Symantec™ Data Loss Prevention Reporting API Developers Guide

Version 11.0

Symantec

Symantec Data Loss Prevention Reporting API Developers Guide

Documentation version: 11.0

Legal Notice

Symantec Corporation
350 Ellis Street
Mountain View, CA 94043

http://www.symantec.com

Technical Support

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

Symantec's support offerings include the following:

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

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

www.symantec.com/business/support/

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

Contacting Technical Support

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

www.symantec.com/business/support/

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

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

- Product release level

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

Licensing and registration

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

www.symantec.com/business/support/

Customer service

Customer service information is available at the following URL:

www.symantec.com/business/support/

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

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

Support agreement resources

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

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

Additional enterprise services

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

Enterprise services that are available include the following:

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

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

www.symantec.com/business/services/

Select your country or language from the site index.

Contents

Chapter 1

Introducing the Symantec Data Loss Prevention Reporting API

This chapter includes the following topics:

About the Reporting API

The Symantec Data Loss Prevention Reporting API enables a skilled Web Services developer to create applications that pull incident data from the Enforce Server administration console. The incident data can be integrated with other applications or systems to provide dynamic reporting or to support business processes that rely on Data Loss Prevention incidents.

A Symantec Data Loss Prevention incident records all of the details 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 print operation, an HTTP request, or any other protocol message that you have configured Symantec Data Loss Prevention to monitor. Data 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 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.

The Reporting API provides an interface to access the incident data for reporting or for integrating the data with other applications. The API includes XML schemas to help you work with the different incident types that Symantec Data Loss Prevention supports.

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. The combined information can provide valuable information to security experts who are tasked with analyzing the data or with remediating security incidents.

Note: The Reporting API only enables client applications to retrieve incident data from the Enforce Server. Updating incident data in the Enforce database is not supported. For example, you cannot use the Reporting API to change the severity of an incident or execute a response action.

The Symantec Data Loss Prevention Reporting 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. Symantec recommends using Metro Web Services 1.5 or Microsoft .NET 3.5. The Reporting API Web Service is implemented using the Metro Web Services 1.5 stack, and runs in the same Tomcat container as the Enforce Server administration console under Java version 1.6.

Components of the Reporting API SDK

The Symantec Data Loss Prevention Reporting API SDK includes the following components.

Table 1-1 Reporting API SDK components

Component	Description
Reporting API Web Service Definition Language document (WSDL)	The WSDL document fully defines each of the request, response, and fault types provided by the Reporting API Web Service. You can obtain the WSDL document directly from an installed Enforce Server from the URL: https://*enforce_server*/ProtectManager/services/reporting?wsdl Replace *enforce_server* with the IP address or hostname of the Enforce Server. The WSDL is also available in the `c:\Vontu\Protect\tomcat\webapps\ProtectManager\WEB-INF\lib\reportingapi-schema.jar` file. You use the WSDL document to generate code when developing Reporting API clients. The WSDL also provides the Web Service bindings to client applications at runtime. See "Generating Web Service client proxy code" on page 18.
XML Schema Definitions (XSD files)	The Reporting API XSD files describe the contents and structure of the XML request and response documents 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 Reporting API WSDL. XSD files are also available in the `c:\Vontu\Protect\tomcat\webapps\ProtectManager\WEB-INF\lib\reportingapi-schema.jar` file. See "About incident detail types" on page 23.

Requirements for using the Reporting API

To use the Reporting 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 1.6 and .NET 3.5 implementations.

Symantec recommends that you use the Metro Web Service 1.5 framework or Microsoft .NET 3.5 platform to automatically generate code from the supplied WSDL document. Because the Web Service itself was developed using the Metro Web Services 1.5 stack, you may choose to use the same stack to speed client development.

See "Generating Web Service client proxy code" on page 18.

The Reporting 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

Reporting 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 Reporting API security

The Reporting 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 Reporting 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 the user name and password using the Enforce console before accessing the Web Service.

An authenticated user is authorized to access the Reporting API Web Service if the user is assigned a role that contains the **Reporting API** privilege.

See "Creating a user and role for a Reporting API client" on page 15.

