Symantec™ Data Loss Prevention Incident Reporting and Update API Developers Guide

Version 15.5

Symantec Data Loss Prevention Incident Reporting and Update API Developers Guide

Documentation version: 15.5a

Legal Notice

Symantec Corporation
350 Ellis Street
Mountain View, CA 94043

http://www.symantec.com

Contents

Chapter 1

Introducing the Symantec Data Loss Prevention Incident Reporting and Update API

This chapter includes the following topics:

About updates to this guide

You can find the latest version of the *Symantec Data Loss Prevention Incident Reporting and Update API Developers Guide*, *Symantec Data Loss Prevention Incident Reporting and Update API Examples*, and the sample clients at the following link to the Symantec Support Center: http://www.symantec.com/docs/DOC9264.

About the Incident Reporting and Update API

The Symantec Data Loss Prevention Incident Reporting and Update API enables a Web Services developer to create applications that retrieve and update incident data that is stored in a Symantec Data Loss Prevention deployment. You can use this API to integrate incident data with other applications or systems to provide dynamic reporting, create a custom incident

remediation process, or to support business processes that rely on Symantec Data Loss Prevention incidents.

A Symantec Data Loss Prevention incident records all of the details that are associated with a message that violated a Data Loss Prevention policy. A message in this context may refer to an email message, an instant message, a file transfer, a copy or a print operation, an HTTP request, or any other protocol message that you have configured Symantec Data Loss Prevention to monitor. The data that is recorded in an incident includes the time the violation occurred, the severity of the violation, and information about the originator and recipient of the message that triggered the violation. Incidents also record data such as the text and headers of the original message and files that were attached to the original message. Finally, an incident may also contain historical data that is associated with efforts to remediate the incident in the Enforce Server administration console. This historical data includes changes to the incident severity or status and a list of any actions that were performed to help resolve or manage the incident.

For example, you can use the API to correlate Symantec Data Loss Prevention incident data with logs of the message sender's telephone calls or network usage. Or, you can create dashboard applications that integrate Symantec Data Loss Prevention incident data with data from other systems, such as intrusion detection systems. By using the update functionality of the API, you can create applications that perform custom remediation actions and then update the results of the remediation in the Symantec Data Loss Prevention incident database. The combined information from third-party systems and Symantec Data Loss Prevention, and the ability to update the status of incidents, can provide valuable information to security experts who are tasked with analyzing the data or with remediating security incidents.

The Incident Reporting and Update API is implemented as a Web Service that resides on the Enforce Server. The Web Service conforms to the Simple Object Access Protocol (SOAP) 1.1 standard, and it advertises all available operations using a Web Services Description Language (WSDL) document. You can use the WSDL document with compatible Web Services development frameworks to generate certain client code automatically. Generated proxy code for Java clients is also provided with your Symantec Data Loss Prevention installation.

About Incident Data Views

Symantec Data Loss Prevention provides incident data views that you can use to query incident data from the Enforce Server database. Incident data views expose live up-to-date information about Symantec Data Loss Prevention incidents.

The typical use case for incident data views is to query for unencrypted data in the database, and then use the INCIDENT_ID to query for the encrypted incident data using the Symantec Data Loss Prevention Incident Reporting and Update API.

For more information about Incident Data Views, contact Symantec Support.

Features of the Incident Reporting and Update API

Using the Incident Reporting and Update API, you can:

- Retrieve incident lists
- Retrieve incident details
- Retrieve binary files associated with incidents
- Update incident details
- Retrieve incident details using batched requests
- Retrieve a list of incidents by date
- Access the list of custom attributes
- Access custom status values
- Retrieve highlighted matches for one or more incidents

Components of the Incident Reporting and Update API

The Symantec Data Loss Prevention Incident Reporting and Update API includes the components described in Table 1-1.

Table 1-1 Incident Reporting and Update API components

Component	Description
Web Service Definition Language document (WSDL)	The WSDL document fully defines the request, response, and fault types that are provided by the Incident Reporting and Update API Web Service. You can obtain the WSDL document directly from an installed Enforce Server from the following URL: `https://<enforce_server>/ProtectManager/services/v2011/incidents?wsdl` (Where *enforce_server* is the IP address or host name of the Enforce Server.) You use the WSDL document to generate code when you develop Incident Reporting and Update API clients. The WSDL also provides the Web Service bindings to client applications at runtime. See "Generating Web Service client proxy code" on page 21.

Table 1-1 Incident Reporting and Update API components *(continued)*

Component	Description
XML Schema Definitions (XSD files)	The Incident Reporting and Update API XSD files describe the contents and structure of the XML request and response documents that are associated with each Web Service call. The XSD files also define the incident data types that you can use to represent incident details stored in the Enforce Server database. The XML schemas are available directly by the Incident Reporting and Update API WSDL. XSD files are also available in the following JAR file on an Enforce Server host (line break added for legibility): `C:\Program Files\Symantec\DataLossPrevention\EnforceServer\15.5\Protect\tomcat\lib\incidentapi-2011-schema-2.0.jar` See "About incident detail types" on page 26.
Proxy Java classes generated from the WSDL	Generated proxy (skeleton) Java code is available in the following JAR file on an Enforce Server host (line break added for legibility): `C:\Program Files\Symantec\DataLossPrevention\EnforceServer\15.5\Protect\tomcat\lib\incidentapi-2011-generated-2.0.jar` Include this file in your Java `CLASSPATH` when you compile a Java-based Web Services client.
Example Web Service clients	You can download sample code that demonstrates Java and Microsoft .NET client implementations of the Incident Reporting and Update API Web Service. The sample code is available at the Symantec Support Center at the following URL: http://symantec.com/docs/DOC9264

Requirements for using the Incident Reporting and Update API

To use the Incident Reporting and Update API, you should be familiar with the process of developing Web Services clients in a programming language of your choice. Although you can develop SOAP-based Web Service clients in a variety of programming languages, Symantec offers formal support only for Java 1.8 and .NET 4.5 and 4.6 implementations.

In addition, Symantec provides example reporting clients for both Java and .NET. The examples are provided only for demonstration purposes. They are not supported for production use or development, and Symantec provides no ongoing support to resolve bugs or add functionality to the example applications. Symantec offers formal support only for the Incident Reporting and Update API Web Service (WSDL implementation) and schema files, which provide the core functionality of the Incident Reporting and Update API. Contact your Symantec sales representative to obtain the example code.

Symantec recommends that you use the Metro Web Service 2.2 framework or Microsoft .NET 4.5 or 4.6 framework to automatically generate code from the supplied WSDL document. Because the Web Service itself was developed using the Metro Web Services 2.2 stack, you may choose to use the same stack to speed client development.

See "Generating Web Service client proxy code" on page 21.

About localization of system-defined fields

The Incident Reporting and Update API Web Service localizes all system-defined fields returned in Web Service responses. However, user-defined content such as custom attribute fields are not localized either in the Enforce Server administration console or the Incident Reporting and Update API Web Service. Client implementations must consider the possibility of non-localized data when transforming or displaying user-defined content in incident data results.

About Incident Reporting and Update API security

The Incident Reporting and Update API Web Service requires HTTPS for communication with client applications. The underlying SSL transport provides end-to-end encryption of all data transmitted between the Web Service and authorized clients. The Web Service performs no additional encryption for the incident data or binary data contained in responses.

The Incident Reporting and Update API Web Service authenticates each client request using the HTTP basic authentication scheme. Client applications must supply the credentials of a valid Symantec Data Loss Prevention user in the HTTP authentication headers of each request to the Web Service. You must create this user account using the Enforce Server administration console before accessing the Web Service.

Note: Symantec Data Loss Prevention authenticates all Incident Reporting and Update API clients using password authentication. If you configure Symantec Data Loss Prevention to use certificate authentication, any user account that is used to access the Incident Reporting and Update API Web Service must have a valid password in the Enforce Server administration console configuration.

See "Creating a user and role for an Incident Reporting and Update API client" on page 16.

An authenticated user is authorized to access the Incident Reporting and Update API Web Service if the user is assigned to a role where one of the following user privileges is granted:

- Incident Reporting—allows read-only access to incident data
- Incident Update—allows updates to incident data

See "Creating a user and role for an Incident Reporting and Update API client" on page 16.

Chapter 2

Implementing an Incident Reporting and Update API client

This chapter includes the following topics:

About Incident Reporting and Update API client implementations

The Incident Reporting and Update API provides a Web Service interface that clients can use to retrieve and update incident data. Before a client can interact with the Incident Reporting and Update API Web Service, the following items must be configured in the Enforce Server administration console:

- A user account that is assigned to a role that has permission to access the Incident Reporting and Update API Web Service.
  See "Creating a user and role for an Incident Reporting and Update API client" on page 16.
- A saved report that queries Symantec Data Loss Prevention incidents based on the constraints and filters you specify. This report must be available to the Incident Reporting and Update API user account that accesses the Web Service.
  See "Creating a saved report for an Incident Reporting and Update API client" on page 19.

A Web Service client provides the credentials of the Enforce Server user account in each request that it sends to the Incident Reporting and Update API Web Service. The Web Service authenticates the credentials, and then authorizes or denies the request based on whether the authenticated user has privileges to access the Web Service.

About reporting clients

To retrieve incident data, a reporting Web Service client typically begins by requesting a list of incidents that are specified by a saved incident report that was previously defined in the Enforce Server administration console. Although the saved incident report may return a very large list of incidents, the Web Service client can request a subset of incidents by specifying a date in the Web Service call. The Web Service only returns incidents that were created after this specified date. You can also apply filters to the saved incident report to further limit the incidents returned. As a best practice, a Web Service client should log the time of its most recent incident list request. Each Web Service request for an incident list should retrieve only those incidents that were created since the time of the last request.

After obtaining a list of incident IDs, a Web Service client can submit further requests to obtain detailed incident data for a specific incident. The client can request these incidents individually, or the client can make a Web Service call that uses a "batched" approach where the client requests incident data for multiple incidents in a single call. When you request incident IDs in batches, you can improve performance of the client. Symantec recommends that you use batches of 50 to 100 incidents for best performance. The client can also request to retrieve the full binary data associated with a given incident (the complete message, file, or attachment that generated the incident). The complete list of Incident Reporting and Update API Web Service operations, as well as error messages, are defined in the WSDL document.

See "Implementing an Incident Reporting and Update API client" on page 14.

The Incident Reporting and Update API Web Service returns incident lists, incident details, and incident binaries to clients using XML-formatted SOAP 1.1 messages. Incident details reference a common XML schema and conform to specific incident types based on the Symantec Data Loss Prevention product and product component that generated the incident.

See "About incident detail types" on page 26.

About update clients

To update incident data, an Incident Reporting and Update API client application makes a Web Service call that contains one or more batches of incidents to be updated. Each batch specifies a list of incident identifiers and the fields and values that should be updated for those incidents. Using this call, you can update multiple incidents with the same data.

You can update the following incident fields using the Incident Reporting and Update API:

- Incident severity
- Incident status
- Custom attribute values
- Data owner data structure
- Notes/Comments data structure
- Remediation status
- Remediation location

The response to an update call includes information on the success or failure of the updates.

It is important to consider the size of the batches to maximize performance of the API. Including more incidents in a batch increases the performance of the API. Making a Web Service call to update a single incident is the least efficient way to update an incident. For best performance when updating incident data, Symantec recommends that each batch contain between 50 to 500 incident IDs.

Implementing an Incident Reporting and Update API client

The following table summarizes the steps that are involved in implementing an Incident Reporting and Update API Web Service client. See the associated sections for more details about each step.

Table 2-1 Implementing an Incident Reporting and Update API client

Step	Action	Description
1	Install a development system.	See "Installing a development system" on page 15.
2	Create an Incident Reporting and Update API user and role.	See "Creating a user and role for an Incident Reporting and Update API client" on page 16.

Table 2-1 Implementing an Incident Reporting and Update API client *(continued)*

Step	Action	Description
3	Create a saved report. (This step is required only when the client needs to retrieve a list of incident IDs.)	See "Creating a saved report for an Incident Reporting and Update API client" on page 19.
4	Generate client code from the Incident Reporting and Update API Web Service WSDL.	See "Generating Web Service client proxy code" on page 21. For Java clients, you may use the generated proxy code that is available in the following JAR file on an Enforce Server host (line break added for legibility): `c:\Symantec\DataLossPrevention\EnforceServer\15.5\Protect\tomcat\lib\incidentapi-2011-generated-2.0.jar` Include this file in your Java `CLASSPATH` when you compile a Java-based Web Services client.
5	Implement calls to the Incident Reporting and Update API Web Service.	See "About Incident Reporting and Update API Web Service operations" on page 24.
6	Troubleshoot the client implementation.	See "Troubleshooting Incident Reporting and Update API client applications" on page 31.

Installing a development system

To develop and test your client implementations of the Incident Reporting and Update API, set up a test installation of Symantec Data Loss Prevention. You can use a single-tier, two-tier, or three-tier installation. A single-tier Symantec Data Loss Prevention installation, where the Enforce Server, detection server, and the Oracle database are deployed on a single computer, is sufficient for development purposes.

See the *Symantec Data Loss Prevention Installation Guide* at http://www.symantec.com/docs/DOC9257 for more information.

To verify client functionality, you must populate the development system with example incident data.

A SOAP development tool can be useful during deveolpment of client applications. A tool such as soapUI can consume the WSDL and generate XML request documents automatically. The tool can send those requests to the Web Service and displays the XML response document. See http://www.soapui.org/.

Note: Symantec does not recommend testing Web Service clients against a live production server, or using copies of live incident data. Doing so increases the possibility of exposing confidential data (incident data and user credentials) on unprotected development and test computers, or of incorrectly updating your production incident database.

Creating a user and role for an Incident Reporting and Update API client

You must use the Enforce Server administration console to create a Web Service user and role before you can connect to the Web Service from a client application.

To create an Incident Reporting and Update API Web Service role and user

1. Log on to the Enforce Server administration console as an Administrator.
2. Select **System > Login Management > Roles**.
3. Click **Add Role**.
4. Type a name for the new role in the **Name** field. For example, type “Incident API Client Role.”

5 In the **User Privileges** section of the screen, select items as described in the following table.

Item	Description
Incidents: View	Select **View**, and then select the incident types that the Web Service role can view or update. If you do not select a particular incident type, the Web Service does not return incident details of that type to clients that use this role and clients cannot update incidents of that type.
Incidents: Actions	Select the **Remediate Incidents** privilege.
Incidents: API	Select one or both of the following two user privileges: **Incident Reporting** — enables Web Services clients to retrieve incident details using the Incident Reporting and Update API. **Incident Update**—enables Web Services clients to update incident details using the Incident Reporting and Update API.
Incidents: Display Attributes	Select all of the attributes that you want to include in incident detail responses for this role. **Note:** If your client implementation uses the `incidentBinaries()` operation, select the **Attachments/Files** permission.

