HP Server Automation

Software Version: 7.50

Software Discovery Technical Note

Legal Notices

Warranty

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

The information contained herein is subject to change without notice.

Restricted Rights Legend

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

Third-Party Web Sites

HP provides links to external third-party Web sites to help you find supplemental information. Site content and availability may change without notice. HP makes no representations or warranties whatsoever as to site content or availability.

Copyright Notices

Trademark Notices

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

Oracle® is a registered US trademark of Oracle Corporation, Redwood City, California.

Unix® is a registered trademark of The Open Group.

Documentation Updates

This manual's title page contains the following identifying information:

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

To check for recent updates, or to verify that you are using the most recent edition of a document, go to:

http://ovweb.external.hp.com/lpe/doc_serv/

Technical Note: Software Discovery

Introduction to Software Discovery

The HP Server Automation (SA) Software Discovery feature provides a signature-based software discovery mechanism for Windows and UNIX managed servers to help you manage applications and software that are not managed by SA. Specifically, Software Discovery enables you to discover software that was not installed using one of the standard packaging technologies for Windows or Unix, software that is unlicensed or unregistered, and software that was custom-built.

Once the you have discovered all the software on a server (or group of servers,) you can create an "inventory" snapshot of the server and know exactly what software is installed on the server. This can help when you need to upgrade a server, uninstall unwanted software, or make a server conform to your organizations security policies. You can also use an audit to capture the current state of a server's software installations and, by running the audit on a regular schedule, you can monitor any installation changes over time.

Specifically, the Software Discovery feature:

- Discovers unregistered software which is not currently managed by SA.
- Creates an inventory of software that is not installed as part of an OS-registered application.
- Provides system administrators the ability to create snapshots of the discovered software on a server and then periodically audits against the snapshots over time.
- Enables system administrators to track in-house or custom built software.
- Gives auditors a convenient method for discovering unsupported or unlicensed software installed on a server.

Software Discovery is a *read-only* server object designed to provide a rich software inventory of an SA managed server.

Software Discovery Prerequisites

In order to deploy the Software Discovery feature, you must meet the following requirements:

- SA 7.50 installed and configured
- 7.50.01 patch upgrade installed on the SA 7.50 core
- An HP Live Network (HPLN) account, included with standard maintenance
- HP Live Network connector installed and configured on your core
- Additionally, the ETL/Model needs to be installed before the Discovered Software feature can be imported and used. Failure to do so will create issues in the Software Discovery report data. Make sure to perform the following tasks, in this order:
    1. Subscribe to the ETL/Model/Reports stream and allow it to be deployed into SAR on your core
    2. Subscribe to the Discovered Software stream on the HP Live Network

For information on how to subscribe to the Software Discovery stream, consult the HP Live Network site.

For information on installing and configuring the HP Live Network connector, consult the HP Live Network connector User's Guide found on the HPLN Portal.

To request an HPLN account, visit the following URL: https://support1.opsware.com/livenetaccess.php

Supported Platforms and Configurations

Software Discovery is supported on all UNIX platforms and Windows versions which the SA Agent supports for managed servers.

For more information on managed servers supported platforms, see the SA 7.50 Release Notes or the SA 7.50 Planning and Installation Guide.

Software discovery is ISO-8859-1 compliant.

Software Discovery in SA

The Software Discovery feature is distributed via the HP LiveNetwork connector (LNc). After a new version of a Software Discovery package is available at HPLN, and is downloaded by the LNc (which is configured to run and receive content) to an SA core machine, notifications will be posted on the HPLN server module project portal.

Once the Software Discovery packages are available, the HPLN connector downloads the Software Discovery package in the form of a zip file with the following name:

`OPSWsmo_discovered_software-<version>.zip`

to the following directory on your SA core:

`/var/opt/opsware/ogfs/mnt/root/var/opt/opsware/sm`

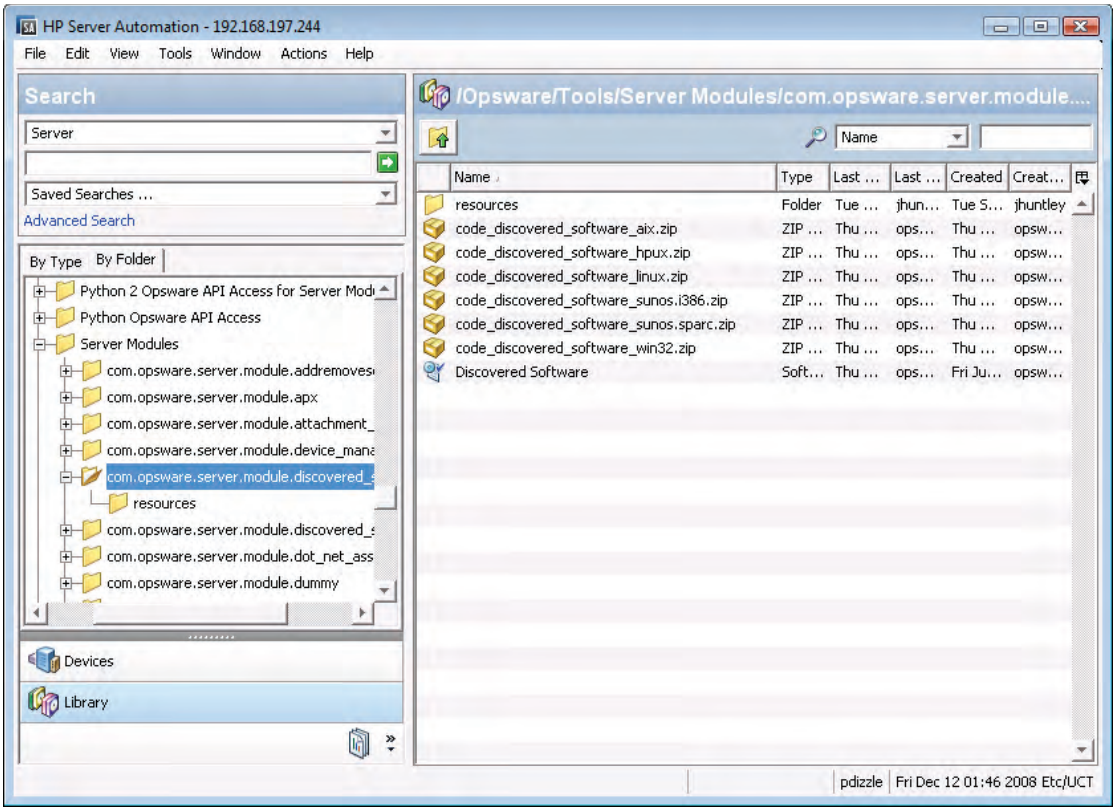
This directory can also be accessed from the SA Global Shell with the following command:

```
/var/opt/opsware/sm
```

