Symantec™ Data Loss Prevention Lookup Plug-In Guide

Version 11.0

Symantec

Symantec Data Loss Prevention Lookup Plug-In Guide

Documentation version: 11.0

Legal Notice

Symantec Corporation
350 Ellis Street
Mountain View, CA 94043

http://www.symantec.com

Technical Support

Symantec Technical Support maintains support centers globally. Technical Support's primary role is to respond to specific queries about product features and functionality. The Technical Support group also creates content for our online Knowledge Base. The Technical Support group works collaboratively with the other functional areas within Symantec to answer your questions in a timely fashion. For example, the Technical Support group works with Product Engineering and Symantec Security Response to provide alerting services and virus definition updates.

Symantec's support offerings include the following:

- A range of support options that give you the flexibility to select the right amount of service for any size organization
- Telephone and/or web-based support that provides rapid response and up-to-the-minute information
- Upgrade assurance that delivers automatic software upgrades protection
- Global support purchased on a regional business hours or 24 hours a day, 7 days a week basis
- Premium service offerings that include Account Management Services

For information about Symantec's support offerings, you can visit our web site at the following URL:

www.symantec.com/business/support/

All support services will be delivered in accordance with your support agreement and the then-current enterprise technical support policy.

Contacting Technical Support

Customers with a current support agreement may access Technical Support information at the following URL:

www.symantec.com/business/support/

Before contacting Technical Support, make sure you have satisfied the system requirements that are listed in your product documentation. Also, you should be at the computer on which the problem occurred, in case it is necessary to replicate the problem.

When you contact Technical Support, please have the following information available:

- Product release level

- Hardware information
- Available memory, disk space, and NIC information
- Operating system
- Version and patch level
- Network topology
- Router, gateway, and IP address information
- Problem description:
  - Error messages and log files
  - Troubleshooting that was performed before contacting Symantec
  - Recent software configuration changes and network changes

Licensing and registration

If your Symantec product requires registration or a license key, access our technical support web page at the following URL:

www.symantec.com/business/support/

Customer service

Customer service information is available at the following URL:

www.symantec.com/business/support/

Customer Service is available to assist with non-technical questions, such as the following types of issues:

- Questions regarding product licensing or serialization
- Product registration updates, such as address or name changes
- General product information (features, language availability, local dealers)
- Latest information about product updates and upgrades
- Information about upgrade assurance and support contracts
- Information about the Symantec Buying Programs
- Advice about Symantec's technical support options
- Nontechnical presales questions
- Issues that are related to CD-ROMs or manuals

Support agreement resources

If you want to contact Symantec regarding an existing support agreement, please contact the support agreement administration team for your region as follows:

Asia-Pacific and Japan	customercare_apac@symantec.com
Europe, Middle-East, and Africa	semea@symantec.com
North America and Latin America	supportsolutions@symantec.com

Additional enterprise services

Symantec offers a comprehensive set of services that allow you to maximize your investment in Symantec products and to develop your knowledge, expertise, and global insight, which enable you to manage your business risks proactively.

Enterprise services that are available include the following:

Managed Services	These services remove the burden of managing and monitoring security devices and events, ensuring rapid response to real threats.
Consulting Services	Symantec Consulting Services provide on-site technical expertise from Symantec and its trusted partners. Symantec Consulting Services offer a variety of prepackaged and customizable options that include assessment, design, implementation, monitoring, and management capabilities. Each is focused on establishing and maintaining the integrity and availability of your IT resources.
Education Services	Education Services provide a full array of technical training, security education, security certification, and awareness communication programs.

To access more information about enterprise services, please visit our web site at the following URL:

www.symantec.com/business/services/

Select your country or language from the site index.

Contents

Chapter 1

Introduction to lookup plug-ins

This chapter includes the following topics:

About custom attributes and lookup plug-ins

When an incident is created, the Enforce Server retrieves data regarding that incident. Some of that data is in the form of "attributes." See the *Symantec Data Loss Prevention Administration Guide* for more information about incident attributes.

"Custom attributes" are a particular kind of attribute that is used to capture and store supplemental data. This data is related to the incident such as the name of a relevant manager or department. You create the custom attributes that you need.

The additional data that is contained in custom attributes can be used for:

- Enabling a workflow

- Executing incident response actions
- Including in report metrics
- Enabling incident response teams to act faster on incidents
- Enabling increased remediation and report automation

To populate custom attributes with actual incident-related data, the Enforce Server uses a lookup plug-in. This plug-in looks up and retrieves the additional data from an external source such as an employee database.

See "Working with custom attributes and lookup plug-ins" on page 11.

The lookup plug-ins can also be used to populate the **Data Owner Name** and **Data Owner Email Address** fields for notification to individual custodians or data owners for remediation.

See "Creating and distributing aggregated incident reports to data owners" on page 26.

How lookup plug-ins are implemented

This section lists the high-level tasks that are needed to implement a lookup plug-in. Subsequent sections explain these steps in detail.

Table 1-1 How to retrieve custom attributes from an external source

Step	Description
Step 1	Decide what external data you want to extract and load into incidents as custom attributes.
Step 2	Identify the sources from which custom attribute data is to be obtained.
Step 3	Create a separate custom attribute for each individual piece of external data that you want to include in incident snapshots and reports. See "Working with custom attributes and lookup plug-ins" on page 11. See "Creating custom attributes" on page 13.
Step 4	If using the Script Lookup Plug-In, write the scripts you need to extract data and populate the custom attributes of each incident.
Step 5	Determine which lookup parameter groups include the specific lookup parameters you need to extract the relevant data from the external sources. See "About lookup parameters" on page 14.
Step 6	Enable and configure the plug-in on the Enforce Server. See "How lookup plug-ins are enabled" on page 18.

Table 1-1 How to retrieve custom attributes from an external source *(continued)*

Step	Description
Step 7	Configure the plug-in to extract data from the external data source and populate the custom attributes. See "CSV Lookup Plug-In configuration" on page 34. See "LDAP Lookup Plug-In configuration" on page 39. See "About Script Lookup Plug-Ins" on page 43.

Working with custom attributes and lookup plug-ins

"Custom attributes" are the data fields that provide a way to capture and store supplemental incident information. You create the custom attributes that you need.

See "About custom attributes and lookup plug-ins" on page 9.

See "Creating custom attributes" on page 13.

When an incident is created, the Enforce Server retrieves data regarding that incident. The Enforce Server exposes some of this incident-related data to the Lookup API in the form of "attribute lookup parameters" (lookup parameters).

See "About lookup parameters" on page 14.

The Lookup API invokes one or more lookup plug-ins. These plug-ins can interface with one or more external data sources to extract the additional information that is related to the incident. Using the lookup parameters, the Lookup API then populates the incident's custom attribute fields with that additional information.

See "How custom attributes are populated" on page 12.

For example, `Department Name` and `Manager Email` are common custom attributes. The `Manager Email` custom attribute might be used in a response rule to notify an employee's manager of an incident that involves the employee. In this example, the email notification can be done automatically, or through a manual Smart Response. The `Department Name` custom attribute might be used for role-based access, to ensure First Responders only review incidents for their own department. Or it might be used to inform department heads of the policies their employee violate.

Symantec Data Loss Prevention provides three types of lookup plug-in:

- CSV Lookup Plug-In. Enables the extraction of pertinent data from a comma-separated values (CSV) file.

See “About the CSV Lookup Plug-In” on page 31.

- Live LDAP Lookup Plug-In. Enables the extraction of pertinent data from a live LDAP system. For example, Microsoft Active Directory, Novell LDAP, Sun LDAP, or IBM LDAP.
  See “About the LDAP Lookup Plug-In” on page 37.
- Script Lookup Plug-In. Enables you to write a custom script in scripting languages such as Perl or Python to extract the pertinent data. Scripts can extract data from sources such as proxy log files, or DNS systems.
  See “About Script Lookup Plug-Ins” on page 43.

In addition to the three plug-ins that Symantec provides, you can use the Lookup API to create your own custom lookup plug-ins.

See “About custom lookup plug-ins” on page 24.

All plug-ins receive a reference to the same `java.util.Map` object. This reference enables a plug-in to use the data that retrieved by a previous plug-in as lookup parameters for the next plug-in.

How custom attributes are populated

For each incident, custom attributes can be populated (their values can be set in the incident data) in the following ways:

- Automatically when the incident is detected by means of a lookup plug-in, as described in this guide
- Automatically when the incident is detected by means of an automated response rule
- Automatically when a user executes a Smart Response Rule
- Manually (though data entry) by specific users after detection

Custom attributes can also be re-populated automatically by clicking on the **Lookup** option in the **Attribute** section of the **Incident Snapshot** screen. This action replaces the existing values that are stored in the custom attribute fields with the values returned by the new lookup.