Item	Description
Incidents: Custom Attributes	Custom attributes are optional data fields that you can use to store supplemental information about an incident. Your organization may use custom attributes to assist in the workflow for remediating or evaluating incidents. Select **View** for each custom attribute that you want to include in incident detail responses for this role or select **View All** to include all custom attributes. Select **Edit** for each custom attribute that you want to allow a Web Service client to update or select **Edit All** to allow updates to all custom attributes. You can use the `listCustomAttributes()` method of Incident Reporting and Update API to retrieve a list of custom attributes that have been defined in the deployment. See "listCustomAttributes()" on page 45. See the *Symantec Data Loss Prevention Administration Guide* at http://www.symantec.com/docs/DOC9261 for more information about custom attributes.

Note: Role-based access privileges provide a way to limit the results of a Web Service incident list request or incident detail request. For example, the Incident Reporting and Update API WSDL does not enable a client to retrieve only Endpoint-related incident IDs when requesting an incident list. However, you can use the **User Privileges** selection to limit the Web Service user role to view only Endpoint-related incidents. Alternately, you can create a saved report that returns only Endpoint-related incidents, and use that report to retrieve an incident list.

See "Creating a saved report for an Incident Reporting and Update API client" on page 19.

6 (Optional) Click the **Incident Access** tab to set additional conditions that limit the incidents that Incident Reporting and Update API clients may access.

7 Click **Save**.

8 Select **System > Login Management > DLP Users**.

9 Click **Add User**.

10 Type the credentials for the new user in the **Name**, **Password**, and **Re-enter Password** fields.

Note: Symantec Data Loss Prevention authenticates all Incident Reporting and Update API clients using password authentication. If you configure Symantec Data Loss Prevention to use certificate authentication, any user account that is used to access the Incident Reporting and Update API Web Service must have a valid password.

---

11 In the **Roles** section of the screen, select the new role you created in step 4. For example, select “Incident API Client Role.”

12 Select the same Incident API Client Role role in the **Default Role** menu.

13 Click **Save**.

Creating a saved report for an Incident Reporting and Update API client

Clients of the Incident Reporting and Update API Web Service request a list of incident IDs by specifying a saved report ID. Use the Enforce Server administration console to create one or more saved reports. The saved report defines a collection of incident IDs that a Web Service client can retrieve with a call to the `incidentList` method. You can create multiple saved reports as necessary for your client application.

See “About Incident Reporting and Update API Web Service operations” on page 24.

The saved report that you create must return an incident list. You cannot access a saved dashboard or summary report using a Web Service client. You cannot retrieve a system-defined incident list using the Incident Reporting and Update API. However, you can use a system-defined incident list to generate a new saved report.

The instructions that follow describe how to create a new saved report for use with the Incident Reporting and Update API.

To create a saved report for an Incident Reporting and Update API Web Service client

1 Logon to the Enforce Server administration console as the Incident Reporting and Update API Web Service user.

Note: The saved report must be accessible to the Incident Reporting and Update API Web Service user.

See "Creating a user and role for an Incident Reporting and Update API client" on page 16.

---

2 Select **Incidents > Incident Reports**.

3 Select an existing incident list from the list of available reports. You may select a system-defined incident list, such as **Incidents – All**, as the basis for the new report.

4 Optionally, use the **Filter** and **Severity** controls report to limit the incident IDs that the report returns.

5 Click **Advanced Filters & Summarization**.

6 In the **Summarize By** menu, verify that **<no primary summary selected>** and **<no secondary summary selected>** are both chosen. You cannot access a summary report using the Incident Reporting and Update API Web Service.

7 Optionally, click **Add filter** and add one or more advanced filters to limit the incident IDs that the report returns.

Note: Role-based access privileges may further limit the results that are returned from the Incident Reporting and Update API Web Service.

See "Creating a user and role for an Incident Reporting and Update API client" on page 16.

---

8 Select **Report > Save As**.

9 Type a name for the report in the **Name** field, and an optional description in the **Description** field.

10 Click **Save**.

The new saved report appears under the **Saved Reports** heading in the left pane.

11 To determine the ID of the saved report, move your mouse pointer over the report name. The tool tip displays the report ID and name of the report. For example, if the tool tip displays "View Report 83: 'Incident Reporting and Update API Saved Report'," a Web Service client can request the incident list by specifying "83" in the `incidentList` call.

The status bar of your browser also displays the saved report ID at the end of the link name when you place the pointer over the saved report name.

Generating Web Service client proxy code

Symantec recommends that you use a Web Service development framework when building an Incident Reporting and Update API client application. Using a development framework enables you to automatically generate Web Service skeleton code for Web Service calls and data types. The skeleton code is generated directly from the Incident Reporting and Update API WSDL document and supporting schema documents. You supply the WSDL as a URL served by your development Symantec Data Loss Prevention server installation.

Although you can use a variety of frameworks to generate client code, Symantec recommends using Metro Web Services 2.2 or another environment that generates Java API for XML Web Services (JAX-WS) style code artifacts, such as the Java EE SDK. A JAR file that contains generated Java skeleton code is available on the Enforce Server host.

See "Components of the Incident Reporting and Update API" on page 9.

Microsoft .NET developers should use the .NET 4.5 or 4.6 runtime, which supports the Windows Communication Foundation (WCF) API for Web Services development. The Incident Reporting and Update API Web Service was developed using Metro Web Services, which provides full interoperability with Microsoft .NET WCF clients. A generated proxy class file is available with the Incident Reporting and Update API example client distribution. The sample code is available at the Symantec Support Center at the following URL:

http://symantec.com/docs/DOC9264

The framework you choose will generally provide both a command-line utility and build script support for consuming a WSDL document and schema to generate code. For example, the Java EE SDK includes the `wsimport` command-line utility for generating Java code. See your framework documentation for details about generating skeleton code from a WSDL file.

Consuming the Incident Reporting and Update API WSDL over SSL

The Incident Reporting and Update API WSDL document is available from the Enforce Server at the following URL:

`https://<enforce server>/ProtectManager/services/v2011/incidents?wsdl`

Where *enforce server* is the host name or IP address of the Enforce Server.

The Enforce Server administration console requires SSL transport for all communication. Any utility that you use to consume the WSDL and generate skeleton code must first be able to negotiate the SSL connection with the Enforce Server. For more information about using certificates with Symantec Data Loss Prevention, see "About configuring certificate authentication" in the *Symantec Data Loss Prevention Administration Guide*.

If your Symantec Data Loss Prevention deployment already uses certificate authentication, use a Web browser to export the Enforce Server certificate. You can then import that certificate

to a keystore that is used in the client environment. The Enforce Server keystore is in the following location on the Enforce Server host:

`c:\Symantec\DataLossPrevention\EnforceServer\15.5\Protect\tomcat\conf\.keystore`

Configure your client to use the Enforce Server keystore file. For example, Java utilities such as `wsimport` can specify the keystore location using command-line options.

For example (line breaks added for legibility):

```
-Djavax.net.ssl.keyStore=/opt/Symantec/DataLossPrevention/EnforceServer/
                                    15.5/Protect/tomcat/conf/.keystore
-Djavax.net.ssl.keyStorePassword=protect
-Djavax.net.ssl.trustStore=/opt/Symantec/DataLossPrevention/EnforceServer/
                           15.5/Protect/tomcat/conf/.keystore
-Djavax.net.ssl.trustStorePassword=protect
```

See your development framework documentation for more information.

Authenticating a client with the Incident Reporting and Update API Web Service

The Incident Reporting and Update API Web Service authenticates each client request using the HTTP Basic authentication scheme. Client applications must supply the credentials of a valid Symantec Data Loss Prevention user in the HTTP authentication headers of each request to the Web Service. Each request must be made over an SSL connection to the Incident Reporting and Update API Web Service.

To authenticate using HTTP basic authentication

1. Obtain user credentials interactively or using a configuration file.
2. Validate SSL certificates as necessary for HTTPS communication with the Web Service.
3. Create a binding to the Web Service port, specifying HTTP Basic authentication as needed.
4. Add the user credentials to the Web Service connection.

The Web Service returns an `authenticationFailedFault` if the Enforce Server cannot authenticate using the supplied credentials. For security reasons, `authenticationFailedFault` provides no details about why the authentication failed.

The exact method for performing these tasks depends on the programming language in which you develop the Web Service client. The following code examples show how to add authentication headers and use HTTP Basic authentication in Java and .NET clients. The examples use hard-coded user credentials for simplicity.

The code examples do not show a method for validating SSL certificates.

The full Java and .NET example clients simplify the HTTPS connection process by bypassing SSL certificate validation. See the full code for the example clients for more information. The sample code is available at the Symantec Support Center at the following URL:

http://symantec.com/docs/DOC9264

Java authentication example

Java clients add user credentials to the request context of the Web Service port binding. The following example shows the Java client methods used to add required authentication headers:

```
// Define user credentials.
//
String client_username = "WS_Client";
String client_password = "welcome";

// Create the Incident Reporting and Update API service.
//
URL serviceUrl = new URL("https://EnforceMachineHostName/IP/ProtectManager/services/v2011/incide
QName serviceNamespace = new QName("http://www.company.com/v2011/enforce/webservice/incident",
   "IncidentService"));
IncidentService service = new IncidentService(serviceUrl, serviceNamespace);

// Bind credentials to the service port.
//
IncidentServicePortType servicePort = service.getIncidentServicePort();
BindingProvider portBP = (BindingProvider) servicePort;
portBP.getRequestContext().put(BindingProvider.USERNAME_PROPERTY, client_username);
portBP.getRequestContext().put(BindingProvider.PASSWORD_PROPERTY, client_password);
```

.NET authentication example

.NET clients use Microsoft Windows Communication Foundation (WCF) HTTP binding methods to supply basic HTTP authentication headers with the Web Service request. The following example shows the Microsoft .NET client methods used to add required authentication headers:

```
// Define user credentials.
//
String client_username = "WS_Client";
String client_password = "welcome";

// Prepare HTTP bindings with username/password
// basic authentication. Init incident reporting
```

```
// service port
//
// An Example of the URL is as follows
this.url = "https://EnforceMachineHostName/IP/ProtectManager/services/incidents?wsdl";

EndpointAddress epAddress = new EndpointAddress(this.url);

BasicHttpBinding basicAuthBindings = setupBasicAuthentication();
client = new IncidentServicePortTypeClient(basicAuthBindings, epAddress);

client.ClientCredentials.UserName.UserName = client_username;
client.ClientCredentials.UserName.Password = client_password;

IEndpointBehavior behavior = new HttpBasicAuthenticationEndpointBehavior();
client.Endpoint.Behaviors.Add(behavior);

// connect to the reporting API service
//
client.Open();
```

About Incident Reporting and Update API Web Service operations

Table 2-2 lists the operations that the Incident Reporting and Update API Web Service supports for retrieving and updating incident data.

Each operation takes a request that encapsulates arguments for the operation. A successful request to the Web Service returns a corresponding response document that contains the incident data or the status of update operations. Failed operations return one of several possible faults.

See “Troubleshooting Incident Reporting and Update API client applications” on page 31.

Table 2-2 Incident Reporting and Update API service operations

Operation	Description	Details
`incidentList()`	The `incidentList()` method returns a list of incident IDs described by a saved report. The client specifies the saved report ID to retrieve and a date value. The date value limits the list of incidents to those created after the specified date. A client application generally begins with a call to `incidentList()`, and wraps the returned values in an array or other container to work with the incident IDs. The application can then use either the `incidentDetails()` or `incidentBinaries()` Web Service calls to retrieve additional details about one or all of the returned incident IDs. The application can also use the `updateIncidents()` method to update incident details.	See "incidentList()" on page 34.
`incidentDetail()`	The `incidentDetail()` method returns the details of a single Symantec Data Loss Prevention incident, such as its creation date, severity, and status. A client can use the fields of the response document as-is, or the client may cast the response document into a more specific incident detail type.	See "incidentDetail()" on page 37. See "About incident detail types" on page 26.
`incidentBinaries()`	The `incidentBinaries()` method returns the binary data of a single Symantec Data Loss Prevention incident, such as the original message that generated the incident or files that were attached to the original message.	See "incidentBinaries()" on page 42.
`updateIncidents()`	The `updateIncidents()` method allows a client program to update various attributes of incident data. The client specifies one or more batches of incident IDs and a set of attributes and values to update for each batch. A client can update incident severity, incident status, custom attributes, data owner details, notes, remediation status, and remediation location.	See "updateIncidents()" on page 47.
`listCustomAttributes()`	The `listCustomAttributes()` method returns a list of custom attributes that are defined in the Symantec Data Loss Prevention deployment.	See "listCustomAttributes()" on page 45.
`listIncidentStatus()`	The `listIncidentStatus()` method returns a list of all custom status values that are defined in the Symantec Data Loss Prevention deployment.	See "listIncidentStatus()" on page 46.

Table 2-2 Incident Reporting and Update API service operations *(continued)*

Operation	Description	Details
`incidentViolations()`	The `incidentViolations()` operation takes the incidentID as input and returns the text-based content for each match in the incident. In the Enforce Server administration console, this content appears as highlighted matches for the incident.	See "incidentViolations()" on page 53.

About incident detail types

A successful request to the `incidentDetail()` operation returns a single XML stanza of the type `IncidentDetailType`. The stanza describes the basic characteristics that are shared by all Symantec Data Loss Prevention incidents. This includes the unique ID of the incident, the date on which the incident was created, the severity and status of the incident, the policy and rule that were violated, and so forth.

Incidents that are created by different Symantec Data Loss Prevention products contain additional information that is unique to the product group. For example, Symantec Data Loss Prevention network products (Network Monitor and Network Prevent) may contain a message header or recipient field. Network Discover incidents include the name of the scan that generated the incident, and may include the name of a file that generated a policy violation.

The Incident Reporting and Update API XML schema defines product-specific incident detail types by extending the base `IncidentDetailType` with additional fields. Six product incident detail types are defined:

Product incident detail type	Products that generate this incident type
`NetworkIncidentDetailType`	■ Network Monitor ■ Network Prevent for Email ■ Network Prevent for Web
`EndpointIncidentDetailType`	■ Endpoint Discover ■ Endpoint Prevent
`DiscoverIncidentDetailType`	■ Network Discover ■ Network Protect
`DiscoverBoxCrawlerIncidentDetailType`	■ Cloud Storage Discover
`RestDARIncidentDetailType`	■ Cloud Service Connector
`RestDIMIncidentDetailType`	■ Cloud Service Connector

Figure 2-1 shows the base fields defined in `IncidentDetailType` and the extension fields defined in the Network Endpoint, and Discover incident detail types; Figure 2-2 shows the REST DAR incident detail types; and Figure 2-3 shows the REST DIM incident detail types. To work with the additional fields of a product incident detail type, a client application can cast the `IncidentDetailType` into a more specific product incident detail type. The `messageSource` field in `IncidentDetailType` provides a key that indicates the Symantec Data Loss Prevention product that generated the incident.