Software Discovery in the SA Client

The installed components of the Software Discovery feature consist of zipfile packages for each supported managed server platform that is accessible from inside the SA Client Library, as shown in Figure 1-1.

Figure 1-1: SA Client Library Showing location of Software Discovery Packages



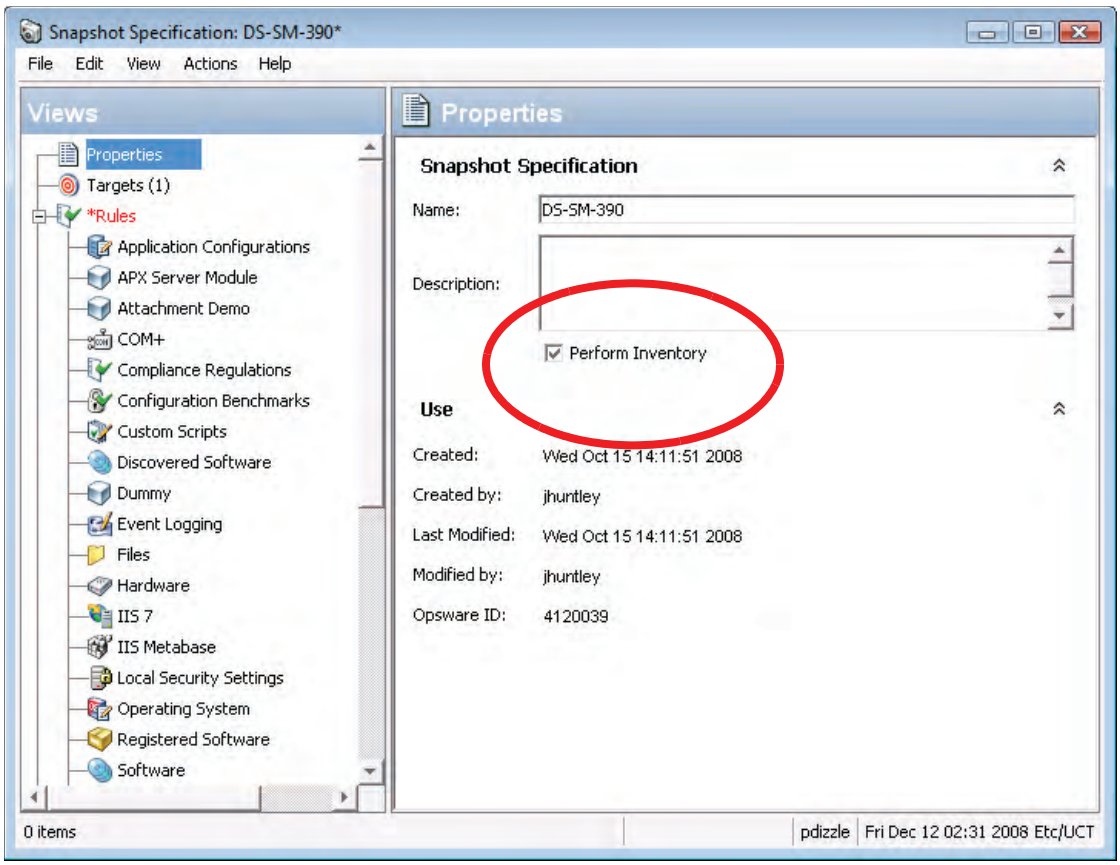
The “Resources” directory contains the `rsrc.zip` file, which itself contains images referred to in `module.jsn` and localization files.

These zip packages that represent the Software Discovery feature are bundled into a Software Policy (named “Discovered Software) that becomes automatically remediated (installed) when the user initiates a Snapshot with the” Perform Inventory” option selected.

Deploying Software Discovery via “Inventory” Snapshots

In order to deploy the Software Discovery feature to inventory software installed on a server, you must run a Snapshot with the Perform inventory option selected, as shown in Figure 1-2.

Figure 1-2: Snapshot with Perform Inventory Option Selected to Enable Software Discovery