Chapter 2

Implementing a Reporting API client

This chapter includes the following topics:

About Reporting API client implementations

The Reporting API provides a “read-only” Web Service interface that clients can use to retrieve Symantec Data Loss Prevention incident data. Before a client can interact with the Reporting API Web Service, these features must be available in the Enforce Server:

- A user account that has permission to access the Reporting API Web Service.
- 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 Reporting API user account that accesses the Web Service.

A Web Service client provides the credentials of the Enforce Server user account in each request that it sends to the Reporting 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.

A Web Service client typically begins by requesting a list of incidents specified by a saved report. The client uses the ID of the saved report to request a list of incident IDs from the Web Service. Although the saved report may return a very large list of incidents, the Web Service client can request a subset of incidents by specifying an incident creation date to constrain the query. 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 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 Reporting API Web Service operations, as well as error messages, are defined in the WSDL document.

See "Implementing a Reporting API client" on page 14.

The Reporting 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 23.

Implementing a Reporting API client

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

Table 2-1 Implementing a Reporting API client

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

Table 2-1 Implementing a Reporting API client *(continued)*

Step	Action	Description
Step 3	Create a saved report.	See "Creating a saved report for a Reporting API client" on page 17.
Step 4	Generate client code from the Reporting API Web Service WSDL.	See "Generating Web Service client proxy code" on page 18.
Step 5	Implement calls to the Reporting API Web Service.	See "About Reporting API Web Service operations" on page 22.
Step 6	Troubleshoot the client implementation.	See "Troubleshooting Reporting API client applications" on page 28.

Installing a development system

A single-tier Symantec Data Loss Prevention installation, having the Enforce and Detection server components deployed on a single server machine, is sufficient for development purposes. See the *Symantec Data Loss Prevention Installation Guide* for more information.

To verify client functionality, you must populate the development system with example incident data. Use automated scripts to generate example incident data, reports, and users.

Note: Although the Reporting API provides only a read-only Web Service, 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.

Creating a user and role for a Reporting API client

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

To create a Reporting API Web Service role and user

1. Log on to the Enforce Server administration console as an Administrator.
2. Select **System > User Management > Roles**.
3. Click **Add Role**.

4 Enter a name for the new role in the **Name** field. For example, enter "Reporting 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 each of the incident types that the Web Service role can view. If you do not select a particular incident type, the Web Service does not return incident details of that type to clients in this role.
Incidents: Actions	Select **Reporting API** to enable Reporting API Web Service access for the role.
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.
Incidents: Custom Attributes	Select **View** for each of the custom attributes that you want to include in incident detail responses for this role. Or, select **View All** to authorize access to all available 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. See the *Symantec Data Loss Prevention Lookup Plug-In Guide* 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 Reporting 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 a Reporting API client" on page 17.

6 Click **Save**.

7 Select **System > User Management > Users**.

8 Click **Add User**.

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

10 In the **Roles** section of the screen, select the new role you created in step 4. For example, select "Reporting API Client Role."

11 Click **Save**.

Creating a saved report for a Reporting API client

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

See "*Reporting API Web Service call reference*" on page 31.

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 Reporting 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 Reporting API.

To create a saved report for a Reporting API Web Service client

1 Log on to the Enforce Server administration console as the Reporting API Web Service user.

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

See "Creating a user and role for a Reporting API client" on page 15.

---

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 Reporting 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 returned from the Reporting API Web Service.

See "Creating a user and role for a Reporting API client" on page 15.

---

8 Select **Report > Save As**.

9 Enter 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: 'Reporting 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 mouse pointer over the saved report name.

Generating Web Service client proxy code

Symantec recommends that you use a Web Service development framework when building a Reporting 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 Reporting 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 1.5 or another environment that generates Java API for XML Web Services (JAX-WS) style code artifacts, such as the Java EE SDK.

Microsoft .NET developers should use the .NET 3.5 platform, which supports the Windows Communication Foundation (WCF) API for Web Services development. The Reporting API Web Service was developed using Metro Web Services, which provides full interoperability with WCF clients.

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. This document uses the Sun `wsimport` command-line utility in example commands. See your framework documentation for details about generating skeleton code from a WSDL file.

