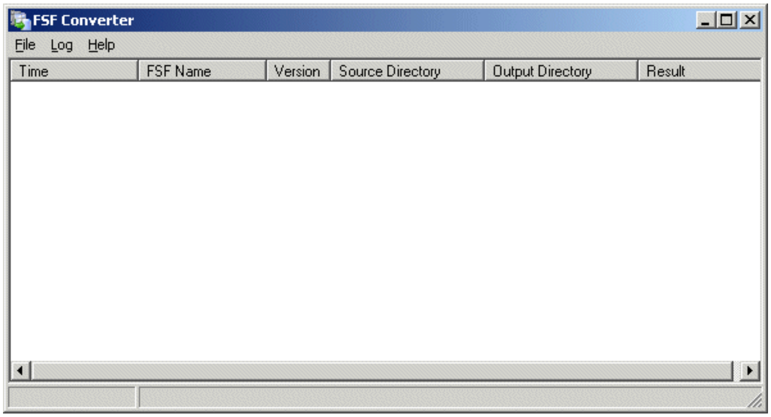
- Choose Which FSFs to Convert
- Select Output Directory
- Choose a Scanner Configuration

Choose Which FSFs to Convert

To convert old FSFs to the Enterprise Discovery version 2.0 .xsf format:

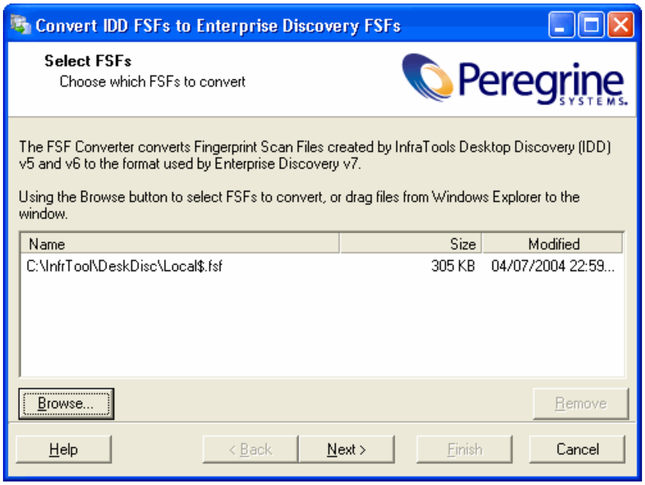
1 Select the FSF Converter entry in the Programs|Peregrine|Enterprise Discovery 2.0.0 submenu of the Start menu.

The FSF Converter main window appears.



2 Select the Convert FSF... option from the File menu.

The Conversion Wizard appears.



Select FSFs

Choose which FSFs to convert

The FSF Converter converts Fingerprint Scan Files created by InfraTools Desktop Discovery (IDD) v5 and v6 to the format used by Enterprise Discovery v7.

Using the Browse button to select FSFs to convert, or drag files from Windows Explorer to the window.

Name	Size	Modified
C:\InfrTool\DeskDisc\Local$.fsf	305 KB	04/07/2004 22:59...

3 Choose which FSFs to convert.

This can be done with the Browse button or by dragging the files from the Windows Explorer to the wizard.

You can remove currently selected (highlighted) items from the list by:

- Clicking the Remove button.
- Pressing the DEL key when the list has focus.

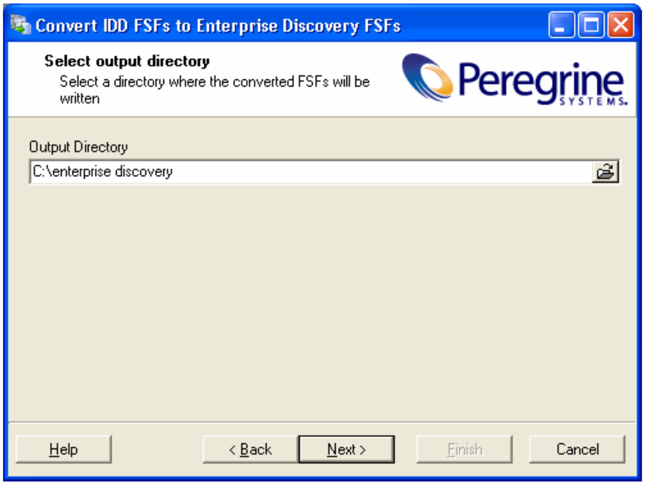
4 The Next button is enabled when at least one file is in the list. Click Next to continue.