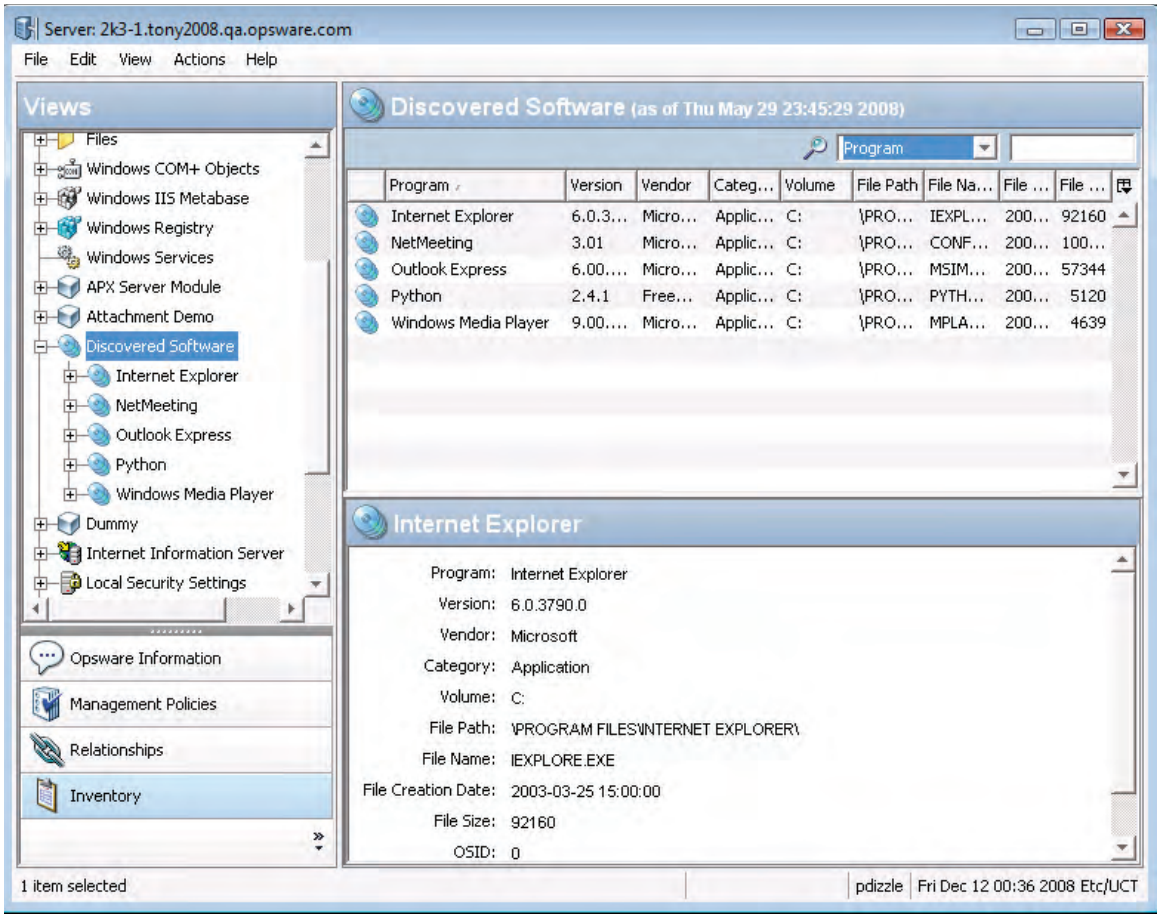
Each time a snapshot is run on a server with Perform Inventory selected (or is initiated via the SA API), SA verifies that the Software Discovery is installed on the managed server. If there is no record of a Software Discovery zip package having been installed on the target server of the Snapshot, a remediate job is launched automatically by the core that installs the contents of the Software Discovery Software Policy. The policy remediation occurs automatically when the snapshot is run.

This results in the server module binary packages being installed and unzipped on the server, including the signature zip package, which in turn enables a user to view Software Discovery on a managed server’s Device Explorer, and or utilize the Snapshot’s software inventory in an Audit.

Viewing Discovered Software on a Managed Server

The SA Client displays Software Discovery data as a *server object* named Discovered Software in the Device Explorer in the Inventory view, as shown in Figure 1-3.

Figure 1-3: Device Explorer Showing Software Discovered by Software Discovery



Unlike other server objects in the managed server inventory that fetch data in real time from a server, the Discovered Software server object is configured to only show data from the latest inventory Snapshot (a Snapshot that has the Inventory checkbox checked when the snapshot specification is created). If there are no inventory snapshots available, the server browser will show an error message suggesting that an inventory Snapshot should be made.

The Device Explorer does display a timestamp indicating the last time the inventory Snapshot was run on the server, at the top of the Device Explorer window.

For example, in Figure 1-3, the Contents pane (right side) of the window shows the label: "(as of Thu May 29th 23:45:29 2008)". This indicates the last time the inventory Snapshot was run.

For more information on how to create and run inventory Snapshots, see the *SA User's Guide: Server Automation*, the Audit and Remediation chapter.

Inventory Snapshots in Audits

Once you have performed an inventory Snapshot, you can add it as the source of an audit, so whenever the audit runs, it will compare the original state of the snapshot with the current state of the server, and you can determine if the installed software has changed since you last ran the Snapshot.

For more information on how to use Snapshots with Audits, see the Software Discovery feature, see the *SA User's Guide: Server Automation*, the Audit and Remediation chapter.

Inventory Snapshot Job Error Messages

The following error codes and messages are specific to Software Discovery as seen when running an inventory Snapshot:

* 1-TADNSW unexpectedly quit!
* 2-Invalid JSON format
* 3-Another instance of DS-SM is running.
* 4-Database merge failed
* 6-TADNSW run failure
* 8-TADNSW Error: `<error>`

Software Discovery Permissions

The Software Discovery feature can be run on a server by any SAS Web Client user who has read access to the server and belongs to a user group that has been granted the client feature permission **Allow Execute Server Modules**.

If a user wishes to add or modify custom entries for Software Discovery, then the user will require **Manage Server Module Read/Write** permission. Permissions can be granted in the SAS Web Client.

For more information on SA permissions, please see the *SA Administration Guide*.

How Software Discovery Works

Software Discovery provides a signature-based software discovery mechanism for SA Windows and UNIX managed servers. It consists of discovery logic, module metadata, and a signature database. This signature database contains application signatures from HP's Asset Management Tool, Discovery Dependency Mapping Inventory (DDMI), which is the Software Application Index (SAI) used by DDMI to discover software.

Software Discovery is a read-only server module designed to provide a rich software inventory of an SA managed server.

This feature is not designed to allow you to remediate or install software on a managed server. For information on using Software Policies to install and remediate software onto managed servers, consult the *SA User's Guide: Server Automation*.