Consuming the Reporting API WSDL over SSL

The Reporting API WSDL document is available from the Enforce Server administration console at https://*enforce_server*/ProtectManager/services/reporting?wsdl. 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.

Default installations of Symantec Data Loss Prevention use a self-signed certificate that is embedded in the software. There is no default keystore file installed. This leaves two options for accessing the certificate from a development framework:

- Use a browser to export the default, self-signed certificate to a file. Then import the certificate to the keystore used by the development framework.
- Generate a dedicated keystore file (and a new self-signed certificate) for the Enforce Server. Then configure the development framework to access the same keystore file for trusted certificates.

The second approach can be used only in development environments where the Enforce Server runs on the local machine. The following procedure describes the second approach.

Generating a shared keystore file for Web Service development

1. Move to the *installdir*/Vontu/jre/bin directory. For example:

   ```
   cd /opt/Vontu/jre/bin
   ```

2. Use the Java `keystore` utility to generate a keystore file and self-signed certificate. For example:

   ```
   ./keytool -genkey -alias tomcat -keyalg RSA -keysize 1024
   -keystore .keystore -validity Expiration -storepass protect
   -dname "cn=common_name,o=organization_name,
   ou=organization_unit,l=city,s=state,c=US"
   ```

3. Move the new keystore file to the Symantec Data Loss Prevention `keystore` directory. For example:

   ```
   mv .keystore /opt/Vontu/Protect/keystore
   ```

4. Restart the Enforce Server to use the new keystore.
5. Configure your development framework to use the Enforce Server keystore file. For example, Java utilities such as `wsimport` can specify the keystore location using command line options, as in:

   ```
   -Djavax.net.ssl.keyStore=/opt/Vontu/Protect/keystore/.keystore
   -Djavax.net.ssl.keyStorePassword=protect
   -Djavax.net.ssl.trustStore=/opt/Vontu/Protect/keystore/.keystore
   -Djavax.net.ssl.trustStorePassword=protect
   ```

   See your development framework documentation for more information.

Authenticating a client with the Reporting API Web Service

The Reporting 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 Reporting API Web Service. The general process is as follows:

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 Enforce 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.

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 Reporting API service.
//
URL serviceUrl = new URL("https://localhost/ProtectManager/services/reporting");
QName serviceNamespace = new QName("http://www.vontu.com/enforce/webservice/incident",
   "ReportingService"));
ReportingService service = new ReportingService(serviceUrl, serviceNamespace);

// Bind credentials to the service port.
//
ReportingServicePortType servicePort = service.getReportingServicePort();
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 specify HTTP Basic authentication directly in the Web Service port binding, and then add the user credentials. 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";

// Create service address for Reporting API.
//
string url = "https://localhost/ProtectManager/services/reporting"
EndpointAddress epAddress = new EndpointAddress(url);

// Create an HTTP binding for Basic authentication.
//
BasicHttpBinding httpBinding = new BasicHttpBinding();
httpBinding.Security.Mode = BasicHttpSecurityMode.Transport;
httpBinding.Security.Transport.ClientCredentialType = HttpClientCredentialType.Basic;