Note: Once the wizard has been used once, the conversion process can be started by dropping files on the main window without using the wizard. In this case, the settings from the previous run will be used. If the wizard has never before been used, an error message appears.

```
No Output directory specified. Use the Convert Wizard to define
one before using drag and drop.
```

Select Output Directory

The second page of the wizard is used to choose the output directory for the converted files.



Convert IDD FSFs to Enterprise Discovery FSFs

Select output directory

Select a directory where the converted FSFs will be written

Output Directory

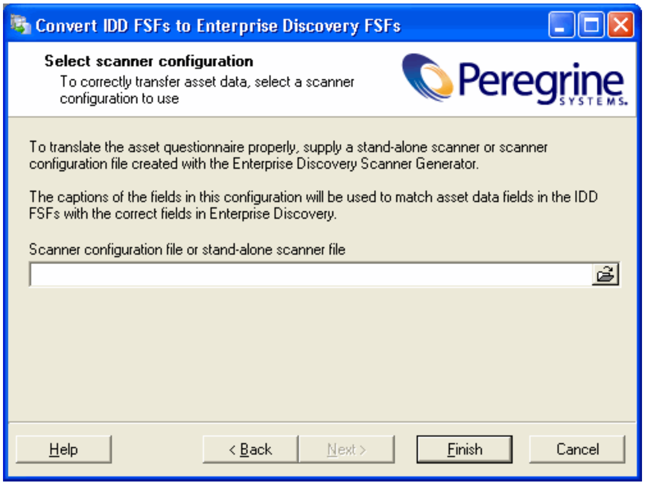
C:\enterprise discovery

Note: The directory must already exist.

1 The Next button is enabled when this field is not blank. Click Next to continue.

Choose a Scanner Configuration

1 On the last page, choose a Scanner configuration to use for converting asset data.



Select scanner configuration

To correctly transfer asset data, select a scanner configuration to use

To translate the asset questionnaire properly, supply a stand-alone scanner or scanner configuration file created with the Enterprise Discovery Scanner Generator.

The captions of the fields in this configuration will be used to match asset data fields in the IDD FSFs with the correct fields in Enterprise Discovery.