Note: If the new lookup returns null or empty values for any custom attribute fields, those empty values overwrite the existing values.

See “Working with custom attributes and lookup plug-ins” on page 11.

Creating custom attributes

Custom attributes can be grouped into attribute groups, similar to how statuses are grouped into status groups, to organize the information in a useful way. Examples of common attribute groups include **Employee Information**, **Manager Information**, and **Remediation Information**. All custom attributes are available for all incidents.

To create custom attributes and add them to a group

1. On the Enforce Server, click **System > Incident Data > Attributes > Custom Attributes**. Note that a number of custom attributes were defined and loaded for you by the Solution Pack that you selected during installation. All existing custom attributes are listed in the **Custom Attributes** window.
2. To create a new custom attribute, click the **Add** option.
3. Type a name for the attribute in the **Name** box. If appropriate, check the **Is Email Address** box.
4. Select an attribute group from the **Attribute Group** drop-down list. If necessary, create a new attribute group. Select **Create New Attribute Group** from the drop-down list, and type the new group name in the text box that appears.
5. Click the **Save** option.
6. Generate a new incident, or view an existing incident, and verify that it contains the new custom attribute.

The name you give to a custom attribute does not matter. But a custom attribute you create must be structured the same as the corresponding external data source. For example, suppose an external source stores department information as separate geographic location and department name. In this case, you must create corresponding location and department name custom attributes. You cannot create a single department ID custom attribute combining both the location and the department name.

Once you define your custom attributes, they become available to every incident. Each incident receives its own set of custom attributes (though some name-value pairs may be empty depending on circumstances). The custom attribute values for an incident can be populated and changed independently of other incidents.

You can edit the custom attribute values if you have been assigned to a role that includes edit access for custom attributes. If you want to update a group of incidents, you can select those incidents on the incident list page. You can then select the **Set Attributes** command from the **Incident Actions** menu. You can select **Lookup Attributes**, to look up the values of custom attributes. Note that

the **Set Attributes** command and **Attributes** section on the **Incident Snapshot** page are available only if at least one custom attribute is defined.

See "Working with custom attributes and lookup plug-ins" on page 11.

About lookup parameters

Attribute lookup parameters are data values inherently related to an incident. When Symantec Data Loss Prevention detects a policy violation and creates an incident, certain incident-related lookup parameters are automatically captured. You can use one or more of these lookup parameters as a key to retrieve additional information from external sources. This additional information is used to populate the custom attribute fields for each incident. For example, an email-related incident can capture the `sender-email` parameter, which contains the email address of the message sender. This email address can then be used as a key to look up that person in a corporate directory or some other data source. The person's phone number, department, and manager's name can then be extracted from that source and used to populate the custom attributes for that incident.

See "Working with custom attributes and lookup plug-ins" on page 11.

Different lookup parameters can be captured for each incident. For administrative convenience, lookup parameters are organized into groups. A given incident captures all of the attributes in all of the lookup parameter groups that have been enabled. However, some of the name-value pairs may be blank depending on the circumstances of the incident. For example, the `sender-email` attribute of a group might be null for a storage incident.

When you enable a lookup plug-in, you specify the lookup parameters to use as keys to extract the external data.

See "How lookup plug-ins are enabled" on page 18.

In the following table, *N* represents the number of multiple instances of an attribute; for example, *attachment-name*1, *attachment-name*2. Note that a numeral is used at the end of an attribute name to distinguish between multiple instances of that attribute.

Table 1-2 Lookup Parameters and Parameter Groups

Lookup parameters	Used for	Description and comments
Lookup Parameter Group: Attachment		
`attachment-name`*N*		Name of the attached file.
`attachment-size`*N*		Original size of the attached file.

Table 1-2 Lookup Parameters and Parameter Groups *(continued)*

Lookup parameters	Used for	Description and comments
Lookup Parameter Group: Incident		
`date-detected`	Network (protocols other than SMTP), Storage, Endpoint	Date and time when the incident was detected.
`incident-id`		The incident ID assigned by Enforce Server. The same ID can be seen in the incident report.
`protocol`	Network (protocols other than SMTP)	The name of the network protocol that was used to transfer the violating message, such as SMTP, HTTP, and so forth.
`data-owner-name`		The person responsible for remediating the incident. This field must be set manually, or with one of the lookup plug-ins. Reports based on this field can automatically be sent to the data owner for remediation. See "Creating and distributing aggregated incident reports to data owners" on page 26.
`data-owner-email`		The email address of the person responsible for remediating the incident. This field must be set manually, or with one of the lookup plug-ins.
Lookup Parameter Group: Message		
`date-sent`	Network (protocols other than SMTP)	Date and time when the message was sent.
`subject`	Network	Subject of the message if it is an email incident.
`file-create-date`	Storage	Date that the file was created in its current location, whether it was originally created there, or copied from another location. Retrieved from the operating system.
`file-access-date`	Storage	Date that the file was examined.
`file-created-by`	Endpoint	User who placed file on endpoint computer.
`file-modified-by`	Endpoint	Fully-qualified user credential for the computer where the violating copy action took place.
`file-owner`	Storage, Endpoint	The name of the user or the computer where the violating file is located.

Table 1-2 Lookup Parameters and Parameter Groups *(continued)*

Lookup parameters	Used for	Description and comments
discover-content-root-path	Storage	Root of path of the file which caused a Discover incident.
discover-location	Storage	Full path of the file that caused a Discover incident.
discover-name	Storage	The name of the violating file.
discover-extraction-date	Storage	(Seldom used or populated.)
discover-server	Storage	The name of repository to be scanned.
discover-notes-database	Storage	(Seldom used or populated.)
discover-notes-url	Storage	(Seldom used or populated.)
endpoint-volume-name	Endpoint	(Seldom used or populated.)
endpoint-dos-volume-name	Endpoint	(Seldom used or populated.)
endpoint-application-name	Endpoint	Name of application most recently used to open (or create) the violating file.
endpoint-application-path	Endpoint	Path of the application that was used to create or open the violating file.
endpoint-file-name	Endpoint	The name of the violating file.
endpoint-file-path	Endpoint	Location the file was copied to.
Lookup Parameter Group: Policy		
policy-name	Storage, Network, Endpoint	The name of the policy that was violated.
Lookup Parameter Group: Recipient		
recipient-email*N*	Network	The email address of the recipient, if applicable.
recipient-ip*N*	Network	The IP address of the recipient, if applicable.
recipient-url*N*	Network	The URL of the recipient, if applicable.
Lookup Parameter Group: Sender		
sender-email	Network (SMTP)	The email address of the sender, if applicable.
sender-ip	Endpoint and Network (protocols other than SMTP)	The IP address of the sender, if known.

Table 1-2 Lookup Parameters and Parameter Groups *(continued)*

Lookup parameters	Used for	Description and comments
`sender-port`	Network (protocols other than SMTP)	The port of the sender, if known.
`endpoint-user-name`	Endpoint	The user who was logged into the computer when the violation occurred.
`endpoint-machine-name`	Endpoint	Name of the computer where the violating file resides.
Lookup Parameter Group: Server		
`server-name`	Storage, Network, Endpoint	The name of the Enforce Server that detected the incident.
Lookup Parameter Group: Monitor		
`monitor-name`		(Seldom used or populated.)
`monitor-host`		(Seldom used or populated.)
`monitor-id`		(Seldom used or populated.)
Lookup Parameter Group: Status		
`incident-status`	Storage, Network, Endpoint	Current status of the incident.
Lookup Parameter Group: ACL		
`acl-principal`*`N`*	Storage	A string that indicates the user or group to whom the ACL applies.
`acl-type`*`N`*	Storage	A string that indicates whether the ACL applies to the file or to the share.
`acl-grant-or-deny`*`N`*	Storage	A string that indicates whether the ACL grants or denies the permission.
`acl-permission`*`N`*	Storage	A string that indicates whether the ACL denotes read or write access.

Separate Enforce Server database queries are required to retrieve each attribute group. These queries are executed for each incident before the lookup. To avoid the performance effect of unnecessary database queries, enable only those attribute groups that your installed lookup plug-ins require.

How lookup plug-ins are enabled

This section identifies the tasks that are needed to enable a lookup plug-in. Subsequent sections explain these tasks in detail. These tasks do not need to be performed in a set sequence.