// Add credentials and use HTTP Basic authentication.
//
ReportingServicePortTypeClient client = new ReportingServicePortTypeClient(httpBinding,
   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 Web Service.
//
client.Open();
```

About Reporting API Web Service operations

The Reporting API WSDL supports three operations for retrieving incident data:

- incidentList() returns a list of incident IDs described by a saved report. The client specifies the saved report ID to retrieve and the date of the oldest incident ID to return from the report. 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 one of the remaining two Web Service calls to retrieve additional details about one or all of the returned incident IDs.
- incidentDetail() returns the basic 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 "About incident detail types" on page 23.
- incidentBinaries() 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.

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. Failed operations return one of several possible faults.

See "Troubleshooting Reporting API client applications" on page 28.

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.

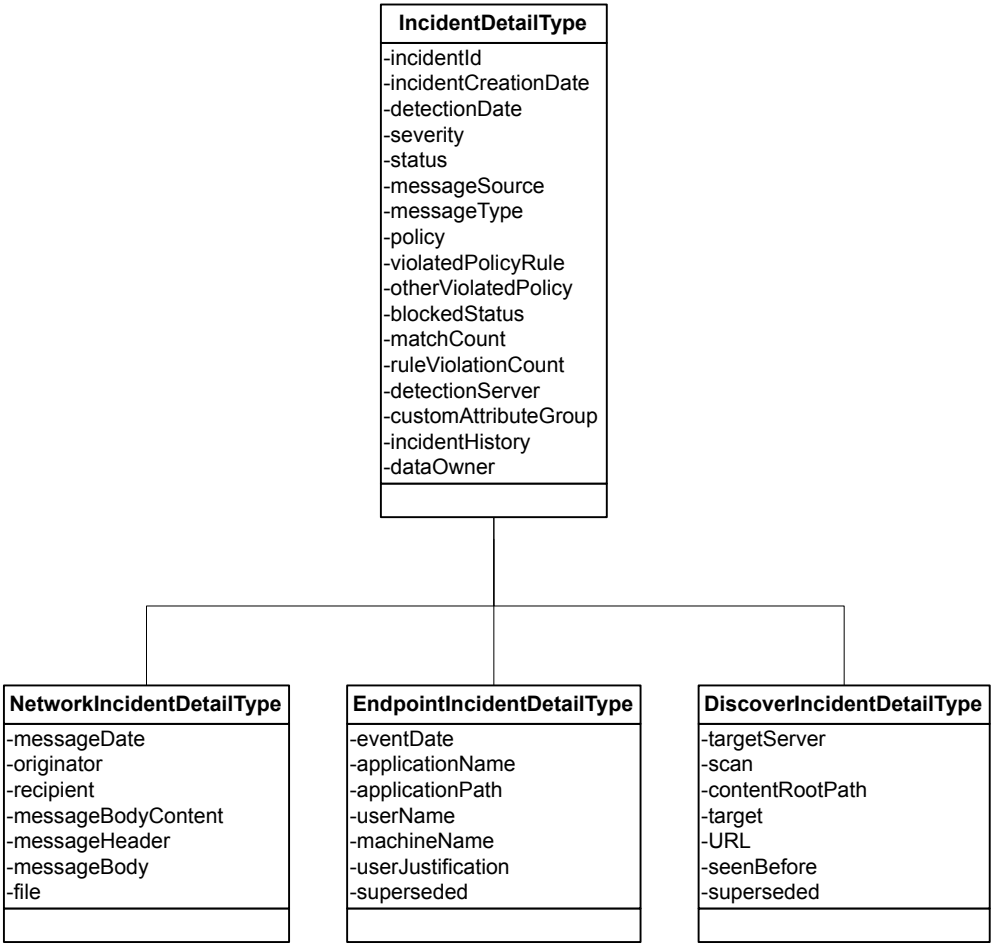
Incidents that are created by different Symantec Data Loss Prevention products contain additional information that is unique to the product group. For example, Symantec Data Loss Prevention network products (Network Monitor and Network Prevent) may contain a message header or recipient field. Network 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 Reporting API XML schema defines product-specific incident detail types by extending the base `IncidentDetailType` with additional fields. Three product incident detail types are defined:

Product incident detail type	Products that generate this incident type
`NetworkIncidentDetailType`	■ Network Monitor ■ Network Prevent (Email) ■ Network Prevent (Web)
`EndpointIncidentDetailType`	■ Endpoint Discover ■ Endpoint Prevent
`DiscoverIncidentDetailType`	■ Network Discover ■ Network Protect

Figure 2-1 shows the base fields defined in `IncidentDetailType` and the extension fields defined in each product incident detail type. 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



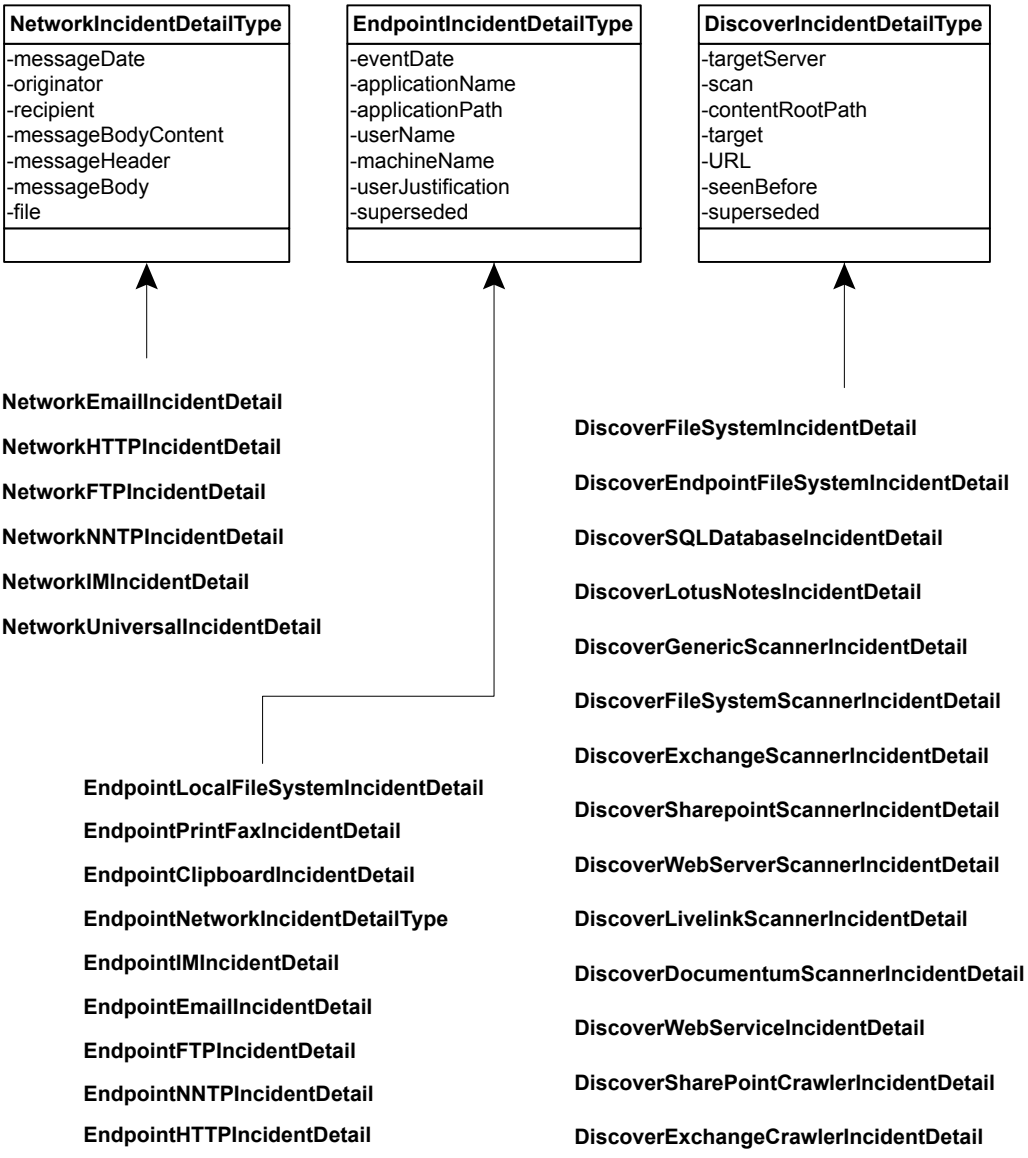