Application Discovery

Software Discovery uses DDMI SAI content for discovering applications on SA managed servers. The SAI signatures are used to generate the SCT (Signature Component Table) for use with Software Discovery. The SCT contains all the signatures imported from the DDMI SAI which are to be used by Software Discovery.

SCT application signatures are grouped together by product IDs and compared to the results retrieved by a file system scan. For every file found on the file system, a hash is computed using the DDMI hash algorithm. The signatures collected from the file scan are referred as 'raw' signatures. Application signatures represented in the DDMI SAI and corresponding SCT contain the same hash values computed from the same algorithm.. The 'raw' signatures are then compared to the DDMI SAI signatures stored in SCT and a rating is calculated off the best possible match. The highest rated match is then reported as the application back to the user who initiated the scan.

All components are compared to the corresponding properties obtained from raw signatures. When a match occurs in any one of the components, the rating is incremented and used for gauging the best possible match among various product versions. When the best match is found, the product is reported using the display components:

```
DISPLAY PRODUCT: <display product-name>
DISPLAY VENDOR: <display vendor-name>
```

```
DISPLAY VERSION: <display version-number>
```

There are a few differences to note between the DDMI and Software Discovery results that are accepted around specific boundary conditions. The first difference in comparison is that Software Discovery will report all instances of software discovered on a server. For example, when evaluating multiple installs of the Java JRE. DDMI will only report the first install of a series of products of the same version. On the other hand, Software Discovery will report all instances of the installed product.

The second difference occurs when no exact matching signature is available in the SAI. In this situation, DDMI and the Software Discovery feature will attempt to find the next best match based off the ratings calculated. DDMI will evaluate all signatures, along with all 'associated' entries, to generate a high rating during guess estimates and find more accurate results. The Software Discovery feature will sometimes reach the same results in the guess. However, the guess does occasionally differ due to remaining 'associated' entries not being imported into SCT from the SAI.

Please note again that due to sizing constraints, not all SAI associated entries are imported into SCT. Only overlapping 'associated' entries between concurrent applications versions will be imported as well. As a result, SA will not pinpoint or discover suites or editions of applications.

Application Signatures

All application signatures are generated using the `DDMI SAI Master.xml` (WIN32) and `unix.xml` (Unix). All application signatures are stored in a database for processing and identification during the scan process. SCT is used for the software discovery process in the database.

During a core install, these application signature packages must be uploaded to the core and attached to the Software Discovery Software Policy to which the Software Discovery package is attached. During upload, each package is associated with all UNIX or all Windows platforms. These packages are automatically installed onto a managed server when the server is remediated with respect to this policy. These signatures are used by the underlying scanning software to determine what software is discovered on a server.

Standard Signatures

The Software Discovery feature comes with a database consisting of signatures derived from the DDMI SAI content. When the Software Discovery package is imported in the core, all applications represented by those signatures are discovered. This signature database is updated periodically (usually monthly). These updates are provided through HPLN and must be imported into the SA core.

The Software Discovery signature update package is a zip file, which has a file name similar to:

```
OPSWsmo_discovered_software_sig_prod-<version>.zip
```

After the HPLN connector downloads the zip file, it is stored in this directory on a core:

```
/var/opt/opsware/ogfs/mnt/root/var/opt/opsware/sm/data
```

This location can also be accessed from the SAS Global Shell with the following command:

```
/var/opt/opsware/sm/data
```

The zip file can be uploaded to the core with dstool with the following command:

```
/opt/opsware/smtool/dstool --user=<username> --
pass=<password> OPSWsmo_discovered_software_sig_prod-
<version>.zip
```

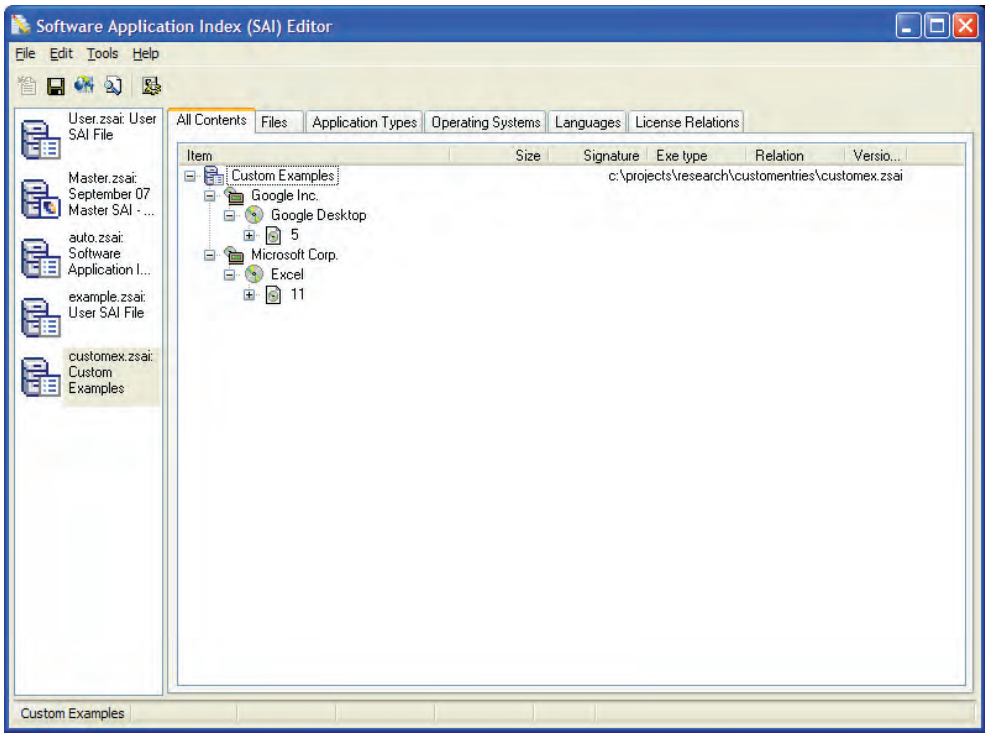
The --help option gives additional details on how to use dstool:

```
/opt/opsware/smtool/dstool --help
```

Custom Signatures

Custom entries are added and maintained by the user using the DDMI SAI Editor and corresponding platform scanners as shown in Figure 1-4.