Table 1-3 How to enable and configure lookup plug-ins

Task	Description	Reference
Decide on the lookup parameters to be used to extract data from the external sources.		See "About lookup parameters" on page 14.
Designate the plug-in (or plug-ins) you want to use.	Edit the `com.vontu.api.incident.attributes.AttributeLookup.plugins` property of the `\Protect\config\Plugins.properties` file on the Enforce Server. The `Plugins.properties` file is a plain ASCII text file that can be edited with any text editor, such as Notepad (Windows) or vi (Linux).	See "Plug-in designation in the AttributeLookup.plugins property" on page 19.
Specify the order in which plug-ins should be executed.	Edit the `com.vontu.plugins.execution.chain` property of the `Plugins.properties` file. If you use only a single plug-in, that plug-in has to be listed in the `com.vontu.plugins.execution.chain` property. If you use multiple plug-ins, the order they are listed in this property determines their order of execution.	See "Plug-in execution order in the plugins.execution.chain property" on page 20.
Configure the plug-in.	Modify the properties file for that type of plug-in (CSV, LDAP, Script). In the `keys` property, specify the lookup parameter to be used as the key to correctly populate the custom attribute fields from an external data source. Also, specify the mapping for all the lookup parameters.	See "About the CSV Lookup Plug-In" on page 31. See "About the LDAP Lookup Plug-In" on page 37. See "About Script Lookup Plug-Ins" on page 43.
Specify the name and location of the corresponding properties file for the plug-in.	Edit the `Plugins.properties` file and specify the properties file name for that type of plug-in (CSV, LDAP, Script).	See "Plug-in configuration file specification" on page 21.

Table 1-3 How to enable and configure lookup plug-ins *(continued)*

Task	Description	Reference
Specify the lookup parameter groups to be used as keys to correctly populate the custom attribute fields from an external data source.	List the parameters to be passed to all plug-ins by listing the lookup parameter groups in the `com.vontu.api.incident.attributes.AttributeLookup.parameters` property in the `Plugins.properties` file.	See "About lookup parameters" on page 14. See "Lookup parameter specification" on page 23.
Specify the output parameters or attributes written into the incident data.	List the parameters to be written into the incident data in the `com.vontu.api.incident.attributes.AttributeLookup.output.parameters` property in the `Plugins.properties` file.	See "About lookup parameters" on page 14.
Modify optional parameters if needed.	(Optional)	See "Optional properties" on page 23.
Restart the Vontu services.	The Vontu Manager service must be restarted for changes to the `Plugins.properties` file to take effect. The Incident Persister service must be restarted for the automatic attribute lookup to populate the incidents as they are created.	

Plug-in designation in the AttributeLookup.plugins property

To enable a lookup plug-in, copy the Specification-Title attribute from the JAR manifest of the plug-in. Then paste it in after the equals sign of the `com.vontu.api.incident.attributes.AttributeLookup.plugins` property in the `Plugins.properties` file.

For example, to load the Script Lookup Plug-In, the entry would look like:

```
com.vontu.api.incident.attributes.AttributeLookup.plugins=
     Vontu Script Lookup
```

To specify more than one plug-in, copy each Specification-Title attribute on the same line. Separate the attributes by commas with no spaces. For example, to specify the three Symantec-supplied plug-ins, enter:

```
com.vontu.api.incident.attributes.AttributeLookup.plugins=
     Vontu Script Lookup,
     Vontu Csv Lookup,
     Vontu Live LDAP Lookup
```

(The entry must be on a single line, with no spaces.)

It does not matter what order multiple plug-ins are listed in the `AttributeLookup.plugins` property. List each plug-in only once.

Note: Typing in the Specification-Title attribute from the JAR manifest name may not work. Symantec recommends copying the attribute from the JAR manifest into the properties file.

See "How lookup plug-ins are enabled" on page 18.

Plug-in execution order in the plugins.execution.chain property

Plug-in execution order (sequence) is specified in the `Plugins.properties` file. Enter each plug-in after the equals sign of the `com.vontu.plugins.execution.chain` property. Their sequence determines the order of execution.

The following rules apply:

- If you only use a single plug-in, it must still be listed in this property.
- When listing multiple plug-ins, list them separated by commas on the same line with no spaces. List them in the order they are to be executed. By listing multiple plug-ins in sequence you create a plug-in chain that makes the output of one plug-in available to the next plug-in.
  See "About multiple lookup plug-in chains" on page 21.
- The Script Lookup Plug-In can be listed multiple times for a given incident. The scripts that the Script Lookup Plug-In runs, and the order in which they are run, are specified in the `ScriptLookup.properties` file.

For example, to run the Script Lookup Plug-In, LDAP Lookup Plug-In, Script Lookup Plug-In again, CSV Lookup Plug-In, and the Script Lookup Plug-In a third time, enter:

```
com.vontu.plugins.execution.chain=com.vontu.lookup.script.ScriptLookup,
     com.vontu.lookup.liveldap.LiveLdapLookup,
     com.vontu.lookup.script.ScriptLookup,
     com.vontu.lookup.csv.CsvLookup,
     com.vontu.lookup.script.ScriptLookup
```

(The entry must be on a single line, with no spaces.)

See "How lookup plug-ins are enabled" on page 18.

Plug-in configuration file specification

For each plug-in that is listed in the `com.vontu.api.incident.attributes.AttributeLookup.plugins` property in the `Plugins.properties` file, you must specify a corresponding configuration file. These configuration files are stored in the `\Protect\config` directory. (The different configuration files are covered in more detail in the following chapters.) A different property is used for each type of lookup plug-in:

- CSV Lookup Plug-In. Specify the CSV Lookup Plug-In configuration file with the `com.vontu.lookup.csv.CsvLookup.properties` property. For example:

  ```
  com.vontu.lookup.csv.CsvLookup.properties=CsvLookup.properties
  ```

- LDAP Lookup Plug-In. Specify the LDAP Lookup Plug-In configuration file with the `com.vontu.lookup.liveldap.LiveLdapLookup.properties` property. For example:

```
com.vontu.lookup.liveldap.LiveLdapLookup.properties=LiveLdapLookup.properties
```

- Script Lookup Plug-In. Specify the Script Lookup Plug-In configuration file with the `com.vontu.lookup.script.ScriptLookup.properties` property. For example:

```
com.vontu.lookup.script.ScriptLookup.properties=ScriptLookup.properties
```

See "How lookup plug-ins are enabled" on page 18.

About multiple lookup plug-in chains

Lookup plug-ins can be chained together to execute in sequence. By chaining plug-ins you can use the output of one plug-in as input for the next. In a plug-in chain, the first plug-in uses the incident-related lookup parameters that are passed to it by the Enforce Server. The second plug-in uses data that is passed to it by the first plug-in including the lookup parameters and any temporary variables created. And so on for all the plug-ins in the chain.

A plug-in chain is useful when information must be pulled from different sources to populate the Custom Attributes for an incident. A chain is also useful when there are differences or dependencies between the "keys" needed to unlock the correct data.

For example, you can call:

1. The Script Lookup Plug-In to invoke a script that performs a DNS lookup.

2. The Script Lookup Plug-In again to invoke a second script that queries an asset management system.

3. The LDAP Lookup Plug-In to obtain data from a corporate directory.

Whether plug-in chaining is necessary to populate your custom attributes varies according to circumstances. For example:

- Getting the right key for Network email incidents is usually straightforward. The email address of the message sender is automatically captured as a lookup parameter. That lookup parameter can be used as a key to unlock the information about the sender that is stored in an external source. In this instance, it is not necessary to chain multiple plug-ins.
- For Web or FTP incidents, a plug-in chain might be necessary. The lookup parameter that is captured for these kinds of incidents is the IP addresses of the originating hosts. But IP addresses usually are not static identifiers like email addresses. Therefore, you may need to do successive lookups to get to a static identifier that can be used as an information key.

  For example, you can write a Script Lookup Plug-In to pass the `sender-ip` lookup parameter to a DNS server to get the host name. You can then write another script to pass that host name to an asset management system. From the asset management system you can obtain the user name or email of the person using that computer. That user name or email can then be used as the "key" to unlock the rest of the data. This plug-in chain would have three links:

  1. The Script Lookup Plug-In that uses the IP address to return the host name.
  2. The Script Lookup Plug-In that uses the host name to return the user name or email.
  3. The CSV Lookup Plug-In that uses the user name or email to return the rest of the custom attribute data.

  In this example, you must create a new `Host_Name` temporary variable to store the host name information. This temporary variable and its value are then available to the second script and subsequent plug-ins.

You specify which plug-ins should be loaded, in what order, in the `com.vontu.plugins.execution.chain` property in the `Plugins.properties` file on the Enforce Server.