Figure 2-1 Hierarchy of product incident detail types (not including REST incidents)

Figure 2-2 Hierarchy of product incident detail types (including REST DAR incidents)

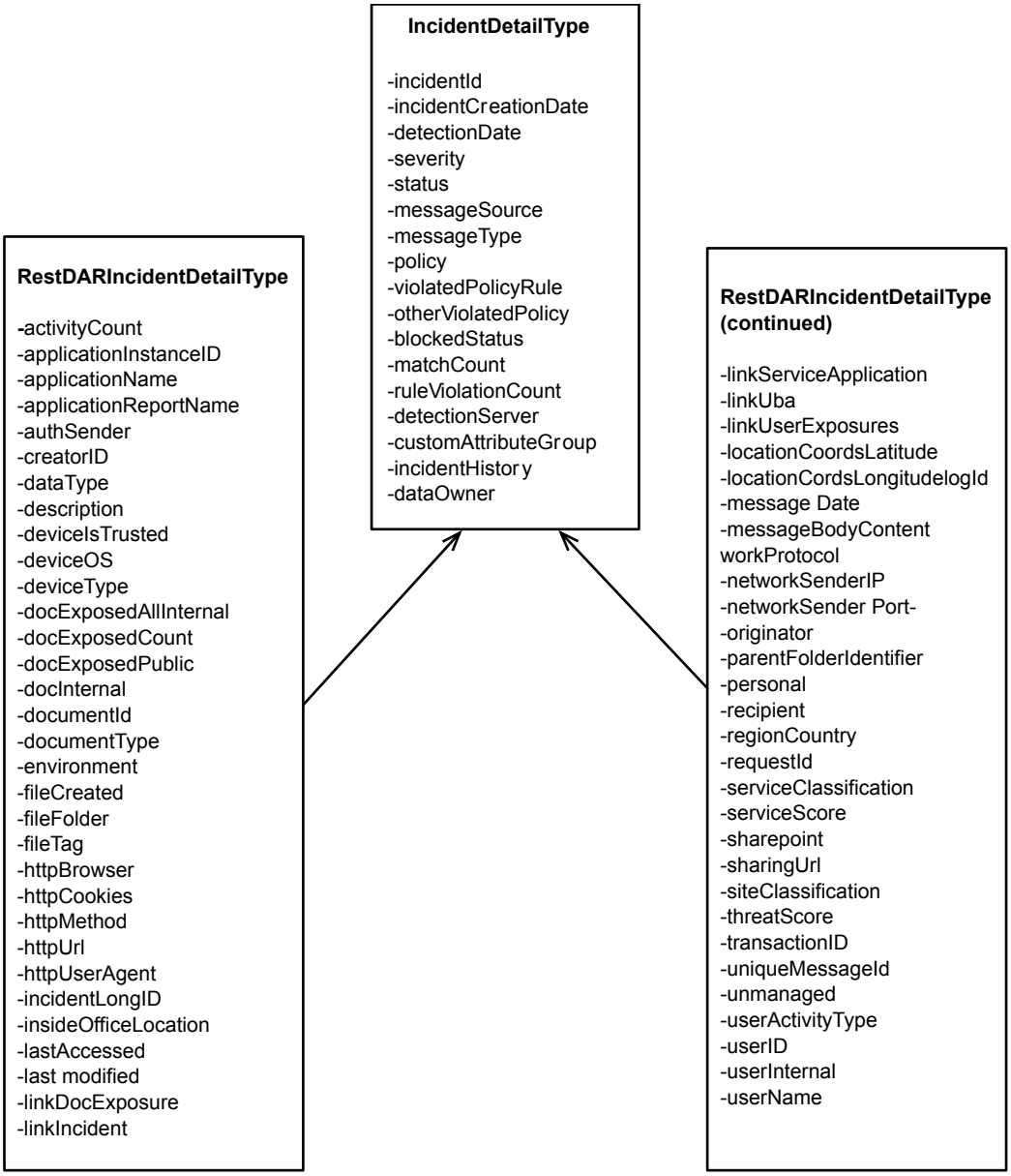
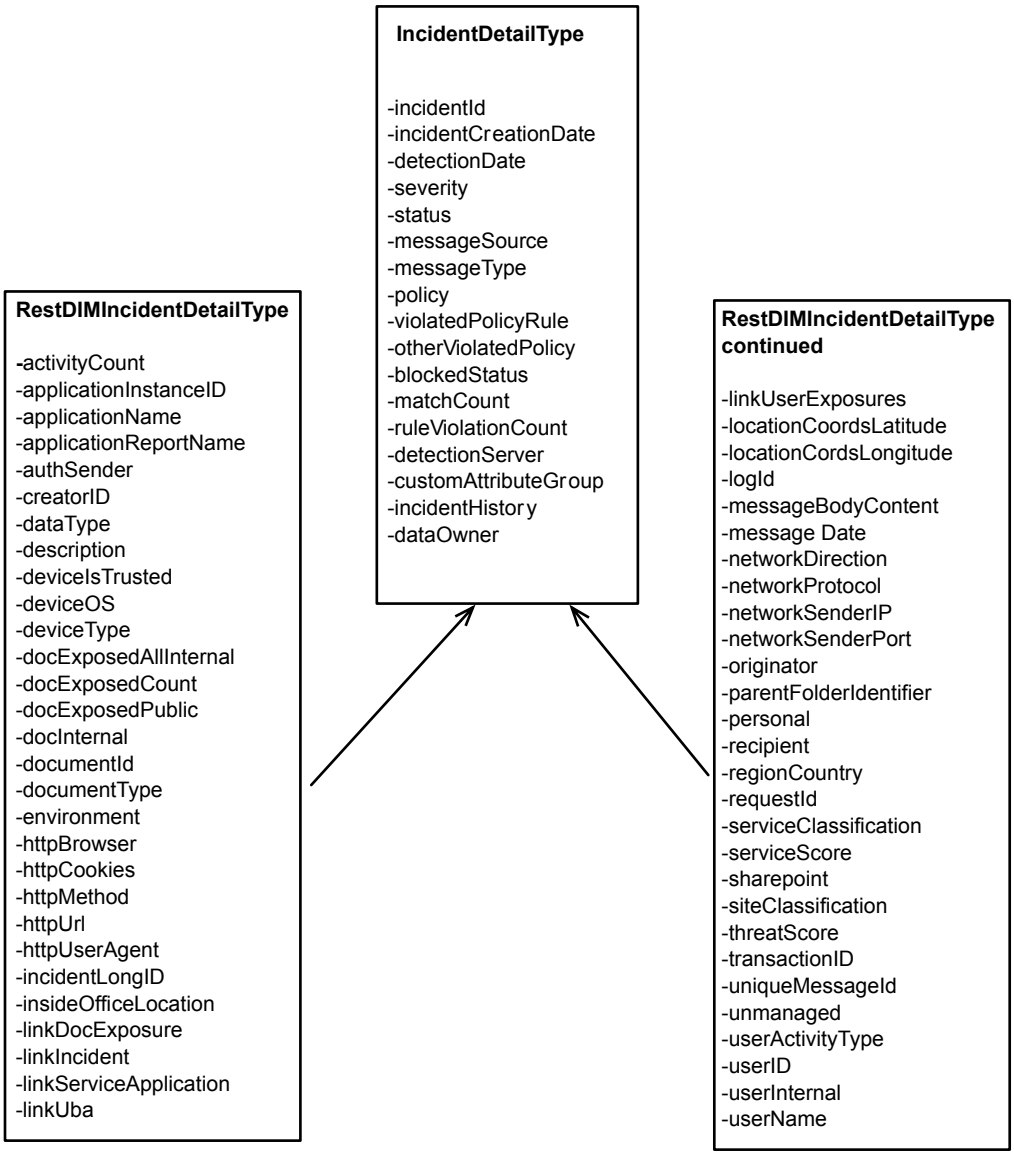


Figure 2-3 Hierarchy of product incident detail types (including REST DIM incidents)



Within a particular Symantec Data Loss Prevention product, incident details may be further differentiated by the product component that detected the incident or by the protocol used to transmit the original message. For example, a user can generate an incident by attempting to copy sensitive data to the clipboard or by sending sensitive data from an email application. These two cases create different kinds of incident data. In the first example, the incident data includes information about the application the user was using when they tried to copy data to the clipboard. In the second example, the incident data records the subject of the email message that generated the incident.

To account for these differences, the Incident Reporting and Update API XML schema further extends the product group incident types into product component incident detail types. The `messageType` field in the `IncidentDetailType` provides a key that identifies a valid component type for the incident. Based on the `messageType`, you can cast to the most specialized component type or to the associated product category type as needed.

Figure 2-4 shows the product component incident detail types that extend the five product group incident types. (Note that the attributes for individual product and component types are omitted for space considerations.)

Figure 2-4 Hierarchy of component incident detail types

Troubleshooting Incident Reporting and Update API client applications

Each of the Incident Reporting and Update API operations can potentially return one of several faults to indicate why an operation failed. As a best practice, client applications should display or log any faults that cannot be resolved internally by the application code.

For example, if an application prompts the user to enter credentials or individual incident IDs, the application should notify the user when either an **authenticationFailedFault** or **incidentNotFoundFault** occurs. If the application uses hard-coded credentials or derives incident IDs from `incidentList` requests, the application may instead log these faults to an application-specific log file.

In addition to the faults that are provided by the Incident Reporting and Update API, these Symantec Data Loss Prevention operational log files store additional information about the behavior of the Web Service implementation:

- `webservice_access.log` records both successful attempts and failed attempts to access the Incident Reporting and Update API Web Service. This log file records many of the same authentication error conditions that are returned to Web Service clients as API faults. However, the log file aggregates this information for all clients that access the Incident Reporting and Update API Web Service on the Enforce Server. `webservice_access.log` also logs successful client requests with:
  - Time of the request
  - Name of the user who made the request
  - Success or failure of the request
  - Type of request
  - Amount of time taken to complete the request
- `webservices_soap.log` records the entire SOAP request and response for most requests to the Incident Reporting and Update API Web Service. The log records all requests and responses except responses to incident binary requests. Although this log also records Incident Reporting and Update API faults, any Java exceptions generated by the Enforce Server are logged to the Tomcat log file.
  `webservices_soap.log` is not created by default. To begin logging to this file, edit the `c:\Program Files\Symantec\DataLossPrevention\EnforceServer\15.5\Protect\config\ManagerLogging.properties` (Windows) or `/opt/Symantec/DataLossPrevention/ Enforce Server/15.5/Protect/config/ManagerLogging.properties` (Linux) file to set the `com.vontu.enforce.reportingapi.webservice.log. WebServiceSOAPLogHandler.level` property to `INFO`.

  Note: The `webservices_soap.log` file includes sensitive information in the response details. Symantec recommends securing these log files to avoid loss of the sensitive information they include. Symantec also recommends cleaning up these log files after you have finished troubleshooting.

In addition to these operational log files, the following log files may contain additional information about the health or availability of the Web application that implements the Incident Reporting and Update API Web Service:

- `manager_operational.log` is a Symantec Data Loss Prevention operational log file that records lifecycle and system events that are associated with the Enforce Server. The Incident Reporting and Update API Web Service works with the Enforce Server to provide incident data through SOAP requests.
  See “Operational Log Files and Codes” in the *Symantec Data Loss Prevention Administration Guide* for more information about this log file.

- The Tomcat server log file may also contain information about failed deployment for the Symantec Data Loss Prevention Web applications. Consult this log file last, after you first examine the Incident Reporting and Update API faults, Web Service operational logs, and Symantec Data Loss Prevention operational logs. The Enforce Server stores the local Tomcat log file in the following location:
  - Windows (line break added for legibility):
    `c:\ProgramData\Symantec\DataLossPrevention\EnforceServer\15.5\Protect\logs\tomcat\localhost.`*`timestamp`*`.log`
  - Linux:
    `/var/log/Symantec/DataLossPrevention/EnforceServer/15.5/tomcat/localhost.`*`timestamp`*`.log`

The following troubleshooting task pertains to Windows 2008 R2 operating systems running a .NET client where FIPS mode is being used on the server.

If a client is trying to connect to a server installed in FIPS mode and during execution the system throws a connection error that is related to the TLS version, you need to make the following changes to the Windows 2008 R2 computer where you are running the client:

- Set the registry setting to FIPS:
  HKLM\System\CurrentControlSet\Control\Lsa\FIPSAlgorithmPolicy\Enabled
  This registry value reflects the current FIPS setting. If this setting is enabled, the value is 1. If this setting is disabled, the value is 0. Enable FIPS by setting the value to 1 if it is 0.
- Select Run > Local Security Policy > Local Policies > Security Options > System cryptography.
  Use FIPS compliant algorithms for encryption, hashing and signing to 'enabled'.

Refer to the .NET example client that is available at the Symantec Support Center at the following URL:

http://symantec.com/docs/DOC9264

Appendix A

Incident Reporting and Update API Web Service call reference

This appendix includes the following topics:

incidentList()

`incidentList()`—returns a list of incident IDs by executing a saved report on the Enforce Server.

Syntax

```
IncidentListResponse = IncidentServicePortType.incidentList(IncidentListRequest);
```

Inputs

This call takes a single `IncidentListRequest` type as its argument. `IncidentListRequest` encapsulates the ID of the saved report to execute on the Enforce Server and the date used to constrain the list of incident IDs returned by the call.

The following table describes the `IncidentListRequest` fields.

Table A-1 IncidentListRequest instance variables

Name	Type	Description
`savedReportId`	`int`	Specifies the ID of the saved report to execute on the Enforce Server. This report must be created using the Enforce Server administration console before executing the Web Service call. There is no mechanism to create a saved report by the Incident Reporting and Update API Web Service. See "Creating a saved report for an Incident Reporting and Update API client" on page 19.

Table A-1 IncidentListRequest instance variables *(continued)*

Name	Type	Description
`incidentCreationDateLaterThan`	`dateTime`	Constrains the list of returned incident IDs to include only incidents that were created after the `incidentCreationDateLaterThan` date. A null value retrieves no reports. As a best practice, client applications should record the creation time of the latest incident they retrieved and use that time to retrieve only newly created incident IDs. If you need to further constrain the list of returned incident IDs, either: ■ Customize filters for the saved report that you reference, or ■ Configure role-based access controls for the Web Service client user to limit the type of incidents that can be viewed. See "Creating a user and role for an Incident Reporting and Update API client" on page 16.

Outputs

Returns an `IncidentListResponse` object that encapsulates a list of incident IDs. The list is a subset of the IDs described by the saved report, constrained by the value of the `incidentCreationDateLaterThan` instance variable and any role-based access controls applied to the Web Service user.

See "Creating a saved report for an Incident Reporting and Update API client" on page 19.

See "Creating a user and role for an Incident Reporting and Update API client" on page 16.

Example

The following example shows how a Java client retrieves a list of incidents with `incidentList()`.

Note: The Java code in this section is for instructional purposes only and does not demonstrate a complete client implementation. The sample code is available at the Symantec Support Center at the following URL:

http://symantec.com/docs/DOC9265

```
IncidentListRequest request = new IncidentListRequest();
XMLGregorianCalendar setDate = null;
GregorianCalendar c = new GregorianCalendar();
c.setTime(incidentDate);
setDate = DatatypeFactory.newInstance().newXMLGregorianCalendar(c);
request.setIncidentCreationDateLaterThan(setDate);
request.setSavedReportId(Integer.parseInt(reportId));
try
   {
     IncidentServicePortType clientPort = client.getPortClient();
     IncidentListResponse response=clientPort.incidentList(request);
     List<Integer> incidentIds = response.getIncidentId();
     for(int incidentid:incidentIds)
       {
         System.out.println("Incident ID: " + incidentid);
       }

     }
```

Faults

The `incidentList` call can return the following general faults:

- `AuthenticationFailedFault`
- `AuthorizationFailedFault`
- `ServiceErrorFault`

  If a client specifies an invalid saved report ID, the error is captured in a `ServiceErrorFault`. The `ServiceErrorFault` also captures any unknown runtime errors.

incidentDetail()

`incidentDetail()`—requests the details of a specified incident.