Figure 1-4: Software Application Index (SAI) Editor



The SAI editor provides a convenient user interface for adding new publishers and their corresponding applications. Its scanners are used for retrieving raw application signatures related to the specific applications you are interested in reporting as applications to SA.

It is recommended that you run the scanner prior to working with the SAI editor. The entire file system or specific directories can be targeted with the scanner by using the following command line options:

```
scanwin32-x86-2.50.000.7199.exe -fast -
p:C:\projects\research\edscan\ -paths:"C:\Program Files"
```

(`scanwin32-x86-2.50.000.7199.exe` is the latest version of the executable and may change without notice.)

In the SAI Editor, you can add the publisher, application, release, and version details for the specific applications that need to be categorized in SAI as shown in Figure 1-5 through Figure 1-8.

Figure 1-5: Publisher Properties

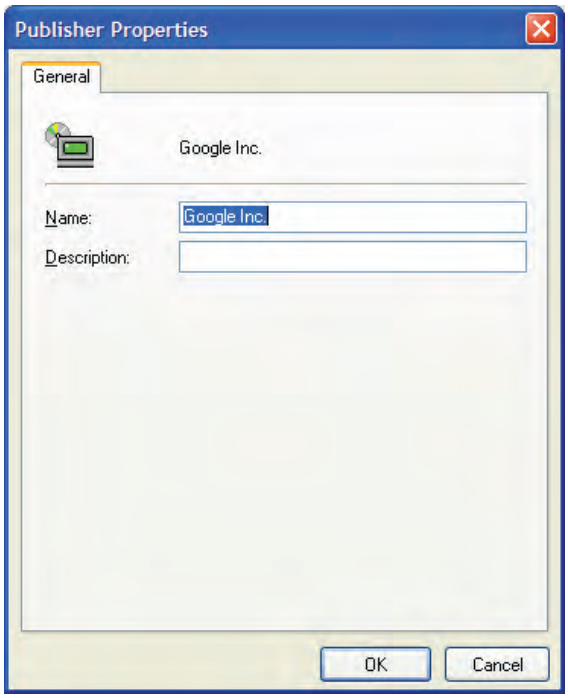


Figure 1-6: Application Properties

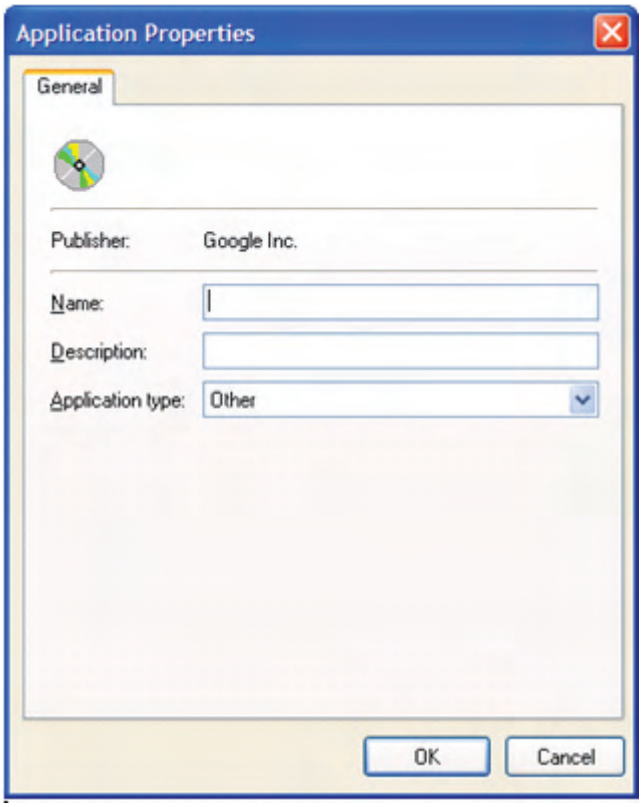


Figure 1-7: Release Properties

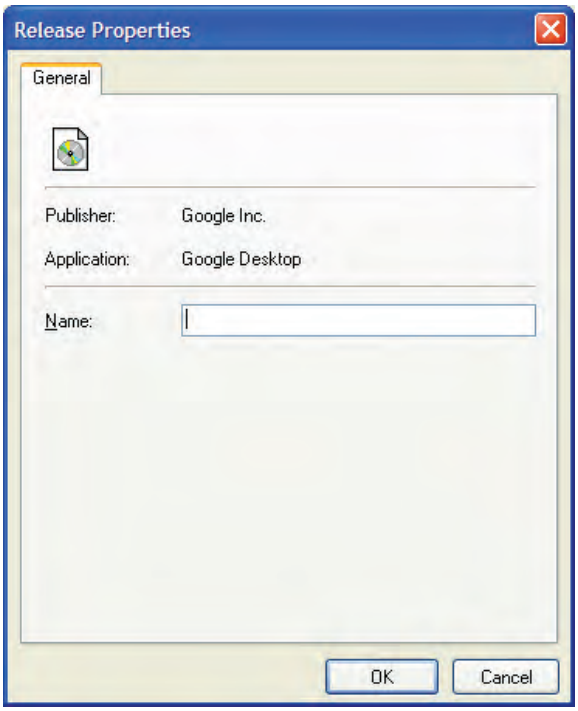
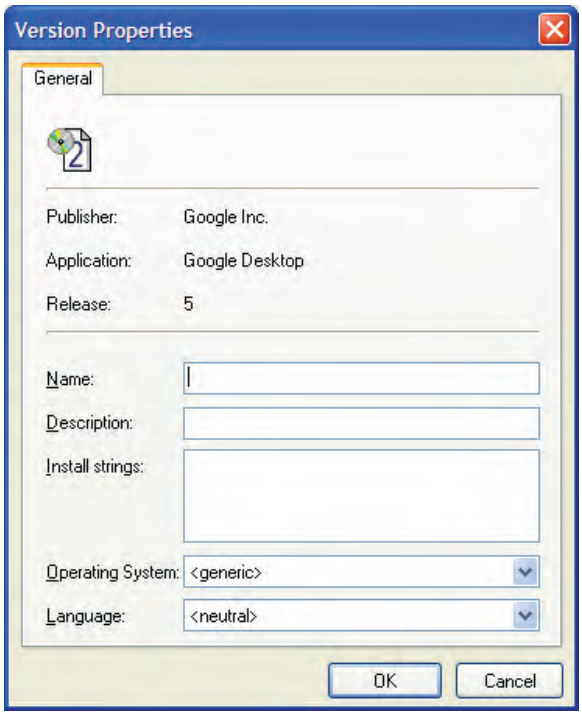