See "How lookup plug-ins are enabled" on page 18.

See "Plug-in execution order in the plugins.execution.chain property" on page 20.

See "Chaining the Script Lookup Plug-In" on page 48.

Lookup parameter specification

For all the types of lookup plug-ins, you must specify the lookup parameters to be passed to the plug-in. The plug-in uses these lookup parameters as keys to extract the data from an external source and populate custom attributes of an incident.

Specify the lookup parameters to be passed to all plug-ins by listing the lookup parameter groups in the `com.vontu.api.incident.attributes.AttributeLookup.parameters` property in the `Plugins.properties` file. Note that you cannot specify individual lookup parameters, you can only specify attribute groups. The group names that can be listed are shown in the Lookup Parameters Table. To specify multiple attribute groups, list them separated by commas.

For example, to specify that the `sender`, `policy`, and `message` lookup parameter groups be passed to the plug-ins that you have enabled:

```
com.vontu.api.incident.attributes.AttributeLookup.parameters=sender,policy,message
```

To ensure optimum performance, only list the lookup parameter groups that contain the individual lookup parameters that you need for your plug-ins.

See "How lookup plug-ins are enabled" on page 18.

Optional properties

Other optional properties are in the `Plugins.properties` file. As a general rule, the default values of these properties do not need to be changed. But if it becomes necessary, you can modify the following parameters:

- The `Lookup timeout (in milliseconds)` property sets the timeout limit. The default is one minute.
- The `Automatic lookup` property determines whether lookups should be performed automatically as new incidents are received. The default is True.
- The `Maximum lookup thread count` property sets the maximum number of Lookups that can occur simultaneously. This property should be set to a value greater than the thread-count that is set in the new-incident-command configuration. The default value of 5 is usually sufficient unless traffic volume is very high.
- The `Automatic plugin reload` property sets whether plug-ins should be automatically reloaded at 3:00 each morning. Automatic reloads are performed to reflect any changes that may occur in outside data sources such as corporate directories. The default is True.

See the comments in the `Plugins.properties` file for information modifying these parameters.

See "How lookup plug-ins are enabled" on page 18.

About custom lookup plug-ins

In addition to the plug-ins that are included, you can use the Lookup API to create custom lookup plug-ins.

See "About custom attributes and lookup plug-ins" on page 9.

You package lookup plug-ins in one or more JAR files. You then place the JAR files in the `\Protect\plugins` directory on the Enforce Server, along with all the JAR files on which they depend. For example, if your custom plug-in requires CIFS network share access through Java, you must copy the `jcifs.jar` file to the `\plugins` directory.

Each JAR file must contain only one plug-in. If there is more than one plug-in in a particular JAR file, it is impossible to specify their order of execution.

A new plug-in implementation that you create should result in an `AttributeLookupException` when the lookup fails for some reason. This exception is not considered fatal and does not cause the removal of the plug-in. The exception is logged as a warning in the Tomcat log and in the Symantec system events.

Most plug-ins require configuration in the form of property files. It is determined within each plug-in how it locates its property files. Symantec makes the following lookup property files available for use:

- `CsvLookup.properties`
- `LiveLdapLookup.properties`
- `ScriptLookup.properties`

One way of pointing to the property file is to use a system property and set its value in the `Plugins.property` file. All properties that are loaded from this file are loaded into system properties and are available to all lookup plug-ins. All relative paths are resolved relative to the `Plugins.property` file location.

See "Custom plug-in configuration" on page 25.

For example, the `com.my_company.MyPlugin.config=MyPlugin.properties` property in the `Plugins.properties` file can be read from within the plug-in in the following manner:

```
File propertyFile = new
File(System.getProperty("com.my_company.MyPlugin.config"));
```

In this case, `MyPlugin.properties` is expected to be in the configuration folder.

The best practice is for each plug-in to use its own property files and to use `Plugins.properties` only to point to the location of such files.

Custom plug-in configuration

You list all custom plug-in dependencies in the `com.vontu.api.incident.attributes.AttributeLookup.plugins` property of the `\Plugins.properties`. The order in which the dependencies are listed does not matter.

See "About custom lookup plug-ins" on page 24.

You can specify a plug-in JAR file using either the Specification-Title value from the manifest, or the file name. For example, the following line specifies two JAR files, `ldapjdk.jar` and a JAR file with the Specification-Title "Vontu DLP":

```
com.vontu.api.incident.attributes.AttributeLookup.plugins=ldapjdk.jar,Vontu DLP
```

Note that if several JAR files have the same Specification-Title, they are all loaded, and you cannot define their execution order. In this situation you must specify their file names to establish execution order.

You must restart the Vontu Manager service for any changes to the plug-in JAR files or the `\Plugins.properties` file to take effect.

The Incident Persister service must be restarted for the automatic attribute lookup to populate the incidents as they are created.

Configuring the lookup plug-ins to populate the Data Owner fields

You can configure the lookup plug-ins to populate the **Data Owner Name** and **Data Owner Email Address** attributes in the incident data.

For example, you can perform an LDAP lookup of the **Data Owner Email Address** after you have populated the **Data Owner Name**.

To configure the lookup plug-ins to populate the Data Owner fields

1. Configure the selected lookup plug-ins.

   See "How lookup plug-ins are enabled" on page 18.

   To configure the Data Insight plug-in, see the *Symantec Data Loss Prevention Data Insight Implementation Guide*.

2. To write to a Data Owner field, specify the attribute in the `com.vontu.api.incident.attributes.AttributeLookup.output.parameters` property in the `Plugins.properties` file.

   See "About lookup parameters" on page 14.

   For example:

   ```
   com.vontu.api.incident.attributes.AttributeLookup.output.parameters=
   data-owner-email
   ```

Creating and distributing aggregated incident reports to data owners

You can create and automatically distribute aggregated incident reports to data owners for remediation.

An automatic workflow can be set up for the following use cases:

- Automatically or manually set the **Data Owner Name** and **Data Owner Email Address** for new incidents.
- Set a custom status value or custom attribute to mark that the **Data Owner Name** for an incident has been verified. Custom attributes and custom status values can also mark incidents for other workflow steps.
- Set up a recurring email schedule.
  Reports can be configured to be sent on a recurring schedule, sending only the incidents that have not yet been distributed.
- Mark the incident as sent.
  After the report is sent, the status attributes and custom attributes can optionally be set, to flag the incidents for the next stage of the workflow.
- Automate the tasks.
  Lookup plug-in scripts and chained lookup plug-ins can automate the tasks in the workflow sequence.

The following process describes a complex use case that includes the setup tasks, and suggestions to automate some steps in the process.

Table 1-4 Setting up, creating, and distributing aggregated incident reports to data owners

Step	Action	Description
Step 1	Install and set up the Symantec Data Insight Management Server. Make sure that the Symantec Data Insight Management Server has access to the files or file systems of interest.	See the following Symantec Data Insight documentation: ■ *Symantec Data Insight Installation Guide* ■ *Symantec Data Insight Administrator's Guide*
Step 2	Install the Symantec Data Loss Prevention product, including at least one Network Discover Server.	See the *Symantec Data Loss Prevention Installation Guide*.
Step 3	Set up the connection between the Enforce Server and the Symantec Data Insight Management Server.	**Note:** Symantec Data Insight is a separately licensed option. If Symantec Data Insight is not licensed on the Enforce Server, the menu option to configure the connection to the Symantec Data Insight Management Server does not appear.
Step 4	Test the connection from the Enforce Server to the Symantec Data Insight Management Server.	
Step 5	On the Enforce Server, create a custom status value or custom attribute for the Data Owner Name verification, and any workflow status attributes.	
Step 6	Map the details from the Symantec Data Insight Management Server into the custom attributes that you created.	Edit the properties file for Symantec Data Insight on the Enforce Server, to map the details from the Symantec Data Insight Management Server into the custom attributes that you created.
Step 7	Map any of the Symantec Data Insight attributes directly into the **Data Owner Name** field.	To map the Symantec Data Insight data user (the person who uses the file most frequently) to the **Data Owner Name**, set the `Data_User` parameter.