Syntax

```
IncidentDetailResponse = IncidentServicePortType.incidentDetail(IncidentDetailRequest);
```

Inputs

This call takes a single `IncidentDetailRequest` type as its argument. `IncidentDetailRequest` encapsulates a list of incident IDs. The response returns details for each of the incidents in the list. The request may optionally indicate whether the Web Service should also return incident violation data and historical information.

For best performance, Symantec recommends that you retrieve multiple incidents in a single call to the `incidentDetails()` method. Retrieving 50 – 100 incidents per call provides the best performance.

The following table describes the `IncidentDetailRequest` fields.

Table A-2 IncidentDetailRequest fields

Name	Type	Description
`incidentLongId`	long	This required field identifies the unique ID of the Symantec Data Loss Prevention 12.5.2 or later incident whose details you want to retrieve. You may include any number of incident IDs in the request.
`incidentId`	int	This required field identifies the unique legacy (Symantec Data Loss Prevention 12.5.1 or previous) ID of the incident whose details you want to retrieve. You may include any number of incident IDs in the request.

Table A-2 IncidentDetailRequest fields *(continued)*

Name	Type	Description
`includeViolations`	Boolean	This optional element indicates whether the Web Service should return policy violation data with the basic incident details. A single message may violate multiple policies, and additional fields are added to the response for each policy violation. Each Symantec Data Loss Prevention component logs different violation data. For example, the `NetworkIncidentDetailType` returns a violating header, body, or file attachment component as part of the violation data. See "About incident detail types" on page 26.
`includeHistory`	Boolean	This optional element indicates whether the Web Service should return incident history information with the basic incident details.

Outputs

Returns an `IncidentDetailResponse` document. The response includes an `IncidentDetailType` object for each `incidentId` requested. The `IncidentDetailType` describes the common features that are shared by all Symantec Data Loss Prevention incidents. A client may choose to cast the `IncidentDetailType` to a product group detail type or a specific product component detail type to access unique features of the incident.

Table A-3 Incident Detail Response

Name	Type	Description
`incidentLongId`	long	The incident ID of the requested incident (Symantec Data Loss Prevention 12.5.2 or subsequent).

Table A-3 Incident Detail Response *(continued)*

Name	Type	Description
`incidentId`	int	The incident ID of the requested legacy incident (Symantec Data Loss Prevention 12.5.1 or previous). If the incident ID exceeds the maximum integer value, the `IncidentDetailResponse` will return a value of 0 for this field.
`statusCode`	string	The status code returns the status of the incident detail request. The status code contains one of the following values: ■ `SUCCESS` ■ `AUTHORIZATION_FAILED` ■ `INCIDENT_NOT_FOUND`
`IncidentDetailType`	incident	Contains a data structure that returns the details for the requested incident. See "About incident detail types" on page 26. See "About extended incident detail types" on page 71.

See "About incident detail types" on page 26.

See "About extended incident detail types" on page 71.

Example

The following example shows how a Java client retrieves the details of a single incident with `incidentDetail()`.

Note: The Java code in this section is for instructional purposes only and does not demonstrate a complete client implementation. The sample code is available at the Symantec Support Center at

http://symantec.com/docs/DOC9265

```
int incidentId=1;
```

```
boolean includeHistory = true;
boolean includeViolations = true;
IncidentDetailRequest request = new IncidentDetailRequest();
request.getIncidentId().add(incidentId);
request.setIncludeHistory(includeHistory);
request.setIncludeViolations(includeViolations);

try
{
IncidentServicePortType clientPort = client.getPortClient();
IncidentDetailResponse response = clientPort.incidentDetail(request);

 for(IncidentDetailResult result: response.getResponse())
 {
  System.out.println("-----------------------------------------------------------");
  System.out.println("  Incident details for the incident id: "+ result.getIncidentId());
  System.out.println("-----------------------------------------------------------\n");
  System.out.println("The Response status code is: "+result.getStatusCode());
  IncidentDetailType incidentDetails = result.getIncident();
  System.out.println("Incident creation date=" + incidentDetails.getIncidentDetectionDate());

 }

}
catch (AuthenticationFailedFault ex)
{
  System.out.println("User or password is not valid.");
}
catch (AuthorizationFailedFault ex)
{
  System.out.println("User is not authorized for this action.");
}
catch (Exception ex)
{
  System.out.println(ex.getMessage());
}
```

Faults

The `incidentDetail` call return the following general faults:

- `ServiceErrorFault`
- `AuthenticationFailedFault`

- `AuthorizationFailedFault`

incidentBinaries()

`incidentBinaries()` — retrieves additional components of the message that generated an incident, such as the message header, body, and binary attachments.

Syntax

```
IncidentBinariesResponse =
   IncidentServicePortType.incidentBinaries(IncidentBinariesRequest);
```

Inputs

This call takes a single `IncidentBinariesRequest` type as its argument. `IncidentBinariesRequest` encapsulates the ID of the incident for which to retrieve additional components. The request can also indicate whether the response document should include the original message, all message components, or a specific message component in the response.

Table A-4 IncidentBinariesRequest fields

Name	Type	Description
`incidentLongId`	`long`	This required field identifies the unique ID of the Symantec Data Loss Prevention 12.5.2 or later incident whose components you want to retrieve.
`incidentId`	`int`	This required field identifies the unique legacy (Symantec Data Loss Prevention 12.5.1 or previous) ID of the incident whose components you want to retrieve. If the incident ID exceeds the maximum integer value, the `IncidentBinariesResponse` will return a value of 0 for this field.
`includeOriginalMessage`	`Boolean`	This optional element indicates whether the Web Service should include the original message in the response document.

Table A-4 IncidentBinariesRequest fields *(continued)*

Name	Type	Description
`includeAllComponents`	`Boolean`	This optional element indicates whether the Web Service should include all message components (for example, headers and file attachments) in the response document.
`componentLongId`	`long`	This optional element indicates a specific message component ID to return in the response document. Each message component has a unique component ID. The component long ID is used in incidents created by Symantec Data Loss Prevention 12.5.2 or later
`componentId`	`int`	This optional element indicates a specific message component ID to return in the response document. Each message component has a unique component ID. The component integer ID is used in incidents created by Symantec Data Loss Prevention 12.5.1 or previous. If the component ID exceeds the maximum integer value, the `IncidentBinariesResponse` will return a value of 0 for this field.

Example

The following example shows how a Java client retrieves a list of incidents with `incidentBinaries()`.

Note: The Java code in this section is for instructional purposes only and does not demonstrate a complete client implementation. The sample code is available at the Symantec Support Center at:

http://symantec.com/docs/DOC9265

```
int incidentId=1;
boolean getAllComponents=true;
```

```
int componentId = id;
boolean getOriginalMessage = true;


IncidentBinariesRequest request = new IncidentBinariesRequest();
request.setIncidentId(incidentId);
request.setIncludeAllComponents(getAllComponents);
if(getAllComponents)
{
   request.setComponentId(componentId);
}
request.setIncludeOriginalMessage(getOriginalMessage);
request.setIncludeAllComponents(getAllComponents);

try
{
 IncidentServicePortType clientPort = client.getPortClient();
 IncidentBinariesResponse response = clientPort.incidentBinaries(request);
 if(response.getComponent()!=null)
 {
   for(IncidentBinariesResponse.Component responseComponent : response.getComponent())
   {
    //downloading all the components including the email body text as an attachment
    File file = new File(".");
    saveFile(responseComponent.getContent().getInputStream(), file.getCanonicalPath() + \
       "\\"+responseComponent.getName());
    System.out.println("Binary component Name: "+responseComponent.getName());
    System.out.println("Binary component ID: "+responseComponent.getComponentId());
    System.out.println("Binary component Type: "+responseComponent.getComponentType());
    System.out.println("Saved the binary conent to file: "+ file.getCanonicalPath() + \
       "\\"+responseComponent.getName());
    }
  }
  if(Boolean.parseBoolean(getOriginalMessage)
  {
   File file = new File(".");
   saveFile(response.getOriginalMessage().getInputStream(), file.getCanonicalPath() + \
        "\\"+"original_message");
   System.out.println("Saved the orignal message to: "+ file.getCanonicalPath()+"\\"+ \
        "original_message");
  }
}
```

```
catch(Exception exp)
{
  System.out.println(exp.getMessage());
  exp.printStackTrace();
}

...


private void saveFile(InputStream stream, String fileName) throws IOException
{
  // code to write contents of the input stream to a specified file name
}
```

Faults

The `incidentBinaries` call can return the following general faults:

- `ServiceErrorFault`
- `AuthenticationFailedFault`
- `AuthorizationFailedFault`
- `IncidentNotFoundFault`
- `ComponentNotFoundFault`
- `InvalidRequestFault`

listCustomAttributes()

`listCustomAttributes()`—returns a list of all custom attribute names defined in the Symantec Data Loss Prevention deployment.

Syntax

```
CustomAttributeList attributeList = servicePort.listCustomAttributes();
```

Inputs

This method call takes no arguments.

Outputs

Returns a `customAttributeList` object that contains a list of custom attribute names. You define custom attributes in the Enforce Server administration console. See "About custom attributes" in the *Symantec Data Loss Prevention Administration Guide*.

Examples

Note: The Java code in this section is for instructional purposes only and does not demonstrate a complete client implementation. The sample code is available at the Symantec Support Center at:

http://symantec.com/docs/DOC9265

```
String serviceNamespace ="http://www.company.com/v2011/enforce/webservice/incident";
QName serviceQName = new QName(serviceNamespace,"IncidentService");

URL serviceWsdlUrl = new URl(wsdlLocation);
// The location should be the complete URL starting with https.
// For example: https//enforce_server/ProtectManager/services/v2011/incidents?wsdl

IncidentService webService = new IncidentService(serviceWsdlUrl, serviceQName);
IncidentServicePortType servicePort = webService.getIncidentServicePortType();
CustomAttributeList attributeList = servicePort.listCustomAttributes();
```

Faults

- `AuthenticationFailedFault`
- `AuthorizationFailedFault`
- `ServiceErrorFault`

listIncidentStatus()

`listIncidentStatus()`—returns a list of the custom status values defined in the Symantec Data Loss Prevention deployment.

Syntax

```
IncidentStatusList statusList = servicePort.listIncidentStatus();
```

Inputs

The `listIncidentStatus()` method call takes no arguments.

Outputs

Returns a `incidentStatusList` object that contains a list of status values. You define status values in the Enforce Server administration console. See "Configuring status attributes and values" in the *Symantec Data Loss Prevention Administration Guide*.

Example

Note: The Java code in this section is for instructional purposes only and does not demonstrate a complete client implementation. The sample code is available at the Symantec Support Center at:

http://symantec.com/docs/DOC9265

```
String serviceNamespace ="http://www.company.com/v2011/enforce/webservice/incident";
QName serviceQName = new QName(serviceNamespace,"IncidentService");

URL serviceWsdlUrl = new URL(wsdlLocation);

// location should be the complete url starting with https
// For example: https//enforce_server/ProtectManager/services/v2011/incidents?wsdl

IncidentService webService = new IncidentService(serviceWsdlUrl, serviceQName);
IncidentServicePortType servicePort = webService.getIncidentServicePortType();
IncidentStatusList statusList = servicePort.listIncidentStatus();
```

Faults

- `AuthenticationFailedFault`
- `AuthorizationFailedFault`
- `ServiceErrorFault`

updateIncidents()

`updateIncidents()`—updates incident details for one or more incidents.

Syntax

```
IncidentUpdateResponse response = servicePort.updateIncidents(request);
```

Inputs

Each invocation of the `updateIncidents()` method defines one or more batches that each contain a list of one or more incident IDs to be updated. Each batch also defines the fields and values to update. Only the fields that are specified in the request are updated. To reset the value of a field, set the new value to a blank string.

Using this batching logic, you can update multiple incidents with the same values. For example, in a single call to the `updateIncidents()` method, you can set all of the incidents IDs specified in the batch to a new status value. You can also specify multiple batches in a single call. Each batch can specify different incident IDs and values to update.

For best performance, Symantec recommends that you update multiple incidents in a single batch wherever possible. Updating 50 – 500 incidents per batch provides the best performance.

Incidents are not locked during updates. "Optimistic" locking is used and the most recent update operation sets the values of the fields. The response to this method call contains a status message that describes whether or not all the updates succeeded for each batch. An error is returned when an incident ID is not found.

Table A-5 `updateIncidents()` instance variables

Name	Type	Description
`batchID`	`string`	The required `IncidentUpdateBatch` field is available for use by client applications. You can use this field to track batches of incident updates. The field is not used by Symantec Data Loss Prevention. Symantec recommends that you use a unique identifier such as a GUID to track batches. An incident update call can contain multiple batches that each contain a list of incident IDs. Each batch also contains a set of incident fields and values that are to be updated for all incidents in that batch.
`incidentLongId`	`long`	This required field identifies the unique ID of the Symantec Data Loss Prevention 12.5.2 or later incident whose details you want to retrieve. You may include any number of incident IDs in the request.

Table A-5 `updateIncidents()` instance variables *(continued)*

Name	Type	Description
`incidentId`	`int`	This required field identifies the unique legacy (Symantec Data Loss Prevention 12.5.1 or previous) ID of the incident whose details you want to retrieve. You may include any number of incident IDs in the request.
`incidentAttributes`	`IncidentAttributes`	The `IncidentAttributes` object represents the set of attributes and values to be updated for each batch. See Table A-6.

Table A-6 IncidentAttributes types and values

Name	Type	Description
`IncidentSeverity`	`string`	Represents the severity level of the incident. The possible values are: ■ `HIGH` ■ `MEDIUM` ■ `LOW` ■ `INFO` The `IncidentSeverity` field is not case sensitive.
`IncidentStatus`	`string`	Represents the status value of the incident. You define incident status values using the Enforce Server administration console. See "Configuring status attributes and values" in the *Symantec Data Loss Prevention Administration Guide*.
`IncidentNote`	`IncidentNote` ■ `dateAndTime` (type: `dateTime`) ■ `note` (type: `string`)	Represents notes that are stored with an incident. You define both the date and the text of the note.

Table A-6 IncidentAttributes types and values *(continued)*

Name	Type	Description
`CustomAttribute`	`CustomAttributeType` ■ `name` (type: `string`) ■ `value` (type: `string`)	Represents custom attributes that are associated with an incident. You define zero or more custom attributes using the Enforce Server administration console. To update a specific custom attribute, reference its name with the `name` attribute and its value with the `value` attribute.
`DataOwner`	`DataOwnerType` ■ `name` (type: `string`) ■ `email` (type: `string`)	Represents the owner of the data that caused a policy violation.