Within a particular Symantec Data Loss Prevention product, incident details may be further differentiated by the product component that detected the incident or by the protocol used to transmit the original message. For example, a user can generate an incident by attempting to copy sensitive data to 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 Reporting 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 2-2 shows the product component incident detail types, that extend the three product group incident types. (Note that the attributes for individual product and component types are omitted for space considerations.)

Figure 2-2 Hierarchy of component incident detail types



Troubleshooting Reporting API client applications

Each of the three available Reporting 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 provided by the Reporting 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 and failed attempts to access the Reporting 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 Reporting API Web Service on the Enforce Server.
  `webservice_access.log` also logs successful client requests with the:
  - 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 Reporting API Web Service. The log records all requests and responses except responses to incident binary requests. Although this log also records Reporting API faults, any Java exceptions generated by Enforce are logged to the Tomcat log file.
  `webservices_soap.log` is not created by default. To begin logging to this file, edit the `c:\Vontu\Protect\config\Manager.Logging.properties` file to set the `com.vontu.enforce.reportingapi.webservice.log.WebServiceSOAPLogHandler.level` property to `INFO`.

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 Reporting API Web Service:

- `manager_operational.log` is a Symantec Data Loss Prevention operational log file that records lifecycle and system events associated with the Enforce administration console. The Reporting API Web Service works with the Enforce administration console to provide incident data through SOAP requests. See "Operational Log Files and Codes" in the *Symantec Data Loss Prevention System 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 Reporting API faults, Web Service operational logs, and Symantec Data Loss Prevention operational logs. The Enforce Server stores the local Tomcat log on *installdir*`\Vontu\Protect\logs\tomcat\localhost.`*timestamp*`.log`.

Appendix A

Reporting API Web Service call reference

This appendix includes the following topics:

incidentList()

`incidentList()` – Returns a sequence of incident IDs by executing a saved report on the Enforce Server.

Syntax