Table 1-4 Setting up, creating, and distributing aggregated incident reports to data owners *(continued)*

Step	Action	Description
Step 8	Set up all your lookup plug-ins. For example, you may want to chain the LDAP Lookup Plug-In to take the **Data Owner Name** and set the **Data Owner Email Address** as either the data owner or the manager of the data owner. No built-in capability provides consistency between the data owner and email address. This action must be customized. The **Data Owner Email Address** can have multiple email addresses that are separated with commas. **Note:** If duplicate attribute names exist between these names and custom attributes, then both fields are updated.	Edit the `Plugins.properties` file to set up all your lookup plug-ins. See "About multiple lookup plug-in chains" on page 21. See the *Symantec Data Loss Prevention Lookup Plug-in Guide*.
Step 9	Verify that the Enforce Server general settings are set up to send email notifications.	Set up the SMTP notification settings. Set the option **Send report data with emails**.
Step 10	Verify that the incident responder has the privileges to run the reports.	The **Remediate Incidents** privilege is required to configure and run the reports. The **Lookup Attributes** privilege is required to set attributes from the lookup plug-ins. The User Privilege **CSV Attachment in Email Reports** is required to attach the CSV report to the email.
Step 11	Set up a Network Discover and run a sample scan of the file systems of interest.	

Table 1-4 Setting up, creating, and distributing aggregated incident reports to data owners *(continued)*

Step	Action	Description
Step 12	Set up any custom reports.	Set up a filtered report, or set up any report that you want to distribute. For example, you can filter based on the new incidents. Select the option **Change Incident Status / Attributes** of the reports scheduling to set incident status or attributes when the email is sent. You can also manually set the custom attribute that indicates these incidents were verified. Select any or all incidents in the list. Use the drop-down **Incident Actions** and select **Set Attributes**. You can also set a custom status from this drop-down menu.
Step 13	Save the custom reports and set up a distribution schedule.	

Chapter 2

Using the CSV Lookup Plug-In

This chapter includes the following topics:

About the CSV Lookup Plug-In

The CSV Lookup Plug-In extracts data from a delimited comma-separated values (CSV) file stored on the Enforce Server. The plug-in uses that data to populate custom attributes for an incident at the time the incident is generated.

The process works as follows: The CSV Lookup Plug-In receives a group of lookup parameters that contain data about an incident from the Enforce Server. One or more of the lookup parameters in the group are mapped to column heads in a CSV file. For example, the `sender-email` lookup parameter might be mapped to the `Email` column in the CSV file. The value in the lookup parameter is used as a key to find a matching value in the corresponding CSV column. When a match is found, the CSV row that contains the matching value provides the data that is returned to the Enforce Server. The Enforce Server uses the data in that row to populate the custom attributes for that incident. For example, if the `sender-email` lookup parameter contains the value `mary.smith@mycompany.com`, the plug-in searches the `Email` column for a row that contains `mary.smith@mycompany.com`. That row is then used to provide the data to populate the custom attributes for the incident.

Note that when multiple plug-ins are chained together, input from a previous plug-in can be similarly used as a key.

See "About multiple lookup plug-in chains" on page 21.

The CSV Lookup Plug-In must meet certain requirements before it can be used.

See "CSV Lookup Plug-In requirements" on page 33.

CSV Lookup Plug-In and CSV data files

The CSV Lookup Plug-In requires a CSV file that is stored on the Enforce Server. You can use tab, pipe, or comma characters to delimit the CSV file. The recommended practice is to use the pipe character ("|") as the delimiter. A CSV file can be stored anywhere on the Enforce Server. You specify the CSV file location in the `CsvLookup.properties` file. The recommended practice is to place it in the `\Protect\plugins` directory.

See "About the CSV Lookup Plug-In" on page 31.

When creating a CSV file, keep in mind the following requirements:

- The first data row of the CSV file must contain column headers.
- Column header fields cannot be blank.
- Make sure that there are no white spaces at the end of the column header fields.
- Make sure that all rows have the same number of columns.
- Each row of the file must be on a single, non-breaking line.
- One or more columns in the file are used as key-fields for data lookups. You specify in the file `CsvLookup.properties` which column heads are to be used as key fields. You also specify the key field search order. Common key fields typically include email address, Domain\UserName (for Endpoint incidents), and user name (for Storage incidents).
- The data values in the key field columns must be unique. If a single column is the key field, none of the values in that column can be blank. If multiple columns are used as key fields (for example, `EMP_EMAIL` and `USER_NAME`), the combination of values in each row must be unique. When multiple columns are used as key fields, at least one of the fields in each row must contain non-blank data.
- Fields in data rows (other than the column header row) can be empty, but at least one key field in each row must contain data.

- If the CSV file uses a character set that is not the Enforce Server default, the `csv_file_charset` property must be set in the `CsvLookup.properties` file.
- The recommended CSV delimiters (separators) are pipe, comma, or a tab. The same delimiter must be used for all values in the column header and data rows. The delimiter must be designated in the `delimiter` property of the `CsvLookup.properties` file.
- For Discover scan incidents, the `file-owner lookup` parameter does not include a domain. To use `file-owner` as the key, the CSV file column that corresponds to `file-owner` should be in the format *owner*. The format *DOMAIN\owner* does not result in a successful lookup. This restriction only applies to Discover incidents, other kinds of incidents can include a domain.

For example, the column-header row and a data-row of a pipe-delimited CSV file might look like:

```
email|first_name|last_name|domain_user_name|user_name|department|manager|manager_email
jsmith@acme.com|John|Smith|CORP\jsmith1|jsmith1|Accounting|Mei Wong|mwong@acme.com
```

If more than 10% of the rows in the CSV file violate any of these requirements, the plug-in does not load.

For accuracy in the lookup, the CSV file needs to be kept up to date.

CSV Lookup Plug-In requirements

The CSV Lookup Plug-In must meet certain requirements before it can be used.

- The CSV file must be created and placed on the Enforce Server.
  See "CSV Lookup Plug-In and CSV data files" on page 32.
- The CSV Lookup Plug-In must be enabled on the Enforce Server.
  See "How lookup plug-ins are enabled" on page 18.
- The CSV Lookup Plug-In must be configured in the `CsvLookup.properties` file.
  See "CSV Lookup Plug-In configuration" on page 34.

To use the CSV Lookup Plug-In, the following files must exist on the Enforce Server:

- `\Protect\Config\Plugins.Properties`
- `\Protect\Config\CsvLookup.properties`
- `\Protect\plugins\csv-lookup.jar`

These three files are included in a normal installation of Symantec Data Loss Prevention software.

CSV Lookup Plug-In configuration

To configure the CSV Lookup Plug-In:

- Specify the name and location of the CSV file that contains the data to be used to populate custom attributes for an incident.
  See "CSV file name and location specification for the CSV Lookup Plug-In" on page 34.
- Specify the delimiter that is used in the CSV file.
  See "CSV file delimiter specification for the CSV Lookup Plug-In" on page 34.
- Map the system and the custom attributes to the CSV file column heads.
  See "CSV file column head to attribute mapping for the CSV Lookup Plug-In" on page 35.
- Define the keys which are used to extract custom attribute data.
  See "Key mapping for the CSV Lookup Plug-In" on page 36.

These tasks are performed by editing the `CsvLookup.properties` file.

CSV file name and location specification for the CSV Lookup Plug-In

Use the `csv_file_path` property of the `CsvLookup.properties` file to specify the name and location of the CSV file. For example:

```
csv_file_path = ../plugins/employees.csv
```

You can enter either an absolute file path or a relative file path to the directory where the `CsvLookup.properties` file resides.

Note: You must use forward-slashes (/) instead of the backslashes (\) in the file path regardless of platform. So on a Windows platform, an absolute path would look like: `C:/Vontu/Protect/plugins/employees.csv`.

See "CSV Lookup Plug-In configuration" on page 34.

CSV file delimiter specification for the CSV Lookup Plug-In

Use the `delimiter` property of the `CsvLookup.properties` file to specify the CSV file delimiter. For example, to specify the pipe character as the delimiter:

```
delimiter = |
```

While commas, pipes, and tabs are all supported as delimiters, using the pipe is recommended. Use of a comma as a delimiter is discouraged because commas are often included in data fields as part of the data. For example, a street address might contain a comma.

To specify a tab character as the delimiter: `delimiter = \t`.

See "CSV Lookup Plug-In configuration" on page 34.

CSV file column head to attribute mapping for the CSV Lookup Plug-In

CSV data file column heads are mapped to attributes in the `CsvLookup.properties` file using the format: `attr.`*`attribute_name=column_head`*. Each attribute-column head pair is mapped on a separate line of the file. For example, to map the `sender-email` lookup parameter to the `Email` column head, you would enter a line in the `CsvLookup.properties` file:

```
attr.sender-email = Email
```

If an attribute name contains a space or a tab, you must prepend each instance of the white-space character with a backslash. For example, to map the `Employee Email` custom attribute to the `emp_email` column head, you would enter:

```
attr.Employee\ Email = Emp_email
```

Note that these names are case sensitive.