Table A-6 IncidentAttributes types and values *(continued)*

Name	Type	Description
RemediationStatus	string	Represents the remediation status of an incident. You can set the following values: ■ BLOCKED ■ CONTENT_REMOVED ■ CUSTOM_ACTION_ON_EMAIL ■ EMAIL_APPROVED ■ EMAIL_BLOCKED ■ EMAIL_MESSAGE_EXPUNGED ■ EMAIL_QUARANTINED ■ ENDPOINT_BLOCK ■ ENDPOINT_NOTIFY ■ ENDPOINT_FILE_QUARANTINED ■ ENDPOINT_FILE_QUARANTINE_FAILED ■ ENDPOINT_NOTIFY_CANCEL_ALLOW ■ ENDPOINT_NOTIFY_CANCEL_BLOCK ■ ENDPOINT_NOTIFY_CANCEL_TIMEOUT_ALLOW ■ ENDPOINT_NOTIFY_CANCEL_TIMEOUT_BLOCK ■ FLEX_RESPONSE_ERROR ■ FLEX_RESPONSE_EXECUTED ■ FLEX_RESPONSE_REQUESTED ■ MESSAGE_MODIFIED ■ PASSED ■ PLACE_HOLDER_DO_NOT_USE ■ PROTECT_FILE_COPIED ■ PROTECT_FILE_DELETED ■ PROTECT_FILE_QUARANTINED ■ PROTECT_REMEDIATION_ERROR ■ REST_ENCRYTPED ■ REST_PERFORMED_DRM ■ REST_PERFORMED_BREAK_LINKS ■ REST_PERFORMED_CUSTOM_ACTION
RemediationLocation	string	Represents the remediation location of the incident. Values can be user-defined.

Outputs

IncidentUpdateBatchResult [batchID,InaccessibleIncidentId,StatusCode]

Returns an `IncidentUpdateResponse` object.

The response to this method contains a separate data structure for each batch that was sent in the request. The data structure contains the fields described in Table A-7.

Table A-7 IncidentUpdateResponse fields

Name	Type	Description
`batchID`	`string`	The batch number that is defined in the update request.
`InaccessibleIncidentId`	`int`	Contains the incident ID of incidents for which an update was requested but the incident cannot be found.
`statusCode`	`string`	The status code indicates the success or failure of the requested updates of each batch. The possible values are: ■ `SUCCESS` ■ `PARTIAL SUCCESS` ■ `VALIDATION ERROR` ■ `ALL_INCIDENTS_INACCESSIBLE_OR_DELETED` ■ `AUTHORIZATION_FAILED` ■ `SQL_ERROR`

Example

Note: The Java code in this section is for instructional purposes only and does not demonstrate a complete client implementation. The sample code is available at the Symantec Support Center at:

http://symantec.com/docs/DOC9265

```
IncidentUpdateRequest request = new IncidentUpdateRequest();
IncidentUpdateBatch batch = new IncidentUpdateBatch();
batch.setBatchId("_" + UUID.randomUUID().toString());
List<IncidentUpdateBatch> actions = request.getUpdateBatch();
actions.add(batch);

List<Integer> incidentIDs = batch.getIncidentId();
incidentIDs.add("IncidentId");
//User provides incident ID

IncidentAttributes attributes =  new IncidentAttributes();
batch.setIncidentAttributes(attributes);
```

```
IncidentStatusType status = new IncidentStatusType();
status.setValue("Value");
//User provides the value to be updated

attributes.setStatus(status);

IncidentServicePortType servicePort = client.getPortClient();
IncidentUpdateResponse response = servicePort.updateIncidents(request);
```

Faults

- `AuthenticationFailedFault`
- `AuthorizationFailedFault`
- `ServiceErrorFault`

incidentViolations()

incidentViolations() returns the highlighted matches for an incident.

Syntax

```
IncidentViolationsResponse = IncidentServicePortType.
     incidentViolations(IncidentViolationsRequest);
```

Inputs

This call takes a single `IncidentViolationsRequest` type as its argument. `IncidentViolationsRequest` encapsulates the incident ID of the incident.

Table A-8 IncidentViolations instance variable

Name	Type	Description
`incidentLongId`	`long`	This required field identifies the unique ID of the Symantec Data Loss Prevention 12.5.2 or later incident whose violations you want to retrieve. You may include any number of incident IDs in the request.

Table A-8 IncidentViolations instance variable *(continued)*

Name	Type	Description
`incidentId`	`int`	This required field identifies the unique legacy (Symantec Data Loss Prevention 12.5.1 or previous) ID of the incident whose violations you want to retrieve. You may include any number of incident IDs in the request.
`includeImageViolations`	`Boolean`	This optional element specifies whether image violations should be included in the `IncidentViolationsResponse`.

Output

Returns an `IncidentViolationsResponse` object that includes all matches in the incident.

Example

Figure A-1 shows an `incidentViolationRequest` object. The request includes one or more incident IDs.

Figure A-1 incidentViolationsRequest example

```
– <soapenv:Envelope>
    <soapenv:Header/>
  – <soapenv:Body>
    – <sch:incidentViolationsRequest>
        <!--Zero or more repetitions:-->
        <sch:incidentId>00001874</sch:incidentId>
      </sch:incidentViolationsRequest>
    </soapenv:Body>
  </soapenv:Envelope>
```

The response object returns a text or file size or document violation.

Figure A-2 shows an `incidentViolationResponse` object with violating text.

Figure A-2 incidentViolationsResponse example

```
- <S:Envelope>
  - <S:Body>
    - <ns4:incidentViolationsResponse>
      - <ns4:incidentViolation>
          <ns4:incidentId>1874</ns4:incidentId>
          <ns4:statusCode>SUCCESS</ns4:statusCode>
        - <ns4:violatingComponent id="451251">
            <ns4:name>LICENSE.txt</ns4:name>
            <ns4:documentFormat>ascii</ns4:documentFormat>
            <ns4:violatingComponentType id="3">Attachment</ns4:violatingComponentType>
            <ns4:violationCount>3</ns4:violationCount>
          - <ns4:violatingSegment>
              <ns4:text type="NonViolation">...programs constitute</ns4:text>
              <ns4:text type="Violation" ruleId="101" ruleName="Keyword">confidential</ns4:text>
            - <ns4:text type="NonViolation">
                proprietary information of eviware Soft...ovide, or otherwise make available such
              </ns4:text>
              <ns4:text type="Violation" ruleId="101" ruleName="Keyword">confidential</ns4:text>
            - <ns4:text type="NonViolation">
                information in any form to any third pa...nable security measures to protect such
              </ns4:text>
              <ns4:text type="Violation" ruleId="101" ruleName="Keyword">confidential</ns4:text>
              <ns4:text type="NonViolation">information, but wi...</ns4:text>
            </ns4:violatingSegment>
          </ns4:violatingComponent>
        </ns4:incidentViolation>
      </ns4:incidentViolationsResponse>
    </S:Body>
  </S:Envelope>
```

The following example shows an `incidentViolationResponse` object with violating file size:

```
<ns4:violatingSegment>
   <ns4:fileSizeViolation ruleId="1101" ruleName="File Size Rule">
       <ns3:violatingFileSize>3310</ns3:violatingFileSize>
       <ns3:units>bytes</ns3:units>
   </ns4:fileSizeViolation>
</ns4:violatingSegment>
```

The following example shows an `incidentViolationResponse` object with violating document (line breaks added for legibility):

```
<ns4:violatingSegment>
    <ns4:documentViolation ruleId="1001" ruleName="IDM Rule">
        <ns3:documentProfileName>IDM Profile</ns3:documentProfileName>
        <ns3:documentPath>C:\Symantec\DataLossPrevention\ServerPlatformCommon\15.5\Protect\
                          documentprofiles\loremipsum-101\
                          loremipsum.txt</ns3:documentPath>
        <ns3:matchPercentage>100</ns3:matchPercentage>
    </ns4:documentViolation>
</ns4:violatingSegment>
```

Appendix B

Base Incident Detail Types

This appendix includes the following topics:

IncidentDetailType

`IncidentDetailType` – defines the common fields that are shared by all Symantec Data Loss Prevention incidents.

Base fields

`IncidentDetailType` defines the following fields.

Note that the exact XML fields returned in the `IncidentDetailResponse` document depend on the role-based access controls for the Web Service client user. For example, custom attribute elements are returned only if you explicitly enable that permission for the role to which the Web Service client belongs.

See "Creating a user and role for an Incident Reporting and Update API client" on page 16.

Table B-1 IncidentDetailType fields

Field	Type	Occurrences	Description
`incidentLongId`	`long`	1	The incident ID of the requested incident (Symantec Data Loss Prevention 12.5.2 or subsequent).
`incidentId`	`int`	1	The incident ID of the requested legacy incident (Symantec Data Loss Prevention 12.5.1 or previous). If the incident ID exceeds the maximum integer value, the `IncidentDetailType` will return a value of 0 for this field.
`incidentCreationDate`	`datetime`	1	The date and time when the incident was added to the Enforce Server database. Products such as Endpoint Prevent may create an incident some time after the actual violation occurs. This can happen when endpoints are disconnected from the network.
`detectionDate`	`datetime`	1	The date and time at which the Symantec Data Loss Prevention software detected the incident.

Table B-1 IncidentDetailType fields *(continued)*

Field	Type	Occurrences	Description
`severity`	`IncidentSeverityType` (string)	1	The severity label of the incident. This field also contains an integer value `severityId` attribute that corresponds to the severity of the incident.
`status`	`IncidentStatusType` (string)	1	The status label of the incident. This field also contains an integer value `statusId` attribute that corresponds to the incident status.
`messageSource`	`MessageSource` (string)	1	The localized label that corresponds to the Symantec Data Loss Prevention product that generated the incident. This field also contains a string value `sourceType` attribute that indicates the Symantec Data Loss Prevention product that generated the incident. `sourceType` can be one of: ■ `NETWORK`—Network Monitor, Network Prevent for Email, or Network Prevent for Web ■ `DISCOVER`—Network Discover or Network Protect ■ `ENDPOINT`—Endpoint Discover or Endpoint Prevent ■ `DIM` ■ `DAR`
`messageType`	`MessageType` (string)	1	Contains a string value (and integer value typeId attribute) that corresponds to the Symantec Data Loss Prevention product component that generated the incident. Client applications can use the `messageType` value to cast the basic `IncidentDetailType` into a sub-type of the product group or product component that generated the incident. These sub-types provide additional fields unique to the group or component. See "About incident detail types" on page 26.

Table B-1 IncidentDetailType fields *(continued)*

Field	Type	Occurrences	Description
`policy`	`PolicyType`	1	Describes the policy that the message violated to generate the incident. If a message violates multiple policies, additional policies are described in the `otherViolatedPolicy` field. The `PolicyType` contains a `policyId` attribute. This integer attribute uniquely identifies the policy in the Enforce Server administration console. `PolicyType` also contains a name and version element to describe the violated policy further.
`violatedPolicyRule`	`PolicyRuleType`	1–many	Describes the exact rule(s) within the policy that the message violated. A single policy can define many rules, and a given message can potentially violate each of the policy rules. The PolicyRuleType contains a `ruleId` attribute. This integer attribute uniquely identifies the policy in the Enforce Server administration console. PolicyRuleType also contains the name of the rule in a `ruleName` element.
`otherViolatedPolicy`	`PolicyType`	0–many	Describes any additional policies that the message violated. See the description of policy above.
`blockedStatus`	`BlockedStatusType` (string)	1	A string value that indicates whether the message was blocked. This field also contains an integer value `blockedStatusId` attribute that corresponds to the incident status.

Table B-1 IncidentDetailType fields *(continued)*

Field	Type	Occurrences	Description
`matchCount`	`int`	1	Indicates the number of detection rule matches in this incident. The exact meaning of the `matchCount` field depends on the criteria used to detect the incident. For example, if a policy rule uses a pattern to detect incidents, `matchCount` indicates the number of pattern matches that were found. If a rule uses a file type or file size criterion to detect incidents, the `matchCount` value is 1 if the file type or size is detected.
`ruleViolationCount`	`int`	0–1	Indicates the number of policy rules that were violated. This field is included only when the client requests violation data with the `includeViolations` field in the `incidentDetail()` request.
`detectionServer`	`string`	1	The name of the detection server that created the incident.
`customAttributeGroup`	`CustomAttributeGroupType`	0–many	One or more of these elements are present when custom attributes are returned in the incident detail. Custom attributes are optional data fields that you can use to store supplemental information about an incident. Your organization may use custom attributes to assist in the workflow for remediating or evaluating incidents. See the *Symantec Data Loss Prevention Administration Guide* for more information about custom attributes. See "Creating a user and role for an Incident Reporting and Update API client" on page 16.

Table B-1 IncidentDetailType fields *(continued)*

Field	Type	Occurrences	Description
`incidentHistory`	`IncidentHistoryEntryType`	0–many	One or more of these elements are present when you request incident history to be included in an `incidentDetail` call. The incident history records changes to the incident status or severity, as well as any changes enacted by response rules.
`dataOwner`	`DataOwnerType`	0-1	The `dataOwner` field includes a sequence of names and email addresses that describe the people who are responsible for remediating this incident. Each name and email address must be configured manually, or with a lookup plug-in. See the *Symantec Data Loss Prevention Administration Guide* for more information.

The following table describes the `messageType` string values and integer ID values associated with the `messageType` field. Note that some string values, such as HTTP and FTP, can describe either a network product category or endpoint product category. Use the `typeId` attribute value or examine the `messageSource` field to determine the exact component type.