```
IncidentListResponse = ReportingServicePortType.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 Reporting API Web Service. See "Creating a saved report for a Reporting API client" on page 17.

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

Name	Type	Description
`oldestIncidentCreationDate`	`dateTime`	Constrains the list of returned incident IDs by specifying the oldest incident creation date. The Web Service returns only those incidents that were created on or after the `oldestIncidentCreationDate`. A null value retrieves all incident IDs described by the saved report. As a best practice, client applications should record the time of the last `incidentList` call, and use that time to retrieve only the 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 a Reporting API client" on page 15. A null value retrieves all incident IDs described by the saved report.

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 oldest incident creation date and any role-based access controls applied to the Web Service user.

See "Creating a saved report for a Reporting API client" on page 17.

Examples

The following example shows how a Java client retrieves a list of incidents with `incidentList()`. Note that the `savedReportId` and `oldestIncidentCreationDate` variables are populated from a form.

```
// Create an IncidentListRequest.
//
IncidentListRequest request = new IncidentListRequest();

// Obtain and set the savedReportId.
//
String reportId = form.getText(form.reportId);
request.setSavedReportId(Integer.parseInt(reportId));

// Obtain, format, and set the oldestIncidentCreationDate.
//
Date incidentDate = form.oldestIncidentDateTime.getDate();

if(form.oldestIncidentDateTime.isEnabled() && incidentDate != null)
{
 GregorianCalendar calendar = new GregorianCalendar();
 calendar.setTime(incidentDate);
 DatatypeFactory dataFactory = DatatypeFactory.newInstance();
 XMLGregorianCalendar xmlCalendar = dataFactory.newXMLGregorianCalendar(calendar);

 request.setOldestIncidentCreationDate(xmlCalendar);
}

// Request a list of Incident IDs, adding them to an ArrayList.
//
IncidentListResponse response = servicePort.incidentList(request);
List<Integer> incidentIds = new ArrayList<Integer>();
incidentIds.addAll(response.getIncidentId());
```

Faults

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

- `authenticationFailedFault`
- `authorizationFailedFault`
- `serviceErrorFault`

If you specify an invalid saved report ID, the error is captured in a `serviceErrorFault`.

incidentDetail()

`incidentDetail()` – request details of a specified incident ID.

Syntax

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

Inputs

This call takes a single `IncidentDetailRequest` type as its argument. `IncidentDetailRequest` encapsulates the ID of the incident for which to provide details. The request may optionally indicate whether the Web Service should also return incident violation data and historical information.

The following table describes the `IncidentDetailRequest` fields.

Table A-2 IncidentDetailRequest fields

Name	Type	Description
`incidentId`	int	This required field identifies the unique ID of the incident whose details you want to retrieve.
`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 the incident detail reference pages for more information.

Table A-2 IncidentDetailRequest fields *(continued)*

Name	Type	Description
`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` that 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.

See "About incident detail types" on page 23.

See "About extended incident detail types" on page 57.