A set of four lookup parameters to column head mappings, followed by nine custom attributes to column head mappings:

```
attr.sender-email = Email
attr.endpoint-user-name = Username
attr.file-owner = File-owner
attr.sender-ip = IP

attr.First\ Name = FIRST_NAME
attr.Last\ Name = LAST_NAME
attr.Business\ Unit = Org
attr.Manager\ Email = Mgr_email
attr.Employee\ ID = EMPLOYEE_NUMBER
attr.Phone\ Number = Phone
attr.Manager\ Last\ Name = Mgr_lastname
attr.Manager\ First\ Name = Mgr_firstname
attr.Employee\ Email = Emp_email
```

See "CSV Lookup Plug-In configuration" on page 34.

Key mapping for the CSV Lookup Plug-In

Use the `keys` property of the `CsvLookup.properties` file to identify and sequence the keys that are used to extract custom attribute data. Separate multiple keys with a colon. For example, to specify the `Email`, `Username`, `File-owner`, and `IP` keys enter:

```
keys = Email:Username:File-owner:IP
```

The order in which you list the keys determines the search sequence. In the previous example: the plug-in first searches the `Email` column for a value that matches the lookup parameter value of the`sender-email` which has been passed to the plug-in. If no matching value is found, the plug-in then searches the `Username` column for a value that matches the `endpoint-user-name` lookup parameter. If no matching value is found in that column, it then goes on to search the next key (`File-owner`), and so on.

Note: The plug-in searches the keys in order until it finds the first matching value. When a matching value is located, the plug-in stops searching. The plug-in uses the data in the row that contains the first matching value to populate the relevant custom attributes. Therefore, key values are not used in combination, but rather the first value that is found becomes the key. Because the plug-in stops searching after it finds the first matching value, the order in which you list the `keys` column heads is significant.

CSV file character set specification for the CSV Lookup Plug-In

If the CSV file uses a character set that is not the default for the Enforce Server, the `csv_file_charset` property must be set in the `CsvLookup.properties` file. By default, this property is commented out (not enabled) and the CSV file is read using the default character set of the operating system. In cases where the CSV file uses a character set other than the default, specify it in the format: `csv_file_charset = name`, where *name* identifies a character set supported by Java 1.6. For example, `csv_file_charset = UTF-8`

Supported character sets and their naming conventions are described at Class Charset.

See "CSV Lookup Plug-In configuration" on page 34.

Chapter 3

Using the LDAP Lookup Plug-In

This chapter includes the following topics:

About the LDAP Lookup Plug-In

The LDAP Lookup Plug-In pulls data from a live LDAP system (such as Microsoft Active Directory, Novell LDAP, Sun LDAP, or IBM LDAP). It then uses that data to populate custom attributes for an incident at the time the incident is generated.

The process works as follows: The LDAP Lookup Plug-In receives a group of lookup parameters that contain data about an incident from the Enforce Server. These lookup parameters can be used in LDAP queries to pull data out of an existing LDAP directory. For example, the value of the`sender-email` lookup parameter might be compared to the values in the `email`attribute of the directory. If the `sender-email` lookup parameter contains `mary.smith@mycompany.com`, a query can be constructed to search for a record whose `email` attribute contains `mary.smith@mycompany.com`. Data in the record that the search returns is inserted into the custom attributes for the incident.

Note that when multiple plug-ins are chained together, output from a previous plug-in can be used as a key.

See “About multiple lookup plug-in chains” on page 21.

The LDAP Lookup Plug-In must meet certain requirements before it can be used.

See “LDAP Lookup Plug-In requirements” on page 38.

The connection to the LDAP server should be tested before you use the LDAP Lookup Plug-In.

See “ LDAP Lookup tests for the LDAP Lookup Plug-In” on page 42.

LDAP Lookup Plug-In requirements

To use the LDAP Lookup Plug-In, the following requirements must be met:

- A functioning connection to an LDAP server must be available.
  See “LDAP directory connection requirements for the LDAP Lookup Plug-In” on page 38.
- The LDAP Lookup Plug-In must be enabled on the Enforce Server.
  See “How lookup plug-ins are enabled” on page 18.
- The connection to the LDAP server must be configured in the LDAP Lookup Plug-In.
  See “LDAP server connection—LDAP Lookup Plug-In” on page 39.
- The different attributes must be mapped in the LDAP Lookup Plug-In.
  See “Attribute to LDAP data maps for the LDAP Plug-In” on page 40.

To use the LDAP Lookup Plug-In, the following files must exist on the Enforce Server:

- `\Protect\Config\LiveLdapLookup.properties`
- `\Protect\Config\Plugins.Properties`
- `\Protect\Config\liveldap-lookup.jar`

These three files are included in a normal installation of Symantec Data Loss Prevention software.

LDAP directory connection requirements for the LDAP Lookup Plug-In

The following conditions must be met for Symantec Data Loss Prevention to establish a connection with an LDAP directory:

- The LDAP directory must be running on a host that is accessible to the Enforce Server.

- There must be an LDAP account with that the Symantec Data Loss Prevention can use. This account must have read-only access.You must know the user name and password of the account.
- You must know the Fully Qualified Domain Name (FQN) of the LDAP server (the IP address cannot be used).
- You must know the port on the LDAP server which the Enforce Server uses to communicate with the LDAP server. The default is 389.

Use an LDAP lookup tool such as Softerra LDAP Browser to confirm that you have the correct credentials to connect to the LDAP server. Also confirm that you have the right fields defined to populate your custom attributes.

See "About the LDAP Lookup Plug-In" on page 37.

LDAP Lookup Plug-In configuration

To configure the LDAP Lookup Plug-In, you must perform two tasks:

- Configure the connection to the LDAP server.
  See "LDAP server connection—LDAP Lookup Plug-In" on page 39.
- Map your attributes to the corresponding LDAP directory fields.
  See "Attribute to LDAP data maps for the LDAP Plug-In" on page 40.

Both of these tasks are performed by editing the `\Protect\config\LiveLdapLookup.properties` file. This file is divided into two sections. The first section defines the connection parameters for the LDAP server to communicate with during the lookup. The second section defines the correlations between Symantec Data Loss Prevention attributes and LDAP attributes.

LDAP server connection—LDAP Lookup Plug-In

To configure the connection to the LDAP server, edit the following properties in the `LiveLdapLookup.properties` file:

- `servername` - The Fully Qualified Name (FQN) of the LDAP server. Do not use the IP address. For example, `servername = LDAPserver1.hr.corp`.
- `port` - The LDAP server port. The default is 389. For example,`port = 389`.
- `basedn` - The initial depth within the LDAP directory to start the search. When defining this parameter keep in mind that the closer to the information the query starts, the faster the response. For example, `basedn = DC=corp,DC=hr`.

- `authtype` - The authentication type for the LDAP server. The default is `simple`. For example, `authtype = simple`. The authentication type may need modification if connectivity cannot be established.
- `username` - The user name of the account which has access to the LDAP server. For example, `username = symantec_dlp`. The format of this line can vary depending on the account that has read access to the LDAP server. For example, a user name for the Microsoft Active Directory System might need to be specified in the format *`domain`*`\\`*`username`*. While the user name for a Sun LDAP server might be specified in the format `uid=`*`username`*`,ou=people,o=`*`company`*.
- `password` - The password for the user name that was specified in the previous field. For example,`password = Shazam!44`.

Note: These user name and password credentials are stored in clear text. If they are changed on the LDAP server and not updated in this properties file, the lookup fails.

See "About the LDAP Lookup Plug-In" on page 37.

Attribute to LDAP data maps for the LDAP Plug-In

System and custom attributes are mapped to LDAP data by entries in the `LiveLdapLookup.properties` file. Each mapping is entered on a separate line in the file. The order in which these mapping entries appear in the file does not matter.

A mapping is entered in the format:

```
attr.CustomAttributeName = search_base:
  (search_filter=$variable$):
  ldapAttribute
```

Where:

- *CustomAttributeName*
  The name of the custom attribute as it is defined in the Enforce Server.

  Note: If the name of the attribute contains white-space characters, you must precede each instance of the white space with a backslash. A white-space character is a space or a tab. For example, you need to enter the `Business Unit` custom attribute in the `LiveLdapLookup.properties` file as: `attr.Business\ Unit`

See "Creating custom attributes" on page 13.

- *search_base*
  Identifies the LDAP directory.
- *search_filter*
  The name of the LDAP attribute (field) that corresponds to the lookup parameter (or other variable) passed to the plug-in from the Enforce Server.
- *variable*
  The name of the lookup parameter that contains the value to be used as a key to locate the correct data in the LDAP directory. (In cases where multiple plug-ins are chained together, the parameter might be a variable that is passed to the LDAP Lookup Plug-In by a previous plug-in.)
- *ldapAttribute*
  The LDAP attribute whose data value is returned to the Enforce Server. This value is used to populate the custom attribute that is specified in the first element of the entry.