Table B-2 `messageType` integer values

String value	typeId attribute value	Product category type	Product component type
SMTP	2	`NetworkIncidentDetailType`	`NetworkEmailIncidentDetail`
HTTP	3	`NetworkIncidentDetailType`	`NetworkHTTPIncidentDetail`
FTP	4	`NetworkIncidentDetailType`	`NetworkFTPIncidentDetail`
NNTP	5	`NetworkIncidentDetailType`	`NetworkNNTPIncidentDetail`
MSN	6	`NetworkIncidentDetailType`	`EndpointIMIncidentDetail`
AIM	7	`NetworkIncidentDetailType`	`EndpointIMIncidentDetail`
YAHOO	8	`NetworkIncidentDetailType`	`EndpointIMIncidentDetail`

Table B-2 `messageType` integer values *(continued)*

String value	typeId attribute value	Product category type	Product component type
Filesystem	9	`DiscoverIncidentDetailType`	`DiscoverFileSystemIncidentDetail`
SSL	10	`NetworkIncidentDetailType`	`NetworkHTTPIncidentDetail`
LotusNotes	11	`DiscoverIncidentDetailType`	`DiscoverLotusNotesIncidentDetail`
Removable Storage	13	`EndpointIncidentDetailType`	`EndpointLocalFileSystemIncidentDetail`
Local Drive	14	`EndpointIncidentDetailType`	`EndpointLocalFileSystemIncidentDetail`
EndpointFileSystem	15	DiscoverIncidentDetailType	`DiscoverFileSystemScannerIncidentDetail`
WebServerScanner	18	`DiscoverIncidentDetailType`	`DiscoverWebServerScannerIncidentDetail` The WebServerScanner target was deprecated in Symantec Data Loss Prevention 14.5.
FileSystemScanner	19	`DiscoverIncidentDetailType`	`DiscoverFileSystemScannerIncidentDetail` The FileSystemScanner target was deprecated in Symantec Data Loss Prevention 14.5.
LiveLinkScanner	20	`DiscoverIncidentDetailType`	`DiscoverLivelinkScannerIncidentDetail`
DocumentumScanner	21	`DiscoverIncidentDetailType`	`DiscoverDocumentumScannerIncidentDetail`
GenericScanner	22	`DiscoverIncidentDetailType`	`DiscoverGenericScannerIncidentDetail`
WebServices	23	`DiscoverIncidentDetailType`	`DiscoverWebServiceIncidentDetail`
CD/DVD	24	`EndpointIncidentDetailType`	`EndpointLocalFileSystemIncidentDetail`
SQLDatabase	25	`DiscoverIncidentDetailType`	`DiscoverSQLDatabaseIncidentDetail`
Email/SMTP	26	`EndpointIncidentDetailType`	`EndpointEmailIncidentDetail`
HTTP	27	`EndpointIncidentDetailType`	`EndpointHTTPIncidentDetail`
HTTPS/SSL	28	`EndpointIncidentDetailType`	`EndpointHTTPIncidentDetail`
FTP	29	`EndpointIncidentDetailType`	`EndpointFTPIncidentDetail`
IM:MSN	30	`EndpointIncidentDetailType`	`EndpointIMIncidentDetail`

Table B-2 `messageType` integer values *(continued)*

String value	typeId attribute value	Product category type	Product component type
IM:AIM	31	`EndpointIncidentDetailType`	`EndpointIMIncidentDetail`
IM:Yahoo	32	`EndpointIncidentDetailType`	`EndpointIMIncidentDetail`
Copy to Network Share	33	`EndpointIncidentDetailType`	`EndpointNetworkIncidentDetailType` **Note:** This release of Symantec Data Loss Prevention does not create incidents of this type.
Printer/Fax	34	`EndpointIncidentDetailType`	`EndpointPrintFaxIncidentDetail`
Endpoint clipboard	35	`EndpointIncidentDetailType`	`EndpointClipboardIncidentDetail`
SharePoint Crawler	36	`DiscoverIncidentDetailType`	`DiscoverSharePointCrawlerIncidentDetail`
Exchange	37	n/a	This `messageType` is generated by the Data Classification for Enterprise Vault solution, which is licensed separately from Symantec Data Loss Prevention. There is no product component type available for this `messageType`, and you cannot use the Incident Reporting and Update API to access classification results.
Exchange crawler	38	`DiscoverIncidentDetailType`	`DiscoverExchangeCrawlerIncidentDetail`
Box Crawler	43	`DiscoverIncidentDetailType`	`DiscoverBoxCrawlerIncidentDetail`
n/a	n/a	n/a	`EndpointNNTPIncidentDetail` is not implemented in this release.
Custom protocol	1000 or greater	`NetworkIncidentDetailType`	`NetworkUniversalIncidentDetailType`
REST Data-in-Motion (DIM)	48	`RestIncidentDetailType`	`RestDIMIncidentDetail`
REST Data-at-Rest (DAR)	49	`RestIncidentDetailType`	`RestDARIncidentDetail`

NetworkIncidentDetailType

`NetworkIncidentDetailType` – defines the common properties shared by all incidents in the Network product group (Network Monitor, Network Prevent for Email, and Network Prevent for Web).

Base fields

`NetworkIncidentDetailType` inherits all of the base fields present in `IncidentDetailType`.

Extension fields

`NetworkIncidentDetailType` extends `IncidentDetailType` by adding the following fields:

Table B-3 NetworkIncidentDetailType fields

Field	Type	Occurrences	Description
`messageDate`	`datetime`	1	The date and time at which the network message (for example, an email message, HTTP request, instant message, or other protocol request) was created.
`originator`	`NetworkOriginatorType`	0 - 1	Details about the sender of the network message, including the sender's IP address and port number, as well as an identifying string.
`recipient`	`NetworkRecipientType`	0 - many	Details about the intended recipient of the network message, including the recipient's IP address and port number, as well as an identifying string.
`messageBodyContent`	`string`	0 - 1	The full body text of the message that generated the incident.
`messageHeader`	`MessageComponentType`	0 - 1	The header text of the original message. For example, this field includes the subject header for incidents created by Network Prevent for Email. This field is provided only when you choose to include violation data in the incident detail request.

Table B-3 NetworkIncidentDetailType fields *(continued)*

Field	Type	Occurrences	Description
`messageBody`	`MessageComponentType`	0 - 1	The partial body text that violated the policy. This field is provided only when you choose to include violation data in the incident detail request, and only when violation content appears in the message body.
`file`	`MessageComponentType`	0 - many	The file that generated the incident. For example, this field might describe a file that was transmitted over FTP or a file attachment to an email message. This field is provided only when you choose to include violation data in the incident detail request.

DiscoverIncidentDetailType

`DiscoverIncidentDetailType` – defines the common properties shared by all Network Discover/Cloud Storage Discover products.

Base fields

`DiscoverIncidentDetailType` inherits all of the base fields present in `IncidentDetailType`.

Extension fields

`DiscoverIncidentDetailType` extends `IncidentDetailType` by adding the following fields:

Table B-4 DiscoverIncidentDetailType fields

Field	Type	Occurrences	Description
`targetServer`	`string`	0 - 1	The name of the Network Discover/Cloud Storage Discover Server that performed the scan.
`scan`	`scanAssignmentType` (datetime)	1	The complex type `scanAssignmentType` specifies the date and time that the scan started. `scanAssignmentType` also contains a `scanId` attribute that specifies the integer ID of the scan.
`contentRootPath`	`string`	1	The name of the file share, server, or SQL database that was scanned.
`target`	`string`	0 - 1	The name of the configured Network Discover/Cloud Storage Discover target.
`URL`	`string`	0 - 1	The URL or file path associated with a scan target.
`seenBefore`	`string`	0 - 1	This field indicates whether the incident was previously detected (Yes or No).
`superseded`	`string`	1	This field indicates whether the incident response was superseded by another response (Yes or No).
`remediationLocation`	`string`	0-1	The location where the file was copied or quarantined.

EndpointIncidentDetailType

`EndpointIncidentDetailType` – defines the common properties shared by the Endpoint Prevent and Endpoint Discover products.

Base fields

`EndpointIncidentDetailType` inherits all of the base fields present in `IncidentDetailType`.

Extension fields

`EndpointIncidentDetailType` extends `IncidentDetailType` by adding the following fields:

Table B-5

Field	Type	Occurrences	Description
`eventDate`	`datetime`	1	The date and time at which the violation occurred. This may be different from the time at which the actual incident was created in Symantec Data Loss Prevention.
`applicationName`	`string`	0 - 1	The name of the application that caused the incident.
`applicationPath`	`string`	0 - 1	The full path to the application that caused the incident.
`userName`	`string`	0 - 1	The endpoint user name (for example, MYDOMAIN\bsmith).
`machineName`	`string`	0 - 1	The computer on which the incident occurred.
`userJustification`	`string`	0 - 1	The justification label followed by the text presented to the end user in the on-screen notification (for example, Manager Approved: "My manager approved the transfer of this data."). Symantec Data Loss Prevention uses the label for classification and filtering purposes in reports.
`superseded`	`string`	1	This field indicates whether the incident response was superseded by another response (Yes or No).

RestIncidentDetailType

`RestIncidentDetailType` – defines the common properties that are shared by all Cloud Connector incidents.

Base fields

`RestIncidentDetailType` for both DIM (data in motion) and DAR (data at rest) inherits all of the base fields present in `IncidentDetailType`.

Extension fields

`RestIncidentDetailType` extends `IncidentDetailType` by adding the following attributes:

Note: All attributes are for both DIM and DAR, unless otherwise noted in the Description column. All attributes are strings.

Table B-6 RestIncidentDetailType fields

Extension field	REST attribute	Occurrences	Description
`requestId`	`common.request.id`	1	Request ID.
`applicationName`	`common.application`	0-1	Application name.
`deviceOS`	`device.os`	0-1	OS of the device.
`isTrustedDevice`	`device.isTrustedDevice`	0-1	String value (true or false) indicating if the request came from a trusted device.
`httpCookies`	`http.cookies`	0-1	Cookies from HTTP requests.
`httpMethod`	`http.method`	0-1	Method for detecting requests related to HTTP traffic.
`httpUrl`	`http.url`	0-1	URL for detection requests related to HTTP traffic.
`httpUserAgent`	`http.userAgent`	0-1	User agent for detection requests related to HTTP traffic.

Table B-6 RestIncidentDetailType fields *(continued)*

Extension field	REST attribute	Occurrences	Description
`locationCoordsLatitude`	`location.coords.latitude`	0-1	Geographic location of the device: latitude.
`locationCoordsLongitude`	`location.coords.longitude`	0-1	Geographic location of the device: longitude.
`networkDirection`	`network.direction`	0-1	Acceptable values: `Download`, `Upload`.
`networkProtocol`	`network.protocol`	0-1	OSI L7 network protocol applicable to the detection request: SMTP, HTTP, FTP, and so on.
`restCustom`	custom	0-1	A custom context attribute (Custom context attributes differ from the other well-known context attributes in their treatment by the policy engine.)
`fileTag`	`common.tag`	0-1	File tag field.
`fileFolder`	`common.folder`	0-1	Path where the files and attachments reside.
`LocationIsInsideOffice`	`location.isInsideOffice`	0-1	String value (true or false) indicating if the request came from inside of the office.
`httpSiteRiskScore`	`http.siteRiskScore`	0-1	Describes the risk level of the target site.

Appendix C

Extended Incident Detail Types

This appendix includes the following topics:

About extended incident detail types

The following sections provide a reference for the product component incident detail types included in the Incident Reporting and Update API schema. Component detail types extend the base product types (`NetworkIncidentDetailType`, `DiscoverIncidentDetailType`, `EndpointIncidentDetailType`, and `RestIncidentDetailType`) with fields specific to the protocol or product component that generated the incident.

See "About incident detail types" on page 26.

Network component detail types

The Incident Reporting and Update API schema defines six component detail types as extensions to `NetworkIncidentDetailType` to represent the different protocols that Network Prevent can monitor. Note that three of the component types add no additional fields, but are available as placeholders for future extensions.

Table C-1 describes the new fields (if any) added by each network component detail type.

Table C-1 Network component detail types

Component type	Extension field	Field type	Occurrences	Description
NetworkEmailIncidentDetail	`subject`	string	0-1	`NetworkEmailIncidentDetail` adds a `subject` field to hold the subject of the email message that generated the incident.
NetworkHTTPIncidentDetail	`HTTPS`	Boolean	1	`NetworkHTTPIncidentDetail` adds an `HTTPS` field to indicate whether the Web request was transmitted over a secure connection.
NetworkFTPIncidentDetail	n/a	n/a	n/a	This type adds no additional fields to `NetworkIncidentDetailType`. It is provided as a placeholder type for future extension fields.
NetworkNNTPIncidentDetail	`subject`	string	0-1	`NetworkNNTPIncidentDetail` adds a `subject` field to hold the subject of the message that generated the incident.
NetworkIMIncidentDetail	n/a	n/a	n/a	This type adds no additional fields to `NetworkIncidentDetailType`. It is provided as a placeholder type for future extension fields.
NetworkUniversalIncidentDetail	`protocolName`	string	1	`NetworkUniversalIncidentDetail` adds a `protocolName` field to indicate the custom protocol that was used to transmit the incident message.

Discover component detail types

The Incident Reporting and Update API schema defines the following component detail types as extensions to `DiscoverIncidentDetailType`:

- `DiscoverFileSystemIncidentDetail` (Table C-2)
- `DiscoverEndpointFileSystemIncidentDetail` (Table C-3)
- `DiscoverSQLDatabaseIncidentDetail` (Table C-4)
- `DiscoverLotusNotesIncidentDetail` (Table C-5)
- `DiscoverGenericScannerIncidentDetail` (Table C-6)

- `DiscoverFileSystemScannerIncidentDetail` (Table C-7)
- `DiscoverWebServerScannerIncidentDetail` (Table C-8)
- `DiscoverLivelinkScannerIncidentDetail` (Table C-9)
- `DiscoverDocumentumScannerIncidentDetail` (Table C-10)
- `DiscoverWebServiceIncidentDetail` (Table C-11)
- `DiscoverSharePointCrawlerIncidentDetail` (Table C-12)
- `DiscoverExchangeCrawlerIncidentDetail` (Table C-13)
- `DiscoverBoxCrawlerIncidentDetail`(Table C-14)

Each extension type corresponds to the Network Discover/Cloud Storage Discover detection mechanism that logged the original incident. The tables that follow describe the fields that each type adds to `DiscoverIncidentDetailType`.

Table C-2 DiscoverFileSystemIncidentDetail extension fields

Extension field	Field type	Occurrences	Description
`fileName`	`string`	1	The name of the file that violated the policy.
`filePath`	`string`	1	The full path to the file.
`fileLastAccessDate`	`datetime`	0-1	The date and time of the last user access to the file.
`fileLastModifiedDate`	`datetime`	0-1	The date and time when the file was last changed.
`fileCreateDate`	`datetime`	0-1	The date and time when the file was created.
`fileOwner`	`string`	0-1	The owner of the file at the time the incident was created.

Table C-2 DiscoverFileSystemIncidentDetail extension fields *(continued)*

Extension field	Field type	Occurrences	Description
`fileACL`	`ACLType`	0-many	The Access Control Lists for the file. Access Control Lists (ACL) are lists of the permissions that are attached to an object or piece of data. The list contains information about all users who have read and write permissions for the file. Use the list to view which users have access to the file as well as which actions each user can perform. The permissions for each user or group are not set through Symantec Data Loss Prevention. Administrators set the permissions for each file using other applications. Permissions are generally set at the time that the file is created. A separate `fileACL` entry is included for each permission that was granted on the file. `fileACL` entries are only in certain Network Discover, Endpoint Discover, and Endpoint Prevent incidents.
`file`	`messageComponentType`	0-many	A `messageComponentType` entry that encapsulates the entire file component. This field is provided only when you choose to include violation data in the incident detail request.

Table C-3 DiscoverEndpointFileSystemIncidentDetail extension fields

Extension field	Field type	Occurrences	Description
`fileName`	`string`	1	The name of the file that violated the policy.
`filePath`	`string`	1	The full path to the file.
`fileLastModifiedDate`	`datetime`	0-1	The date and time that the file was last changed.
`fileLastAccessDate`	`datetime`	0-1	The date and time of the last user access to the file.