Scanner configuration file or stand-alone scanner file

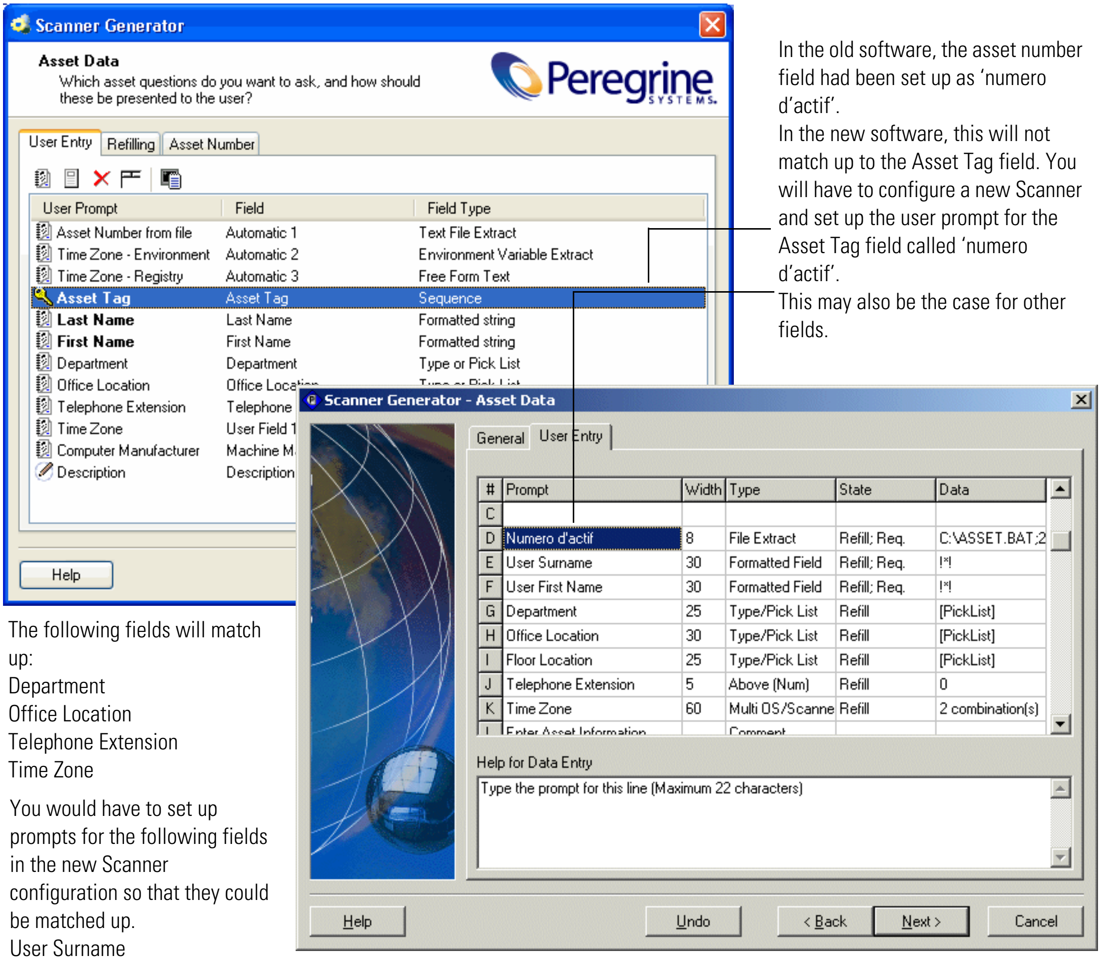
This step is optional and the wizard allows the Finish button to be clicked if the entry field is blank or contains the filename of a valid Enterprise Discovery version 2.0 Scanner or Scanner configuration file.

The reason for doing this is the lack of compatibility between the asset questionnaires for InfraTools Desktop Discovery and Enterprise Discovery 2.0.

You will have to configure a new Scanner using Enterprise Discovery 2.0 so that the captions can be matched up with those that were used in the Desktop Discovery scans.

For example, in the following screen shot, the field called ‘Numero d’actif’ has been set up as the asset number field in the old questionnaire. To ensure that data from this field is stored in the asset number field in Enterprise

Discovery 2.0, configure a new Scanner and set up a user prompt for the asset field as ‘Numero d’actif’ so that the two can be consolidated.



Asset Data

Which asset questions do you want to ask, and how should these be presented to the user?

User Prompt	Field	Field Type
Asset Number from file	Automatic 1	Text File Extract
Time Zone - Environment	Automatic 2	Environment Variable Extract
Time Zone - Registry	Automatic 3	Free Form Text
Asset Tag	Asset Tag	Sequence
Last Name	Last Name	Formatted string
First Name	First Name	Formatted string
Department	Department	Type or Pick List
Office Location	Office Locati	Type or Pick List
Telephone Extension	Telephone	
Time Zone	User Field 1	
Computer Manufacturer	Machine M	
Description	Description	

#	Prompt	Width	Type	State	Data
C					
D	Numero d'actif	8	File Extract	Refill; Req.	C:\ASSET.BAT;2
E	User Surname	30	Formatted Field	Refill; Req.	!*!
F	User First Name	30	Formatted Field	Refill; Req.	!*!
G	Department	25	Type/Pick List	Refill	[PickList]
H	Office Location	30	Type/Pick List	Refill	[PickList]
I	Floor Location	25	Type/Pick List	Refill	[PickList]
J	Telephone Extension	5	Above (Num)	Refill	0
K	Time Zone	60	Multi OS/Scanne	Refill	2 combination(s)
L	Enter Asset Information		Comment		