For example:

```
attr.First\ Name = dc=corp,dc=hr:(mail=$sender-email$):givenName
```

This example entry searches the `hr.corp` LDAP directory for a record with an attribute for `mail` whose value matches the value of the `sender-email` lookup parameter. It returns to the Enforce Server the value of the `givenName` attribute for that record.

A separate entry is made in the `LiveLdapLookup.properties` file for each custom attribute that is to be populated. For example:

```
attr.First\ Name = cn=users:(email=$sender-email$):firstName
attr.Last\ Name = cn=users:(email=$sender-email$):lastName
attr.TempDeptCode = cn=users:(email=$sender-email$):deptCode
attr.Department = cn=departments:(deptCode=$TempDeptCode$):name
attr.Manager = cn=users:(email=$sender-email$):manager
```

Note the use of the `TempDeptCode` temporary variable in the previous example. The department code is needed to obtain the department name from the LDAP hierarchy. But only the department name needs to be stored as a custom attribute. The `TempDeptCode` variable is created for this purpose.

See "About the LDAP Lookup Plug-In" on page 37.

LDAP Lookup tests for the LDAP Lookup Plug-In

Before using the LDAP Lookup Plug-In you should test the connection to the LDAP server.

To test the connection to the LDAP server

1. Save the credentials in the `LiveLdapLookup.properties` file.
2. To reload the LDAP Lookup Plug-In from the **Custom Attributes** tab of the **System > Incident Data > Attributes** screen, click **Reload Lookup Plug-ins**.
3. Look at an existing incident and click on the **Incident Actions** drop-down menu.
4. Click the **Lookup** option on the incident snapshot, and make sure that no connection errors are recorded in the **Incident History** section.

To verify that the custom attributes are correctly populated

1. Save the Custom Attribute definitions in the `LiveLdapLookup.properties` file.
2. To reload the LDAP Lookup Plug-In from the **Custom Attributes** tab of the **System > Incident Data > Attributes** screen, click **Reload Lookup Plug-ins**.
3. Look at an existing incident and click on the **Incident Actions** drop-down menu and then select the **Lookup Attribute** option.
4. When the page returns, select the **Attributes** tab from the **Incident Snapshot** page.
   - The Custom Attributes should be filled with entries retrieved from the LDAP lookup.
   - If the correct values are not populated, or there is no value in a custom attribute you have defined, check the `LiveLdapLookup.properties` file for errors. You can use a lookup tool such as the Softerra LDAP Browser to help confirm that you have the correct fields defined.

See "About the LDAP Lookup Plug-In" on page 37.

Chapter 4

Using the Script Lookup Plug-In

This chapter includes the following topics:

About Script Lookup Plug-Ins

The Script Lookup Plug-In lets you write custom scripts to query data repositories for values to populate custom attributes. For example, you can write a script that queries a DNS server for information about a sender that is involved in an incident. Symantec uses the output from such scripts to populate custom attributes in incident records.

The Script Lookup Plug-In does not use its properties file (`\Protect\config\ScriptLookup.properties`) to specify how lookup parameters are used as keys to extract data from the external source and populate the custom attributes. Instead, you write this functionality into each different script as needed.

The Script Lookup Plug-In works within the existing Lookup API. It supports any scripting language (for example, Perl or Python), as long as the script is written with a standard input and output format (`stdout`).

The Lookup API supports chaining the Script Lookup Plug-In with other plug-ins. You can execute multiple scripts and other plug-ins per incident in the order you specify. Among other things, chaining lets you use the output of one script as the input for a subsequent script.

The Credential Utility lets you encrypt any logon credentials (needed by your scripts) for querying a particular repository or data source.

See "Script Lookup Plug-In requirements" on page 44.

See "About logon credentials in scripts" on page 50.

See "Increasing log levels" on page 52.

See "About the scripts that exceed the timeout" on page 53.

Script Lookup Plug-In requirements

Script Lookup Plug-Ins must meet certain requirements before they can be used.

- A lookup script accessible to the Script Lookup Plug-In must be created.
  See "About writing and preparing scripts for the Script Lookup Plug-In" on page 45.
- The Script Plug-In must be enabled on the Enforce Server.
  See "How lookup plug-ins are enabled" on page 18.
- The Script Plug-In must be configured.
  See "Configuring the Script Lookup Plug-In" on page 46.

To use the Script Lookup Plug-In, the following files must exist on the Enforce Server:

- `\Protect\Config\Plugins.Properties`
- `\Protect\Config\ScriptLookup.properties`
- `\Protect\plugins\script-lookup.jar`

These three files are included in a normal installation of Symantec Data Loss Prevention.

About writing and preparing scripts for the Script Lookup Plug-In

To function with the Script Lookup Plug-In, script output must be in the required format. The script must reside in a directory on the Enforce Server host such as the `\Protect\plugins\scripts` directory. Or it can reside on a file share that is accessible with the appropriate permissions.

Note that scripts must exit with an exit code of '`0`.' If scripts exit with any other code, the Enforce Server assumes that an error has occurred in script execution and terminates the attribute lookup.

See "Script input and output for the Script Lookup Plug-In" on page 45.

See "Preparing scripts for use with the Script Lookup Plug-In" on page 46.

Script input and output for the Script Lookup Plug-In

This section describes calling conventions for the Script Lookup Plug-In, including how the plug-in passes arguments to scripts and the required format for script output.

When writing scripts for use with the Script Lookup Plug-In, note the following:

- The Script Lookup Plug-In passes custom attributes (which you define in the Enforce Server administration console) to scripts as command-line parameters in the form `key=value`.
- To work with the plug-in, scripts must output a set of key-value pairs to standard out (`stdout`). Newline characters must separate output key-value pairs. For example:

  ```
  host-name=mycomputer.company.corp
  username=DOMAIN\bsmith
  ```

  These key-value pairs are used to populate the custom attributes. Key-value pairs format is *`custom_attribute_name=custom_attribute_value`*.
- Make sure that scripts do not print out error or debugging information. For example, in a Python script, you can include a section similar to the following that redirects `stderr` to a file:

  ```
  fsock = open("C:\error.log", "a")
  sys.stderr = fsock
  ```

See "About writing and preparing scripts for the Script Lookup Plug-In" on page 45.

Preparing scripts for use with the Script Lookup Plug-In

After writing a script, you must copy it to the Enforce Server host and make sure that any necessary applications are installed on the host.

To prepare a script for use with the Script Lookup Plug-In

1. On the Enforce Server host, navigate to the `\Protect\plugins` directory and create a scripts subdirectory. For example, create `\Protect\plugins\scripts`.
2. Copy your script file to the scripts subdirectory.
3. Make sure that permissions are set correctly on the directory and the file. The directory and the file must be readable by the `protect` user. Depending on the scripting language, the script may also need to be executable by the `protect` user.
4. Download and install (on the Enforce Server host) any applications necessary for executing the script. For example, if you plan to run a Python script on a Windows system, download and install the latest version of Python. For the scripts that require language libraries, you must install the required files on the Enforce Server host.

See "About writing and preparing scripts for the Script Lookup Plug-In" on page 45.

Configuring the Script Lookup Plug-In

The Script Lookup Plug-In is configured by modifying the `ScriptLookup.properties` file, which is stored in the `\Protect\config` directory. (Note that the use of lookup parameters, temporary variables, or custom attributes as keys to extract data from the external source is usually specified in the script itself rather than in the `ScriptLookup.properties` file.)

To configure the ScriptLookup.properties file

1. Open the `ScriptLookup.properties` file in a text editor.
2. Edit the `script.1.command` key to reference the scripting language executable. For example, for a Python script on a Windows platform, edit the attribute to look similar to one of the following:

On Windows	`script.1.command=c:/python25/python.exe`
On Linux	`script.1.command=/usr/bin/python`

3 Edit the `script.1.custom.args` key to include the command to execute the script. For example, for a Python script named `ip-lookup.py`, edit the attribute to look similar to one of the following:

On Windows	`script.1.custom.args=-u,c:/Vontu/protect/plugins/scripts/ip-lookup.py`
On Linux	`script.1.custom.args=-u,/opt/Vontu/protect/plugins/scripts/ip-lookup.py`

4 Save the file.

5 Go to the **Services** panel and stop and restart the **Manager** service.

You can specify the incident types (by protocol) for passing attribute values to look up scripts.

See "Filtering by protocol" on page 47.