Table C-3 DiscoverEndpointFileSystemIncidentDetail extension fields *(continued)*

Extension field	Field type	Occurrences	Description
`fileCreateDate`	`datetime`	0-1	The date and time when the file was created.
`fileOwner`	`string`	0-1	The owner of the file at the time the incident was created.
`fileACL`	`ACLType`	0-many	The Access Control Lists for the file. Access Control Lists (ACL) are lists of the permissions that are attached to an object or piece of data. The list contains information about all users who have read and write permissions for the file. Use the list to view which users have access to the file as well as which actions each user can perform. The permissions for each user or group are not set through Symantec Data Loss Prevention. Administrators set the permissions for each file using other applications. Permissions are generally set at the time that the file is created. A separate `fileACL` entry is included for each permission that was granted on the file. `fileACL` entries are only in certain Network Discover, Endpoint Discover, and Endpoint Prevent incidents.
`file`	`MessageComponentType`	`0`-many	A `messageComponentType` entry that encapsulates the entire file component. This field is provided only when you choose to include violation data in the incident detail request.

Table C-4 DiscoverSQLDatabaseIncidentDetail extension fields

Extension field	Field type	Occurrences	Description
`body`	`MessageComponentType`	0-1	Partial body text (extracted from the database) that violated the policy. This field is provided under the following conditions: ■ You choose to include violation data in the incident detail request. ■ The violation content appears in the message body.
`bodyContent`	`string`	0-1	Full body text of the message that violated the policy (extracted from the database). This field is provided only when you choose to include violation data in the incident detail request.

Table C-5 DiscoverLotusNotesIncidentDetail extension fields

Extension field	Field type	Occurrences	Description
`documentName`	`string`	1	The name of the file that violated the policy.
`lastModifiedBy`	`string`	0-1	The name of the user who last changed the file.
`createdBy`	`string`	0-1	The creator of the file.
`lastModifiedDate`	`datetime`	0-1	The date and time that the file was last changed.
`createDate`	`datetime`	0-1	The date and time when the file was created.
`bodyContent`	`string`	0-1	Full body text of the message that violated the policy (extracted from the database). This field is provided only when you choose to include violation data in the incident detail request.

Table C-5 DiscoverLotusNotesIncidentDetail extension fields *(continued)*

Extension field	Field type	Occurrences	Description
`body`	`messageComponentType`	0-1	Body text (extracted from the document) that violated the policy. This field is provided only when you choose to include violation data in the incident detail request.
`file`	`messageComponentType`	0-1	A `messageComponentType` entry that encapsulates the entire file component. This field is provided only when you choose to include violation data in the incident detail request.
`fileOwner`	`string`	0-1	The owner of the file at the time the incident was created.
`filePath`	`string`	1	The full path to the file.
`fileACL`	`ACLType`	0-many	The Access Control Lists for the file. Access Control Lists (ACL) are lists of the permissions that are attached to an object or piece of data. The list contains information about all users who have read and write permissions for the file. Use the list to view which users have access to the file as well as which actions each user can perform. The permissions for each user or group are not set through Symantec Data Loss Prevention. Administrators set the permissions for each file using other applications. Permissions are generally set at the time that the file is created. A separate `fileACL` entry is included for each permission that was granted on the file. `fileACL` entries are only in certain Network Discover, Endpoint Discover, and Endpoint Prevent incidents.

Table C-6 DiscoverGenericScannerIncidentDetail extension fields

Extension field	Field type	Occurrences	Description
`fileName`	`string`	1	The name of the file that violated the policy.
`filePath`	`string`	1	The full path to the file.
`bodyContent`	`string`	0-1	Full body text of the message that violated the policy (extracted from the database). This field is provided only when you choose to include violation data in the incident detail request.
`header`	`messageComponentType`	0-1	SQL database metadata that was generated by Symantec Data Loss Prevention. This field is provided only when you choose to include violation data in the incident detail request.
`body`	`messageComponentType`	0-1	Body text (extracted from the database) that violated the policy. This field is provided only when you choose to include violation data in the incident detail request.
`file`	`messageComponentType`	0-1	A `messageComponentType` entry that encapsulates the entire file component. This field is provided only when you choose to include violation data in the incident detail request.
`fileLastModifiedDate`	`datetime`	0-1	The date and time that the file was last changed.

Table C-7 DiscoverFileSystemScannerIncidentDetail extension fields

Extension field	Field type	Occurrences	Description
`fileName`	`string`	1	The name of the file that violated the policy.
`filePath`	`string`	1	The full path to the file.

Table C-7 DiscoverFileSystemScannerIncidentDetail extension fields *(continued)*

Extension field	Field type	Occurrences	Description
`file`	`messageComponentType`	0-many	A `messageComponentType` entry that encapsulates the entire file component. This field is provided only when you choose to include violation data in the incident detail request.
`fileCreateDate`	`datetime`	0-1	The date and time when the file was created.
`fileCreatedBy`	`string`	0-1	The creator of the file.
`fileLastModifiedDate`	`datetime`	0-1	The date and time that the file was last changed.
`fileLastAccessDate`	`datetime`	0-1	The date and time of the last user access to the file.
`fileOwner`	`string`	0-1	The owner of the file at the time the incident was created.
`fileACL`	`ACLType`	0-many	The Access Control Lists for the file. Access Control Lists (ACL) are lists of the permissions that are attached to an object or piece of data. The list contains information about all users who have read and write permissions for the file. Use the list to view which users have access to the file as well as which actions each user can perform. The permissions for each user or group are not set through Symantec Data Loss Prevention. Administrators set the permissions for each file using other applications. Permissions are generally set at the time that the file is created. A separate `fileACL` entry is included for each permission that was granted on the file. `fileACL` entries are only in certain Network Discover, Endpoint Discover, and Endpoint Prevent incidents.

Table C-8 DiscoverWebServerScannerIncidentDetail extension fields

Extension field	Field type	Occurrences	Description
`name`	`string`	1	The name of the Web server on which the scanner detected the incident. This is the name as configured in the `VontuWebServerScanner.cfg` file. See the *Symantec Data Loss Prevention Administration Guide* for more information about configuring Web server scans.
`file`	`messageComponentType`	0-many	A `messageComponentType` entry that encapsulates the entire file component. This field is provided only when you choose to include violation data in the incident detail request.
`lastModifiedDate`	`datetime`	0-1	The date and time that the file was last changed.
`filePath`	`string`	1	The full path to the file.

Table C-9 DiscoverLivelinkScannerIncidentDetail extension fields

Extension field	Field type	Occurrences	Description
`fileName`	`string`	1	The name of the file that violated the policy.
`filePath`	`string`	1	The full path to the file.
`file`	`messageComponentType`	0-many	A `messageComponentType` entry that encapsulates the entire file component. This field is provided only when you choose to include violation data in the incident detail request.
`createDate`	`datetime`	0-1	The date and time when the file was created.
`lastModifiedDate`	`datetime`	0-1	The date and time that the file was last changed.
`createdBy`	`string`	0-1	The creator of the file.

Table C-9 DiscoverLivelinkScannerIncidentDetail extension fields *(continued)*

Extension field	Field type	Occurrences	Description
`lastModifiedBy`	`string`	0-1	The name of the user who last changed the file.
`fileOwner`	`string`	0-1	The owner of the file.

Table C-10 DiscoverDocumentumScannerIncidentDetail extension fields

Extension field	Field type	Occurrences	Description
`fileName`	`string`	1	The name of the file that violated the policy.
`file`	`messageComponentType`	0-many	A `messageComponentType` entry that encapsulates the entire file component.
`fileOwner`	`string`	0-1	The owner of the file.
`createDate`	`datetime`	0-1	The date and time when the file was created.
`lastModifiedDate`	`datetime`	0-1	The date and time that the file was last changed.
`lastModifiedBy`	`string`	0-1	The name of the user who last changed the file.
`filePath`	`string`	1	The full path to the file.

Table C-10 DiscoverDocumentumScannerIncidentDetail extension fields *(continued)*

Extension field	Field type	Occurrences	Description
`fileACL`	`ACLType`	0-many	The Access Control Lists for the file. Access Control Lists (ACL) are lists of the permissions that are attached to an object or piece of data. The list contains information about all users who have read and write permissions for the file. Use the list to view which users have access to the file as well as which actions each user can perform. The permissions for each user or group are not set through Symantec Data Loss Prevention. Administrators set the permissions for each file using other applications. Permissions are generally set at the time that the file is created. A separate `fileACL` entry is included for each permission that was granted on the file. `fileACL` entries are only in certain Network Discover, Endpoint Discover, and Endpoint Prevent incidents.

Table C-11 DiscoverWebServiceIncidentDetail extension fields

Extension field	Field type	Occurrences	Description
`name`	`string`	1	The name of the file that was sent to the Web Service.
`messageHeader`	`MessageComponentType`	0-1	The subject line of the message. This field is provided only when you choose to include violation data in the incident detail request.
`messageBody`	`MessageComponentType`	0-1	The body text of the message. This field is provided only when you choose to include violation data in the incident detail request.

Table C-11 DiscoverWebServiceIncidentDetail extension fields *(continued)*

Extension field	Field type	Occurrences	Description
`component`	`MessageComponentType`	0-many	This component represents the entire file that was sent to the Web Service. This field is provided only when you choose to include violation data in the incident detail request.
`createDate`	`datetime`	0-1	The date and time when the message was created.
`lastModifiedDate`	`datetime`	0-1	The date and time that the message was last changed.
`createdBy`	`string`	0-1	The creator of the message.
`lastModifiedBy`	`string`	0-1	The name of the user who last changed the file.
`fileOwner`	`string`	0-1	The owner of the file at the time the incident was created.
`fileLastAccessDate`	`datetime`	0-1	The date and time of the last user access to the file.

Table C-11 DiscoverWebServiceIncidentDetail extension fields *(continued)*

Extension field	Field type	Occurrences	Description
`fileACL`	`ACLType`	0-many	The Access Control Lists for the file. Access Control Lists (ACL) are lists of the permissions that are attached to an object or piece of data. The list contains information about all users who have read and write permissions for the file. Use the list to view which users have access to the file as well as which actions each user can perform. The permissions for each user or group are not set through Symantec Data Loss Prevention. Administrators set the permissions for each file using other applications. Permissions are generally set at the time that the file is created. A separate `fileACL` entry is included for each permission that was granted on the file. `fileACL` entries are only in certain Network Discover, Endpoint Discover, and Endpoint Prevent incidents.

Table C-12 DiscoverSharePointCrawlerIncidentDetail extension fields

Extension field	Field type	Occurrences	Description
`filePath`	string	1	The full path to the item.
`documentName`	string	1	The name of the item that violated the policy.
`fileCreateDate`	datetime	0-1	The date and time when the item was created.
`fileLastModifiedDate`	datetime	0-1	The date and time when the item was last changed.
`createdBy`	string	0-1	The creator of the item.
`lastModifiedBy`	string	0-1	The name of the user who last changed the item.

Table C-12 DiscoverSharePointCrawlerIncidentDetail extension fields *(continued)*

Extension field	Field type	Occurrences	Description
`messageBodyContent`	`string`	0-1	The body text content of the item in string format. This field is provided only when you choose to include violation data in the incident detail request.
`messageBody`	`MessageComponentType`	0-1	The body text of the item. This field is provided only when you choose to include violation data in the incident detail request.
`messageHeader`	`MessageComponentType`	0-1	The subject line of the item. This field is provided only when you choose to include violation data in the incident detail request.
`file`	`messageComponentType`	0-many	A `messageComponentType` entry that encapsulates the entire SharePoint item. This field is provided only when you choose to include violation data in the incident detail request.

Table C-12 DiscoverSharePointCrawlerIncidentDetail extension fields *(continued)*

Extension field	Field type	Occurrences	Description
`fileACL`	`ACLType`	0-many	The Access Control Lists for the SharePoint item. Access Control Lists (ACL) are lists of the permissions that are attached to an object or piece of data. The list contains information about all users who have read and write permissions for the SharePoint item. Use the list to view which users have access to the item as well as which actions each user can perform. The permissions for each user or group are not set through Symantec Data Loss Prevention. Administrators set the permissions for each item using SharePoint. Permissions are generally set at the time that the item is created. A separate `fileACL` entry is included for each permission that was granted on the item. `fileACL` entries are only in certain Network Discover, Endpoint Discover, and Endpoint Prevent incidents.

Table C-13 DiscoverExchangeCrawlerIncidentDetail extension fields

Extension field	Field type	Occurrences	Description
`subject`	string	0 - 1	The subject line of the item in string format.
`originator`	NetworkOriginatorType	0 - 1	Details about the sender of the item, including the sender's IP address and port number, as well as an identifying string.
`recipient`	NetworkRecipientType	0 - many	Details about the intended recipient of the item, including the recipient's IP address and port number, as well as an identifying string.
`filePath`	string	1	The full path to the item.

Table C-13 DiscoverExchangeCrawlerIncidentDetail extension fields *(continued)*

Extension field	Field type	Occurrences	Description
`documentName`	string	1	The name of the Exchange item (EML file) that violated the policy.
`fileCreateDate`	datetime	0-1	The date and time when the item was created.
`fileLastModifiedDate`	datetime	0-1	The date and time when the item was last changed.
`createdBy`	string	0-1	The creator of the item.
`lastModifiedBy`	string	0-1	The name of the user who last changed the item. This field may be null if the Exchange server or connector does not provide a value.
`messageBodyContent`	`string`	0-1	The body text content of the item in string format. This field is provided only when you choose to include violation data in the incident detail request.
`messageBody`	`MessageComponentType`	0-1	The body text of the item. This field is provided only when you choose to include violation data in the incident detail request.
`messageHeader`	`MessageComponentType`	0-1	For email messages, this corresponds to the email subject. For other Exchange items, this field contains a derived header with metadata about the item. For example, a contact or appointment would provide an address or a business phone as part of the `messageHeader`. This field is provided only when you choose to include violation data in the incident detail request.
`file`	`messageComponentType`	0-many	A `messageComponentType` entry that encapsulates the entire Exchange item. This field is provided only when you choose to include violation data in the incident detail request.

Table C-14 DiscoverBoxCrawlerIncidentDetail extension fields

Extension field	Field type	Occurrences	Description
`fileID`	string	1	Unique identifier for the file.
`fileName`	string	1	The name of the file that violated the policy.
`sharedLinkInfo`	Complex type, see description.	1	`sharedLinkInfo` is a complex field type that includes the following information: ■ `hasLink`: A Boolean value indicating if the file has a shared link. ■ `details`: ■ `passwordProtected`: A Boolean value indicating if the shared link requires a password. ■ `downloadAllowed`: A Boolean value indicating if the shared link allows users to download the file. ■ `expirationDate`: A Boolean value indicating if the shared link has an expiration date.
`filePath`	string	1	The full path to the item.
`fileCreatedBy`	string	0-1	The creator of the file.
`fileModifiedBy`	string	0-1	The name of the user who last changed the item.
`fileLastAccessDate`	datetime	0-1	The date and time of the last user access to the file.
`fileLastModifiedDate`	datetime	0-1	The date and time when the item was last changed.
`fileCreateDate`	datetime	0-1	The date and time when the item was created.
`fileOwner`	string	0-1	The owner of the file at the time the incident was created.