Help for Data Entry

Type the prompt for this line (Maximum 22 characters)

In the old software, the asset number field had been set up as ‘numero d’actif’.
In the new software, this will not match up to the Asset Tag field. You will have to configure a new Scanner and set up the user prompt for the Asset Tag field called ‘numero d’actif’.
This may also be the case for other fields.

The following fields will match up:
Department
Office Location
Telephone Extension
Time Zone

You would have to set up prompts for the following fields in the new Scanner configuration so that they could be matched up.
User Surname
User First Name

This must be done for all fields that were customized in the Desktop Discovery software and deviated from the standard questionnaire supplied. In the example the following fields would automatically be consolidated without you having to do anything:

- Department
- Office Location

- Telephone Extension
- Time Zone

You would have to set up prompts for the following fields in the new Scanner configuration so that they could be matched up.

- User Surname
- User First Name

2 Once you have configured a new Scanner which reflects the asset fields correctly, use the Scanner in this page of the wizard.

3 When the Finish button is pressed, the selected files are converted.

Verify the Results

To verify the result of the conversion, load a converted xsf scan file into the Enterprise Discovery Viewer and navigate to the Hardware and Configuration tab. Click on the Asset Data folder and inspect the data to verify that the desired asset data fields were converted correctly.

How the Asset Data Conversion Works

When converting asset data, each field in the old questionnaire is inspected in turn. If the 'prompt' of the field matches the 'user prompt' of a field in the new configuration, the data is transferred. If not, the converter looks for a matching field name and transfers the data if a match is found. Thus, a department field from InfraTools Desktop Discovery will be transferred to the Enterprise Discovery field with a prompt of Department if it exists, or else to the Department field.

Note: Field contents are validated, invalid values are discarded. This means that any converted value is validated against the constraints defined in the new Scanner setup. For example, if a field was Free Form in InfraTools Desktop Discovery and is defined as Numeric in Enterprise Discovery, only values that are valid when interpreted as a number are converted.

The FSF Converter Log

For each file converted, a line is added to the main converter log window showing a time stamp, the file name, the FSF version, where it was found, where the converted .xsf file is stored. A Result which is 'Success' if the conversion is successful and an error message if it is not. is also shown.

During the conversion process, a dynamic progress indicator is shown and another import process cannot be started.

The log shown in the main window is persistent, that is, it is stored when the program exits and is reloaded when it starts.

Data items can be sorted by clicking on the column headers.

To clear the log:

- Select the Clear Log option from the File menu.

CHAPTER 7 Copyright

Peregrine Systems acknowledges the copyrights belonging to the following third parties. (This page constitutes a continuation of the copyright page.)

ActivePerl

Commercial support for ActivePerl is available through ActiveState at: http://www.ActiveState.com/Support/Enterprise/.

For peer support resources for ActivePerl issues see: http://www.ActiveState.com/Support/

ActivePerl is the up-to-date, quality-assured Perl binary distribution from ActiveState. Current releases, and other professional tools for open source language developers are available at http://www.ActiveState.com.

Apache Ant

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

This product includes also software developed by :

- the W3C consortium (http://www.w3c.org) ,

- the SAX project (http://www.saxproject.org)

Please read the different LICENSE files present in the root directory of this distribution.

Apache HTTPD

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

Portions of this software were developed at the National Center for Supercomputing Applications (NCSA) at the University of Illinois at Urbana-Champaign.

This software contains code derived from the RSA Data Security Inc. MD5 Message-Digest Algorithm, including various modifications by Spyglass Inc., Carnegie Mellon University, and Bell Communications Research, Inc (Bellcore).