Figure 1-8: Version Properties



At this stage, you can open the scan results in the SAI Editor and begin adding corresponding signatures to the versions previously added in the SAI Editor as shown in Figure 1-9 and Figure 1-10.

Figure 1-9: Recognition Verification

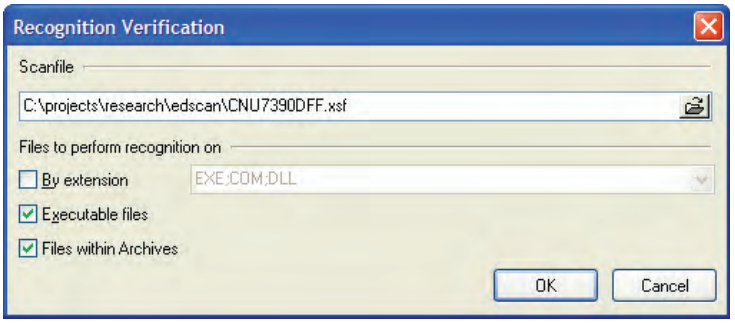
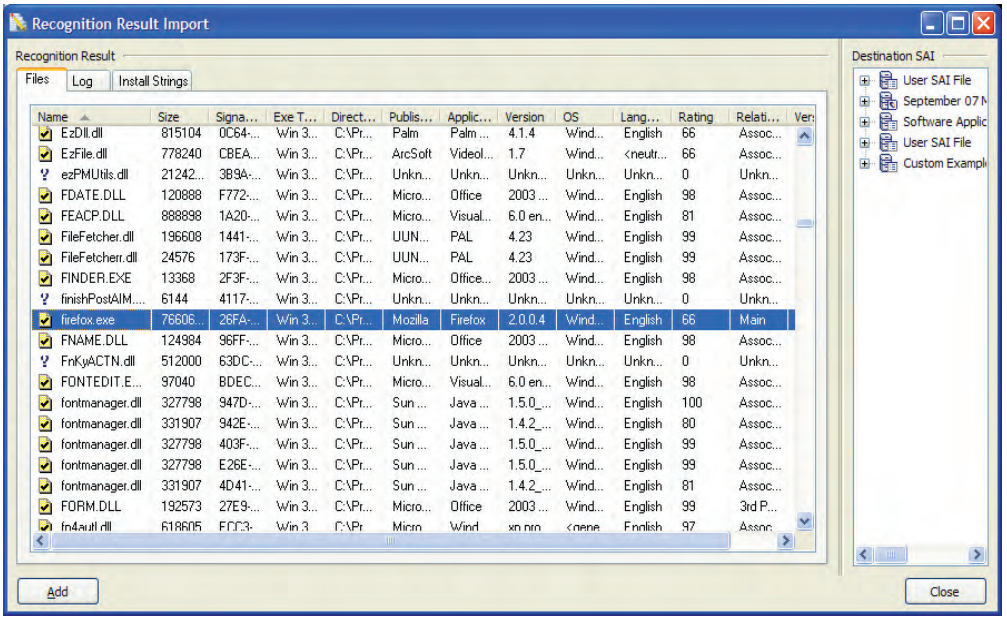
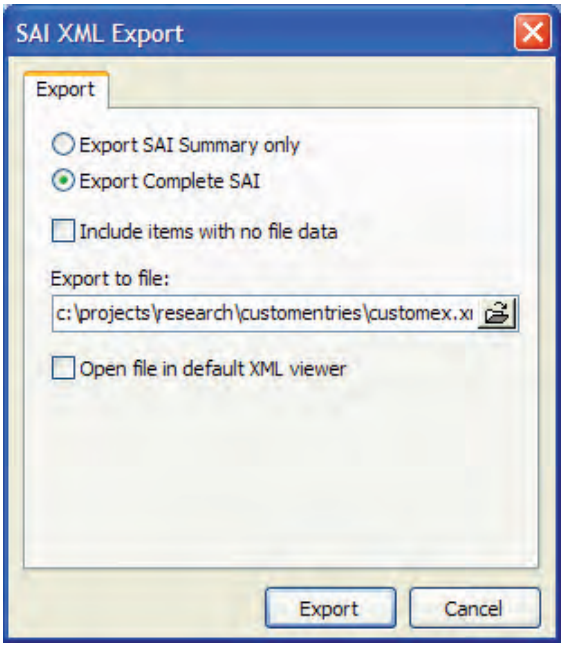


Figure 1-10: Recognition Result Import



Once the content is modified you can then export the content to the SAI XML database (User XML) as shown in Figure 1-11.