Examples

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

```
// Set variables for example.
//
int id = 2033;
IncidentDetailType incident;
IncidentDetailRequest detailRequest;
IncidentDetailResponse detailResponse;
boolean includeHistory = false;
boolean includeViolations = false;


// Create an IncidentDetailRequest.
//
detailRequest = new IncidentDetailRequest();

// Set request fields.
//
detailRequest.setIncidentId(id);
detailRequest.setIncludeHistory(includeHistory);
```

```
detailRequest.setIncludeViolations(includeViolations);

// Execute detail request and retrieve IncidentDetailType object.
//
detailResponse = servicePort.incidentDetail(detailRequest);
incident = detailResponse.getIncident();

// Access messageSource and messageType fields.
//
form.addIncidentDetailProperty("Message source",
   IncAccess.isAvailable(incident.getMessageSource())?incident.getMessageSource().getSourceTy
form.addIncidentDetailProperty("Message type",
   IncAccess.isAvailable(incident.getMessageType())?incident.getMessageType().getValue():null
```

Faults

The `incidentDetail` call can return only the following general faults:

- `serviceErrorFault`
- `authenticationFailedFault`
- `authorizationFailedFault`
- `incidentNotFoundFault`

incidentBinaries()

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

Syntax

```
IncidentBinariesResponse =
   ReportingServicePortType.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-3 IncidentBinariesRequest fields

Name	Type	Description
`incidentId`	int	This required field identifies the unique ID of the incident whose components you want to retrieve.
`includeOriginalMessage`	Boolean	This optional element indicates whether the Web Service should include the original message in the response document.
`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.
`componentId`	int	This optional element indicates a specific component ID to return in the response document.

Examples

The following example shows how a Java client retrieves a list of incidents with `incidentBinaries()`. Note that the example populates variables using a form.

```
// Connect to service, invoke 'incidentBinaries' method
// and disconnect
//
client = new ReportingServiceClient(url, user, passw);
client.connect();
IncidentBinariesResponse response = client.getServicePort().incidentBinaries(request);

// Incident original message
//
if (form.includeOriginalMessageCheckBox.isSelected())
  {
    DataHandler messageContent = response.getOriginalMessage();
      if (messageContent != null)
        {
          Component origMessage = new Component();
          origMessage.setComponentId(-1);
          origMessage.setName("ORIGINAL MESSAGE");
          origMessage.setContent(messageContent);
          origMessage.setComponentType("");
          origMessage.setComponentTypeId(-1);

          form.addIncidentBinariesComponent(origMessage);
        }
  }
form.setProgress(progress_step * 2);