You can chain scripts together and chain scripts with other lookup plug-ins.

See "Chaining the Script Lookup Plug-In" on page 48.

Filtering by protocol

You can specify the incident types (by protocol) for passing attribute values to look up scripts. (The protocol values are displayed at the top of the incident snapshot.)

For example, you can limit the passing of attribute values to those incidents that are detected over HTTP. When you filter by protocol, Enforce Server still captures the incidents that are detected over other protocols. But it does not use the Script Lookup Plug-In to populate those incidents with custom attribute values.

To configure protocol filtering

1. In the `Plugins.properties` file (located by default in `\Protect\config`), enable the `incident` lookup parameter group. (The `incident` group includes an attribute for `protocol`, which specifies the protocols to include.) For example, to enable both the `sender` and `incident` groups, the entry would look like:

   ```
   com.vontu.api.incident.attributes.AttributeLookup.parameters=
       sender,incident
   ```

   Where `sender` and `incident` appear after the equal sign in the list of attributes. Separate the different lookup by commas. The order in which they are listed does not matter.

   See "About lookup parameters" on page 14.

2. Save the file.
3. In `ScriptLookup.properties` (located by default in `\Protect\config`), set the `protocol.filtering.enabled` attribute to `true`, as in this example:

   ```
   protocol.filtering.enabled=true
   ```

   Then specify the protocols for which to perform lookup, as in this example:

   ```
   protocols.allow=FTP,HTTP
   ```

   Specify the protocols you want to include in the lookup in the same format that they are displayed at the top-left of the incident snapshot. For example, to enable script lookup on emails, you specify:

   ```
   protocols.allow=Email/SMTP
   ```

4. Save the file.
5. Go to the **Services** panel and stop and restart the Manager service.

See "Configuring the Script Lookup Plug-In" on page 46.

Chaining the Script Lookup Plug-In

To set up a plug-in chain, configure the plug-in order in `Plugins.properties` and then configure the exact commands and arguments in the `ScriptLookup.properties` file.

To chain scripts and other plug-ins

1 List the plug-ins in the order in which Enforce Server should execute them in the `ScriptLookup.properties` file. Separate different entries with commas. By default, the `ScriptLookup.properties` file is located in `\Protect\config`.

For example, to call the Script Lookup Plug-In, then the LDAP Lookup Plug-In, then the Script Lookup Plug-In a second time, and then the Script Lookup Plug-In a third time, the entry looks like:

```
com.vontu.plugins.execution.chain=com.vontu.lookup.script.ScriptLookup,
    com.vontu.lookup.liveldap.LiveLdapLookup,
    com.vontu.lookup.script.ScriptLookup,
    com.vontu.lookup.script.ScriptLookup
```

Note that this entire entry should appear on a single line.

2 Save the file.

3 In the `ScriptLookup.properties` file (located by default in `\Protect\config`), configure the exact commands and arguments in the order in which Enforce Server should execute them. If the Script Lookup is called multiple times, indicate each time by a sequential number in the `script.X.command` element. For example, on a Windows platform the entries would look like:

```
script.1.command=c:/python25/python.exe
script.1.custom.args=-u,c:/Vontu/Protect/plugins/scripts/ip-lookup.py
script.2.command=c:/python25/python.exe
script.2.custom.args=-u,c:/Vontu/Protect/plugins/scripts/
   system-lookup.py
script.3.command=c:/python25/python.exe
script.3.custom.args=-u,c:/Vontu/Protect/plugins/scripts/name-lookup.py
```

In this example, the configuration file tells Enforce Server to run three scripts (`ip-lookup.py, system-lookup.py`, and `name-lookup.py`) in consecutive order.

Note: For security reasons, be sure to specify the full path to `python`, `perl`, `cscript`, or other script command. Do not specify `cmd` (Windows) or `sh` (Linux) as the command.

---

4 Save the file.

5 Make sure that all referenced lookup scripts are copied to the `\Protect\plugins\scripts` directory.

6 Go to the **Services** panel and stop and restart the **Manager** service.

See "Configuring the Script Lookup Plug-In" on page 46.

About logon credentials in scripts

You can encrypt credentials (using the Credential Utility) and then use them to authenticate to an external data repository source. When the script is invoked, the Script Lookup Plug-In decrypts the credentials and passes them as attributes to the script. The Credential Utility uses the same platform encryption keys that are used to protect user accounts and incident information within the Symantec Data Loss Prevention system.

Credentials that your script calls should be encrypted. If you choose to use credentials in clear text, you must hard code them into your script.

- Encrypting credentials for use in scripts
- About writing the scripts that use credentials

Encrypting credentials for use in scripts

This section explains how to encrypt credentials for use in your lookup scripts.

To encrypt credentials

1 Create a text file that contains the credentials that are needed to access the appropriate external systems. The format of this file is *`key=value`*, where *key* is the name of the credential. For example, `username=msantos password=esperanza9`. This text file is stored in `\Protect`.

2 Open a command line and change directories to `\Protect\bin`. This directory contains the Credential Utility.

3 Issue a command in the following format:

```
CredentialGenerator.bat input_filepath output_filepath
```

where the encrypted output file is named `ScriptLookupPassword.properties` and it is stored in `\Protect\config`. For example, on a Windows platform you would issue the following:

```
CredentialGenerator.bat C:\Vontu\Protect\mydirectory.txt
    C:\Vontu\Protect\config\ScriptLookupPassword.properties
```

4 Locate the `ScriptLookup.properties` file. By default, this file is located in the `\Protect\config` directory.

5 Open the file and edit it to include the following entries:

```
credentials.enabled=true
credentials.file.path=file_path
```

where *file_path* is the path to the output credential file (for example, `\Protect\config\ScriptLookupPassword.properties`).

6 Save the file.

7 To reload the Script Lookup Plug-In from the **Custom Attributes** tab of the **Attributes** screen (**System > Incidents > Attributes**), click **Reload Lookup Plug-ins**.

8 You can now use the encrypted credentials to authenticate to an external system.

9 Do one of the following:

- If you want to save the clear-text credentials file, move it to a secure location. It can be useful to save the file if you plan to update and re-encrypt it later.
- If you do not want to save the file, delete it now.

See "About logon credentials in scripts" on page 50.

About writing the scripts that use credentials

When the Enforce Server invokes the Script Lookup Plug-In, the plug-in decrypts any credentials at run time and passes them to the script as attributes. The credentials are then available for use within the script.

See "Encrypting credentials for use in scripts" on page 50.

Enforce Server passes the values you exported to the clear-text credential file. These values are passed in the following format: *key=value*

For example, `username=msantos` and `password=esperanza9`. In your script, use the keys to look up the corresponding values.

See "About logon credentials in scripts" on page 50.

Increasing log levels

When first implementing lookup scripts, increase log levels for the Script Lookup Plug-In and for the general lookup framework (as described in this section). Increased log levels can aid in troubleshooting.

To increase log levels

1. Enable detailed logging for the Script Lookup Plug-In.

   To do so, locate the `ManagerLogging.properties` file (located by default in `\Protect\config`) and open it with a text editor. Then edit the script lookup entry to appear as follows:

   ```
   com.vontu.lookup.script.level=FINEST
   ```

2. Enable detailed logging for the lookup framework.

   To do so, in the `ManagerLogging.properties` file, edit the servlet log handler entry to appear as follows:

   ```
   com.vontu.logging.ServletLogHandler.level=FINEST
   ```

   Then look for the `com.vontu.manager.admin.workflow.attributes.level` entry. If it already exists, make sure that it is not commented out. If it does not exist, add it. The entry should appear as follows:

   ```
   com.vontu.manager.admin.workflow.attributes.level=FINEST
   ```

   Save the file.

3 Go to the **Services** panel and stop and restart the **Manager** service.

4 When you finish testing the Script Lookup Plug-In, return the log levels to their previous state.

To do so, open `ManagerLogging.properties`, comment out or remove the `com.vontu.manager.admin.workflow.attributes.level` entry, and set `com.vontu.logging.ServletLogHandler.level` to INFO.

About the scripts that exceed the timeout

If at run time a script exceeds the timeout, the attribute framework unloads the associated plug-in. Therefore, when there is a runaway script the Enforce Server cannot execute that particular script for any subsequent incidents. If the script times out frequently, you can extend the timeout period by opening the `Plugins.properties` file and modifying the following entry:

```
com.vontu.api.incident.attributes.AttributeLookup.timeout=60000
```

Note that increasing this value may result in slower incident processing times because of slow attribute lookups.

See "Configuring the Script Lookup Plug-In" on page 46.

Index

S

T

U

W