Figure 1-11: SAI XML Export



The User XML database is then copied using the following command to the core where it is used by dstool:

```
/opt/opsware/smtool/dstool --username=<username> --
password=<password> customex.xml
```

dstool generates a new (User) SCT database which is uploaded as a custom dependency to Software Discovery. During the next inventory snapshot, Software Discovery checks for the custom database and uses it along with the production database to categorize new applications.

If at any stage a mistake is made or you choose to remove all custom entries, invoke the dstool with the following command line options:

```
/opt/opsware/smtool/dstool --remove --username=<username> --
password=<password>
```

Use the following --help option for additional details about the usage of dstool:

```
/opt/opsware/smtool/dstool --help
```

Please refer to the DDMI SAI Editor and scanner documentation for more details.

Configuration and Customization

UNIX and Windows Software Discovery supports configuration attributes which drive how they run and the profile of resource usage on a managed server.

Configuration Attributes

The name of the custom attribute is `HPSW DS SM_CONFIG`.

The value of this custom attribute is in INI file format in Windows. Windows filters are grouped under the FILTERS_WINDOWS section while Unix filters are grouped under the FILTERS_UNIX section. The METHODS section specify the scanning settings and the CONFIGURATION section contains other configuration settings.

The following is the syntax of a custom attribute:

```
[FILTERS_WINDOWS]

FILTER#={INCLUDE | EXCLUDE | EXIST}, { FILENAME | FILEPATH |
FILESYSTYPE | VOLUME | PRODUCT | VENDOR | VERSION},  criteria


[FILTERS_UNIX]

FILTER#={INCLUDE | EXCLUDE | EXIST}, { FILENAME | FILEPATH |
FILESYSTYPE | VOLUME | PRODUCT | VENDOR | VERSION},  criteria


[METHODS]

DELTA_SCAN= {TRUE | FALSE}


[CONFIGURATION]

LOG_MODE= {LOW | MEDIUM | HIGH | DEBUG}

INCLUDE_SYMBOLIC_LINKED_FILES={TRUE | FALSE}

INCLUDE_SYMBOLIC_LINKED_DIRS={TRUE | FALSE}
```

Configuration Options

Table 1-1 describes the configuration options used within the HPSW_DS_SM_CONFIG custom attribute.

Table 1-1: Configuration Options

ATTRIBUTE	PLATFORM	DESCRIPTION
'FILTER#'	All	Create a list of filters for the Software Discovery Gauntlet for Cataloged and/or Uncataloged software. Filters are separated into two categories under INI format according to platform, [FILTERS_WINDOWS] and [FILTERS_UNIX]. Filter arguments are provided below their corresponding platform in the format of FILTER#=Action,Type,Criteria. Action can be one of 'Include', 'Exclude', or 'Exist'. Supported types are 'VOLUME', 'FILEPATH', 'FILENAME', 'PRODUCT', 'VENDOR', 'VERSION', 'FILESYSTYPE'. Criteria must correspond to the type.
'INCLUDE_SYMBOLIC_LINKED_FILES'	Unix	True/False Default is False. Enable if you wish to include symbolic linked files in the scan.
'INCLUDE_SYMBOLIC_LINKED_DIRS'	Unix	True/False Default is False. Enable if you wish for tadnsw to traverse into symbolic linked directories.
'DELTA_SCAN'	All	Enabled/Disabled or True/False. Default is False. Enable/Disable the delta scanning module.

Table 1-1: Configuration Options

ATTRIBUTE	PLATFORM	DESCRIPTION
'LOG_MODE'	All	Value is one of "LOW", "MEDIUM", "HIGH or "DEBUG", default is "LOW". Sets the amount of information written to the log file. "LOW" is used to provide general runtime progress. "MEDIUM" will provide additional details regarding progress along with location if performing filescan. If “DEBUG”or "HIGH" is selected, the module will write enough information to the file so developers can diagnose obscure problems.

Syntax Errors

Refer to “Inventory Snapshot Job Error Messages” on page 7 to see all the Software Discovery error messages and codes. Prior to using the syntax, the values in the custom attributes in Table 1-1 are validated for proper syntax. If the configuration values violate the syntax, Software Discovery returns errors. Such messages are reported in place of <error> referenced in the list in “Inventory Snapshot Job Error Messages” on page 7. Validation only applies to the INCLUDE/EXCLUDE action assignments. If other invalid configuration options are specified, the default values override

Discovery Filtering Process

The use of Software Discovery filters benefits users by preventing unwanted data from consuming valuable network and database resources. The filtering scheme supports three actions, which are Include, Exist and Exclude.

The Software Discovery feature first checks the Include list to see what files should be examined. If a file is included, it progresses to the next step called **Exist**, which checks to make sure that selected fields contain certain data. Last, the discovery module compares the scanned data with all required fields against the **Exclude** list to see if the file should be excluded. If a file passes all configured filters, it is reported, otherwise the file is

discarded and the discovery module continues scanning. All criteria fields accept inputs of <criteria>, prefix, postfix, as well as wild cards. If a wild card is specified at the end of the path string, the exact path is matched and sub-directories are not affected.

How Filters Affect Scanning

Include and exclude filters can affect the Software Discovery in a variety of ways. In general, the rules listed below apply.

Directory Scanning:

- If a directory is excluded, exclude all files and subdirectories.
- If a directory is included, include all files and subdirectories.
- If a directory does not match the include set, the status stays neutral and only the selected directories are scanned.
- If no directories are included or excluded, the status is set to inlcude.

File Scanning:

- If a file is excluded, that file is not reported.
- If a file is included, that file is reported.
- If no files are included or excluded, the default status is set to include.