Regular expression support is provided by the PCRE library package, which is open source software, written by Philip Hazel, and copyright by the University of Cambridge, England. The original software is available from ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/

Apache Tomcat

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

Java Management Extensions (JMX) support is provided by the MX4J package, which is open source software. The original software and related information is available at http://mx4j.sourceforge.net.

The Windows Installer is built with the Nullsoft Scriptable Install Sysem (NSIS), which is open source software. The original software and related information is available at http://nsis.sourceforge.net.

cURL

Flex

This product includes software developed by the University of California, Berkeley and its contributors.

GD

Permission has been granted to copy, distribute and modify gd in any context without fee, including a commercial application, provided that this notice is present in user-accessible supporting documentation.

This does not affect your ownership of the derived work itself, and the intent is to assure proper credit for the authors of gd, not to interfere with your productive use of gd. If you have questions, ask. "Derived works" includes all programs that utilize the library. Credit must be given in user-accessible documentation.

This software is provided "AS IS." The copyright holders disclaim all warranties, either express or implied, including but not limited to implied warranties of merchantability and fitness for a particular purpose, with respect to this code and accompanying documentation.

Although their code does not appear in the current release, the authors also wish to thank Hutchison Avenue Software Corporation for their prior contributions.

Getopt

This product includes software developed by the NetBSD Foundation, Inc. and its contributors.

Mod_Perl

The Apache Software License, Version 1.1

Copyright (c) 2000 The Apache Software Foundation. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment:

> This product includes software developed by the Apache Software Foundation (http://www.apache.org/)."

Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

4. The names "Apache" and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org.

5. Products derived from this software may not be called "Apache", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the Apache Software Foundation. For more information on the Apache Software Foundation, please see <http://www.apache.org/>.

Portions of this software are based upon public domain software originally written at the National Center for Supercomputing Applications, University of Illinois, Urbana-Champaign.

MySQL

Enterprise Discovery includes software whose copyright is owned by MySQL, A.B.

Open_SSL

LICENSE ISSUES

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit. See below for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to OpenSSL please contact openssl-core@openssl.org.

OpenSSL License

Copyright (c) 1998-2004 The OpenSSL Project. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. All advertising materials mentioning features or use of this software must display the following acknowledgment:

> “This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (http://www.openssl.org/)”

4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org.

5. Products derived from this software may not be called "OpenSSL" nor may "OpenSSL" appear in their names without prior written permission of the OpenSSL Project.

6. Redistributions of any form whatsoever must retain the following acknowledgment:

> “This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/)"

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

Original SSLeay License

Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com) All rights reserved.

This package is an SSL implementation written by Eric Young (eay@cryptsoft.com).

The implementation was written so as to conform with Netscapes SSL.

This library is free for commercial and non-commercial use as long as the following conditions are aheared to. The following conditions apply to all code found in this distribution, be it the RC4, RSA, lhash, DES, etc., code; not just the SSL code. The SSL documentation included with this distribution is covered by the same copyright terms except that the holder is Tim Hudson (tjh@cryptsoft.com).

Copyright remains Eric Young's, and as such any Copyright notices in the code are not to be removed.

If this package is used in a product, Eric Young should be given attribution as the author of the parts of the library used. This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. All advertising materials mentioning features or use of this software must display the following acknowledgement:

> "This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)" The word 'cryptographic' can be left out if the rouines from the library being used are not cryptographic related :-).

4. If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include an acknowledgement:

> "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The licence and distribution terms for any publically available version or derivative of this code cannot be changed. i.e. this code cannot simply be copied and put under another distribution licence [including the GNU Public Licence.]

Quartz

This product includes software developed by the OpenSymphony Group (http"//www.opensymphony.com/).

Unicode

International Components for Unicode: Copyright (c) 1995-2003 International Business Machines Corporation and others. All rights reserved. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.

Xerces

Copyright (c) 1999 The Apache Software Foundation. All rights reserved. This product includes software developed by the Apache Software Foundation ([1]http://www.apache.org/).

PEREGRINE

Index

Peregrine SYSTEMS.

9/14/05