Table C-14 DiscoverBoxCrawlerIncidentDetail extension fields *(continued)*

Extension field	Field type	Occurrences	Description
`fileCollaborator`	Complex type, see description.	0-many	`fileCollaborator` is a complex field type that includes the following information: ■ `collaborator`: A string, the name of any collaborator on the file. ■ `role`: A string, the role of any collaborator on the file.
`file`	`messageComponentType`	0-many	A `messageComponentType` entry that encapsulates the entire Exchange item. This field is provided only when you choose to include violation data in the incident detail request.

Endpoint component detail types

Each Endpoint extension type corresponds to the Endpoint detection mechanism that logged the original incident. The Incident Reporting and Update API schema defines the following component detail types as extensions to `EndpointIncidentDetailType`:

- `EndpointLocalFileSystemIncidentDetail` (Table C-15)
- `EndpointPrintFaxIncidentDetail` (Table C-16)
- `EndpointClipboardIncidentDetail` (Table C-17)
- `EndpointNetworkIncidentDetailType` (Table C-18)

The tables that follow describe the fields that each type adds to `EndpointIncidentDetailType`.

Several additional detail types are implemented as further extensions to `EndpointNetworkIncidentDetailType`:

- `EndpointEmailIncidentDetail` (Table C-19)
- `EndpointHTTPIncidentDetail` (Table C-20)
- `EndpointIMIncidentDetail`
- `EndpointFTPIncidentDetail`
- `EndpointNNTPIncidentDetail`

`EndpointIMIncidentDetail` and `EndpointFTPIncidentDetail` add no additional fields to `EndpointNetworkIncidentDetailType`. They are provided as placeholder types for future extension fields.

`EndpointNNTPIncidentDetail` is not implemented in this release.

Table C-15 EndpointLocalFileSystemIncidentDetail extension fields

Extension field	Field type	Occurrences	Description
`fileName`	`string`	0-1	The name of the file that violated the policy.
`filePath`	`string`	0-1	The path to the file.
`volumeName`	`string`	0-1	The volume on which the file is stored.
`DOSVolumeName`	`string`	0-1	The drive letter on which the file is stored.
`fileLastAccessDate`	`datetime`	0-1	The timestamp of the last file access. This information is applicable only to Endpoint Discover and Endpoint Prevent local drive monitoring.
`fileCreateDate`	`datetime`	0-1	The timestamp when the file was created. This information is applicable only to Endpoint Discover and Endpoint Prevent local drive monitoring.
`messageHeader`	`MessageComponentType`	0-1	The subject line of the message. This field is provided only when you choose to include violation data in the incident detail request.
`file`	`MessageComponentType`	0-many	The complete file component (the original file that violated the policy). This field is provided only when you choose to include violation data in the incident detail request.
`sourceFileName`	`string`	0-1	The file name of the original file when it is opened and copied to a new destination.
`sourceFilePath`	`string`	0-1	The file path of the original file when it is opened and copied to a new destination.

Table C-16 EndpointPrintFaxIncidentDetail extension fields

Extension field	Field type	Occurrences	Description
`printerName`	`string`	0-1	The printer name is available only if the file cannot be named through from the `printJobTitle`, or if the file was generated from an Internet browser.
`printJobTitle`	`string`	0-1	The file name of the printing job that generated the incident.
`content`	`MessageComponentType`	0-1	The partial text of the file that violated the policy. This field is provided only when you choose to include violation data in the incident detail request.
`messageHeader`	`MessageComponentType`	0-1	The subject line of the message. This field is provided only when you choose to include violation data in the incident detail request.

Table C-17 EndpointClipboardIncidentDetail extension fields

Extension field	Field type	Occurrences	Description
`applicationWindowTitle`	`string`	1	The title bar of the window from which the data was copied.
`content`	`MessageComponentType`	0-1	The partial text of the file that violated the policy. This field is provided only when you choose to include violation data in the incident detail request.
`messageHeader`	`MessageComponentType`	0-1	The subject line of the message. This field is provided only when you choose to include violation data in the incident detail request.

Table C-18 EndpointNetworkIncidentDetailType extension fields

Extension field	Field type	Occurrences	Description
`originator`	`NetworkOriginatorType`	0-1	The IP address or port of the endpoint computer that originated the incident. This is available only if the incident was created on the endpoint computer.
`recipient`	`NetworkRecipientType`	0-many	The destination endpoint computer associated with the incident. This is available only if the incident was created on the endpoint computer.
`messageHeader`	`MessageComponentType`	0-1	The subject line of the email/SMTP message. This field is provided only when you choose to include violation data in the incident detail request.
`messageBody`	`MessageComponentType`	0-1	The body text of the email/SMTP message. This field is provided only when you choose to include violation data in the incident detail request.
`file`	`MessageComponentType`	0-many	The complete file component (the original file that violated the policy). This field is provided only when you choose to include violation data in the incident detail request.

Table C-19 EndpointEmailIncidentDetail extension fields (extends EndpointNetworkIncidentDetailType)

Extension field	Field type	Occurrences	Description
`subject`	`string`	0-1	The subject of the email message. This field is provided only when you choose to include violation data in the incident detail request.

Table C-20 EndpointHTTPIncidentDetail extension fields (extends EndpointNetworkIncidentDetailType)

Extension field	Field type	Occurrences	Description
isHTTPS	Boolean	1	Indicates whether the Web request was transmitted over a secure connection. This field is provided only when you choose to include violation data in the incident detail request.

REST component detail types

The Incident Reporting and Update API scheme defines the following component detail types as extensions to`RestIncidentDetailType`:

- RestDIMIncidentDetail
- RestDARIncidentDetail

Note: All attributes are for both DIM and DAR, unless otherwise noted in the Description column.

Table C-21 REST component detail types

Extension field	REST attribute	Occurrences	Description
siteClassification	http.siteClassification	0-1	A site classification such as "social media." String type.
serviceClassification	common.service.classification	0-1	A service classification tag such as "sanctioned" from Symantec CloudSOC. String type.
transactionId	common.transactionId	0-1	A transaction ID that is used to link back to an incident on another console. String type.
requestId	common.requestId	1	A request ID. String type.

Table C-21 REST component detail types *(continued)*

Extension field	REST attribute	Occurrences	Description
`messageSource`	`common.messageSource`	0-1	The source of the message used when application names overlap. For example: Gatelet, Securlet. String type.
`sharepoint`	`common.sharepoint`	0-1	The SharePoint site name. String type.
`compliant`	`device.isCompliant`	0-1	Whether the device is compliant based on mobile device management or Symantec CloudSOC information. Boolean type.
`unmanaged`	`device.isUnmanaged`	0-1	Whether the device is managed based on mobile device management or Symantec CloudSOC information. Boolean type.
`personal`	`device.isPersonal`	0-1	Whether the device is personally owned, based on mobile device management or Symantec CloudSOC information. Boolean type.
`serviceScore`	`common.service.score`	0-1	The score of the service used. For example, Box has a score of 80. From the Symantec CloudSOC ShadowIT report. Integer type.

Table C-21 REST component detail types *(continued)*

Extension field	REST attribute	Occurrences	Description
applicationReportName	common.application.reportName	0-1	The display name for an application (such as "Box") that is not specific to a Securlet or Gatelet. String type.
applicationName	common.application	0-1	The DIM or DAR Application name, such as securlet.box or gatelet.box that is specific to a Securlet or Gatelet. String type.
applicationInstanceId	common.application.instanceId	0-1	For multiple subscriptions. Not used in 14.6. String type.
recipient	common.authrecipient	0-1	The authenticated recipient. When the request is inbound to a user's device and the user authentication is known, it is placed in this field. If multiple recipients exist, each recipient is comma separated; for example, "Bob, Alice." String type.
authSender	common.authsender	0-1	The authenticated sender. The meaning varies by detection client. This may be a logged-on user on an endpoint, an authenticated HTTP proxy user, or an authenticated SMTP user.
fileCreated	common.created	0-1	DAR requests only: File creation time stamp string representing the date and time per ISO 8601, such as 2015-10-13T10:11:06.419Z

Table C-21 REST component detail types *(continued)*

Extension field	REST attribute	Occurrences	Description
`dataType`	`common.dataType`	1	The type of the data to process. Consists of one of three values: *DAR* (data-at-rest) or *DIM* (data-in-motion) or *Mobile* (exclusively for the Secure Proxy). This value must be provided with every detection request. Fixed dictionary type.
`description`	`common.description`	0-1	The file description field for DAR requests only. String type.
`activityCount`	`common.doc.activityCount`	0-1	The count of user actions on the documents. Long type.
`creatorId`	`common.doc.creatorId`	0-1	The unique ID of the document creator. String type.
`docExposed`	`common.doc.exposed`	0-1	Shows whether the document has sharing or access exposures. This field is set to True if the document is either publicly shared, shared with everyone inside the enterprise or anyone outside the enterprise. The field is NOT set to True if the document is only shared with specific internal users of the enterprise. Boolean type.
`docExposedAllInternal`	`common.doc.exposures.allInternal`	0-1	Shows whether the document is shared with everyone inside the company. Boolean type.

Table C-21 REST component detail types *(continued)*

Extension field	REST attribute	Occurrences	Description
docExposuresExternal Collaborators	common.doc.exposures.external .collaborators	0-many	The list of email addresses of people outside the enterprise that have access to the document. These are called external collaborators. List type.
docExposuresInternal Collaborators	common.doc.exposures.internal .collaborators	0-many	List of email addresses of people inside the enterprise who have access to the doc. List type.
docExposedPublic	common.doc.exposures.public	0-1	Shows whether the document is shared with everyone publicly. These are documents that have share and access permissions open to anyone on the internet. Boolean type.
documentId	common.doc.id	0-1	Unique identifier of the document on the SaaS application. String type.
docInternal	common.doc.isInternal	0-1	Shows whether the document is internal to the enterprise. A document that is created by a user who belongs to the enterprise is termed as an internal document. Note that creator of the document is considered. Boolean type.
parentFolderIdentifier	common.doc.parentFolderId	0-1	Unique ID of the parent folder. String type.

Table C-21 REST component detail types *(continued)*

Extension field	REST attribute	Occurrences	Description
documentType	common.doc.type	0-1	The type of document. Possible values are file, File, folder, Folder. Also, the type of the primary object to which the activity applies: for example, Account, Site, File, or Folder. String type.
fileFolder	common.folder	0-1	The path where the files or attachments reside. String type.
lastAccessed	common.lastAccessed	0-1	DAR requests only. Last accessed time stamp. The string representing date and time is per ISO 8601. For example: 2015-10-13T10:11:06.419Z
lastModified	common.lastModified	0-1	DAR requests only. Last modified time stamp. The string representing date and time is per ISO 8601. For example: 2015-10-13T10:11:06.419Z
logId	common.log.id	0-1	The unique identifier for the log (use for deep linking). String type.
owner	common.owner	0-1	DAR requests only. The user ID of the owner of the file or attachment. String type.
sharedWith	common.sharedWith	0-many	DAR requests only. The array of user IDs the document is shared with. List type.

Table C-21 REST component detail types *(continued)*

Extension field	REST attribute	Occurrences	Description
`sharingUrl`	`common.sharingUrl`	0-1	DAR requests only. The URL used to share the document. String type.
`fileTag`	`common.tag`	0-1	DAR requests only. The file tag field. String type.
`threatScore`	`common.user.threatScore`	0-1	The threat score associated with the user or event. Long type.
`userInternal`	`common.user.isInternal`	0-1	Indicates whether the user is internal or external to the enterprise. Boolean type.
`docExposedCount`	`common.user.docsExposedCount`	0-1	The number of documents exposed to a user. Long type.
`groupMembership`	`common.user.groupMembership`	0-many	The list of groups that the user belongs on the SaaS application. Long type.
`userId`	`common.user.id`	0-1	The unique ID for the user. String type.
`userName`	`common.user.name`	0-1	The display name. String type.
`userActivityType`	`common.user.activityType`	0-1	The type of action that is performed on the primary object; for example: "Allow" or "Add" or “Create” or “Delete”. String type.

Table C-21 REST component detail types *(continued)*

Extension field	REST attribute	Occurrences	Description
restCustom	custom	0-1	A custom context attribute. Custom context attributes differ from the other well known context attributes in their treatment by the policy engine.
devicesIsTrusted	device.isTrustedDevice	0-1	The Boolean value (true or false) indicating if the request came from a trusted device. Boolean type.
deviceOS	device.os	0-1	The operating system of the device as a string value. String type.
deviceType	device.type	0-1	The type of the device as string value. String type.
httpBrowser	http.browser	0-1	The name of the browser, as derived from user agent. String type.
httpCookies	http.cookies	0-1	The cookies from HTTP requests. String type.
httpMethod	http.method	0-1	The method for detection requests related to HTTP traffic. String type.
httpSiteRiskScore	http.siteRiskScore	0-1	The risk level of the target site, as a numeric value. Long type.

Table C-21 REST component detail types *(continued)*

Extension field	REST attribute	Occurrences	Description
`httpUrl`	`http.url`	0-1	The URL for detection requests related to HTTP traffic. String type.
`httpUserAgent`	`http.userAgent`	0-1	The user agent for detection requests related to HTTP traffic. String type.
`linkDocExposure`	`link.doc.exposure`	0-1	The link to the Symantec CloudSOC console exposures panel for that document. String type.
`linkIncident`	`link.incident`	0-1	The link to the incident in the Symantec CloudSOC console. String type.
`linkServiceApplication`	`link.service.application`	0-1	The link to the Symantec CloudSOC Service Visibility UI with the panel for that user. String type.
`linkUba`	`link.uba`	0-1	Launches the Symantec CloudSOC console to the Investigate UI with a list of the last 7 days of activity for that user. Filtered by the application and the username. String type.

Table C-21 REST component detail types *(continued)*

Extension field	REST attribute	Occurrences	Description
linkDocExposure	link.user.exposures	0-1	The link to the Symantec CloudSOC console Securlets UI with a list of the exposures panel for that user. String type.
locationCoordsLatitude	location.coords.latitude	0-1	The geographic location of the device in latitude. String type.
locationCoordsLongitude	location.coords.longitude	0-1	The geographic location of the device in longitude.
insideOfficeLocation	location.isInsideOffice	0-1	The Boolean value (true or false) indicating if the request came from inside the office.
locationRegion	location.region	0-1	The recorded location of where the activity was performed. Format: City (Country). String type.
regionCountry	location.region.country	0-1	The recorded country where the activity was performed. String type.
networkDirection	network.direction	0-1	Indicates if this DIM call is for content upload or download. The acceptable values are download and upload. Fixed dictionary type.
networkProtocol	network.protocol	0-1	The OSI L7 network protocol applicable to the detection request. For example: "SMTP," "HTTP," or "FTP." String type.

Table C-21 REST component detail types *(continued)*

Extension field	REST attribute	Occurrences	Description
networkRecipientIp	network.recipient.ip	0-1	The network recipient IP applicable to the detection request. String type.
networkRecipientPort	network.recipient.port	0-1	The network recipient port applicable to the detection request. Long type.
networkSenderIp	network.sender.ip	0-1	The network sender IP address applicable to the detection request. String type.
networkSenderPort	network.sender.port	0-1	The network sender port applicable to the detection request. Long type.

Index

E

F

G

H

I

J

L

M

N

O

P

R

S

T

U

V

W

X

Y