Sample Configuration

The following is a sample of a custom attribute in INI format.

```
[FILTERS_WINDOWS]
FILTER0=INCLUDE, VOLUME, "C:"
FILTER1=EXCLUDE, FILEPATH, "<WINDIR>\INSTALL*"
FILTER2=EXCLUDE, FILEPATH, "<WINSYSDIR>\DLLCACHE\"
FILTER3=EXCLUDE, FILEPATH, "<WINSYSDIR>\DRIVERS\"
FILTER4=EXCLUDE, FILENAME, "SETUP*"
FILTER5=EXCLUDE, FILEPATH, "\RECYCLE*"
FILTER6=EXCLUDE, FILEPATH, "\SYSTEM VOLUME INFORMATION*"
FILTER7=EXCLUDE, FILEPATH, "*\TEMPORARY INTERNET FILES\*"
FILTER8=EXCLUDE, FILEPATH, "*\LOCAL SETTINGS\TEMP\*"
FILTER9=EXCLUDE, FILEPATH, "*\LOCAL SETTINGS\HISTORY\*"
```

```
FILTER10=EXCLUDE, FILEPATH, "*DLLCACHE*"
FILTER11=EXCLUDE, FILEPATH, "*$NTUNINST*"
FILTER12=EXCLUDE, FILEPATH, "*$NTSERVICEPACKUNINSTALL$*"
FILTER13=EXCLUDE, FILEPATH, "\I386"
FILTER14=EXCLUDE, FILEPATH, "*SP4\*"
FILTER15=EXCLUDE, FILEPATH, "*$HF_*"
FILTER16=EXCLUDE, FILEPATH, "*SERVICEPACKFILES*"
FILTER17=EXCLUDE, FILEPATH, "<WINSYSDIR>\WinSxS\"
FILTER18=INCLUDE, FILEPATH, "\DOCUMENTS AND SETTINGS*"
FILTER19=INCLUDE, FILEPATH, "\PROGRAM FILES*"
FILTER20=EXIST, PRODUCT
FILTER21=EXIST, VERSION

[FILTERS_UNIX]
FILTER0=EXCLUDE, FILENAME, "*.dll"
FILTER1=EXCLUDE, FILENAME, "*.com"
FILTER2=EXCLUDE, FILENAME, "*.cmd"
FILTER3=EXCLUDE, FILENAME, "*.html"
FILTER4=INCLUDE, FILEPATH, "/etc*"
FILTER5=INCLUDE, FILEPATH, "/opt*"
FILTER6=INCLUDE, FILEPATH, "/bin*"
FILTER7=INCLUDE, FILEPATH, "/usr/bin*"
FILTER8=INCLUDE, FILEPATH, "/usr/lib*"
FILTER9=INCLUDE, FILEPATH, "/usr/games*"
FILTER10=INCLUDE, FILEPATH, "/usr/sbin*"
FILTER11=EXCLUDE, FILEPATH, "/cygdrive*"
FILTER12=EXCLUDE, FILEPATH, "/lost+found*"
FILTER13=EXCLUDE, FILEPATH, "/proc*"
FILTER14=EXCLUDE, FILEPATH, "/tmp*"
```

```
[METHODS]
DELTA_SCAN=TRUE

[CONFIGURATION]
LOG_MODE=LOW
INCLUDE_SYMBOLIC_LINKED_FILES=FALSE
INCLUDE_SYMBOLIC_LINKED_DIRS=FALSE
```

Concurrency and Multi-User Considerations

Software Discovery is configured to look at a Snapshot before attempting to run the inventory Snapshot. If an inventory Snapshot is available, the results are immediately displayed instead of running a new inventory Snapshot. However, if two users initially run an inventory Snapshot, one of the instances will be accepted and run an inventory Snapshot. The other user will see a message when attempting to view the Discovered Software server object in the Device Explorer alerting them that an inventory Snapshot is already in progress.

If Software Discovery is used during an ad-hoc or scheduled Snapshot job, the data from the Discovered Software server object is persisted as a Snapshot package (snapshot.zip) on the SA core. These zip packages are used, for example, by the SAS Web Client when it needs to perform an Audit operation.

Software Discovery Usage Examples

The Software Discovery discovery mechanism can be useful in case of inherited servers and other usage examples. Some examples are listed below:

Example 1

You are an IT Administrator at a company that has 500 managed servers in your data center. You want to know how many servers have 1.3 Java JRE installed. You create a new **Audit** and select a source server that you know has Java JRE installed. You perform the following steps:

1. From the **Discovered Software** tab under **Rules** select and expand **Software** Node.

2. Selects **JRE** installed on the source server and add an additional regex property check for version='1\.3.*'.
3. Form the **Targets** tab select all the active servers in the lab.
4. Save and run the audit to check all the managed servers in the lab for 1.3 versions of Java JRE.

The results come back and you see there are still some servers that are not compliant and proceed to update the remaining servers.

Example 2

For the next task, you want to check all 64-bit machines and see what 32-bit applications were installed. You create a new policy and add your own configuration for Software Discovery using the `HPSW_DS_SM_CONFIG` custom attribute. In that attribute, you assign the following options:

```
[FILTERS_WINDOWS]
FILTER0=INCLUDE, VOLUME,   "C:"
FILTER1=INCLUDE, FILEPATH, "\PROGRAM FILES (X86)*"

[METHODS]
DELTA_SCAN=TRUE

[CONFIGURATION]
LOG_MODE=LOW
```

You attach all the 64-bit Windows 2003 Servers to the new policy and remediate the machines. Next, you create a new snapshot including all 64-bit Windows 2003 Servers and select the wildcard under the **Discovered Software** tab. You save the snapshot and proceed to run the snapshot against all the 64-bit Windows 2003 Servers. Results come back and now you have a list of all the 32-bit applications deployed across all the 64-bit Windows 2003 Servers.

Example 3

You are auditing the lab to make sure all Windows servers have the appropriate virus detection software installed. You create a new Audit and select a source server you know has the correct software installed. You navigate to the **Discovered Software** tab and select **Norton Antivirus** under the **Software** node.

You include all the Windows machines in the lab as target machines and then save the Audit. The Audit is then run and the results come back with half the machines in the lab non-compliant. You proceed to install Norton Antivirus on the remainder noncompliant servers.