// Incident components
//
List<Component> components = response.getComponent();
```

Faults

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

- `serviceErrorFault`
- `authenticationFailedFault`
- `authorizationFailedFault`
- `incidentNotFoundFault`

- `componentNotFoundFault`
- `invalidRequestFault`

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 a Reporting API client" on page 15.

Table B-1 IncidentDetailType fields

Field	Type	Occurrences	Description
`incidentId`	int	1	The unique ID number of the incident.
`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.
`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.

Table B-1 IncidentDetailType fields *(continued)*

Field	Type	Occurrences	Description
`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 (Email), or Network Prevent (Web) ■ DISCOVER—Network Discover or Network Protect ■ ENDPOINT—Endpoint Discover or Endpoint Prevent
`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 Table B-2 on page ?.

Table B-1 IncidentDetailType fields *(continued)*

Field	Type	Occurrences	Description
`policy`	`PolicyType`	1	Describes the policy that 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. See "Detection Methods" in the *Symantec Data Loss Prevention System Administration Guide* for more information.
`ruleViolationCount`	int	0–1	Indicates the number of policy rules that were violated. This field is included only the 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 Lookup Plug-In Guide* for more information about custom attributes. See "Creating a user and role for a Reporting API client" on page 15.

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 System 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
DropFolder	12	DiscoverIncidentDetailType	n/a Symantec Data Loss Prevention no longer supports drop folder targets, and this `messageType` is not used.
Removable Storage	13	EndpointIncidentDetailType	EndpointLocalFileSystemIncidentDetail
Local Drive	14	EndpointIncidentDetailType	EndpointLocalFileSystemIncidentDetail
EndpointFileSystem	15	DiscoverIncidentDetailType	DiscoverFileSystemScannerIncidentDetail
SharePointScanner	16	DiscoverIncidentDetailType	DiscoverSharepointScannerIncidentDetail
ExchangeScanner	17	DiscoverIncidentDetailType	DiscoverExchangeScannerIncidentDetail
WebServerScanner	18	DiscoverIncidentDetailType	DiscoverWebServerScannerIncidentDetail
FileSystemScanner	19	DiscoverIncidentDetailType	DiscoverFileSystemScannerIncidentDetail
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

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

String value	typeId attribute value	Product category type	Product component type
IM:MSN	30	EndpointIncidentDetailType	EndpointIMIncidentDetail
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 Reporting API to access classification results.
Exchange crawler	38	DiscoverIncidentDetailType	DiscoverExchangeCrawlerIncidentDetail
n/a	n/a	n/a	EndpointNNTPIncidentDetail is not implemented in this release.
Custom protocol	1000 or greater	NetworkIncidentDetailType	NetworkUniversalIncidentDetailType

NetworkIncidentDetailType

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

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.

Table B-5 *(continued)*

Field	Type	Occurrences	Description
`superseded`	string	1	This field indicates whether the incident response was superseded by another response (Yes or No).

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 Reporting API schema. Component detail types extend the base product types—NetworkIncidentDetailType, DiscoverIncidentDetailType, and EndpointIncidentDetailType—with fields specific to the protocol or product component that generated the incident.

See "About incident detail types" on page 23.

Network component detail types

The Reporting 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 Reporting 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)
- DiscoverExchangeScannerIncidentDetail (Table C-8)
- DiscoverSharepointScannerIncidentDetail (Table C-9)
- DiscoverWebServerScannerIncidentDetail (Table C-10)
- DiscoverLivelinkScannerIncidentDetail (Table C-11)
- DiscoverDocumentumScannerIncidentDetail (Table C-12)
- DiscoverWebServiceIncidentDetail (Table C-13)
- DiscoverSharePointCrawlerIncidentDetail (Table C-14)
- DiscoverExchangeCrawlerIncidentDetail (Table C-15)

Each extension type corresponds to the Network 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.
remediationLocation	string	0-1	The location where the file was copied or quarantined.
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
`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.
`LOB`	`messageComponentType`	0-1	A `messageComponentType` that encapsulates a large object component retrieved from the database. A SQL Database scan can extract data from many Java Database Connectivity (JDBC) types such as CLOB, BLOB, BIGINT, CHAR, LONGVARCHAR, VARCHAR, TINYINT, SMALLINT, INTEGER, REAL,DOUBLE,FLOAT, DECIMAL,NUMERIC,DATE, TIME, and TIMESTAMP. This field is provided only when you choose to include violation data in the incident detail request. See the *Symantec Data Loss Prevention System Administration Guide* for more information about SQL Database scans and incidents.

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.

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

Extension field	Field type	Occurrences	Description
`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.
`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.

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.
`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.

Table C-6 DiscoverGenericScannerIncidentDetail extension fields *(continued)*

Extension field	Field type	Occurrences	Description
`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.

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.
`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.

Table C-7 DiscoverFileSystemScannerIncidentDetail 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-8 DiscoverExchangeScannerIncidentDetail extension fields

Extension field	Field type	Occurrences	Description
`messageBody`	`MessageComponentType`	0-1	The body text of the Email message. 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 Email message. This field is provided only when you choose to include violation data in the incident detail request.

Table C-8 DiscoverExchangeScannerIncidentDetail 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.
`owner`	string	0-1	The owner of the message.
`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.

Table C-9 DiscoverSharepointScannerIncidentDetail 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.
`owner`	string	0-1	The owner of the file at the time the incident was created.

Table C-9 DiscoverSharepointScannerIncidentDetail 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.
`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.
`lastModifiedBy`	string	0-1	The name of the user who last changed the file.

Table C-10 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 System 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.

Table C-11 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.
`lastModifiedBy`	string	0-1	The name of the user who last changed the file.

Table C-12 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.

Table C-13 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.
`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.

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

Extension field	Field type	Occurrences	Description
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.

Table C-14 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.
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.

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

Extension field	Field type	Occurrences	Description
`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.
`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-15 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.

Table C-15 DiscoverExchangeCrawlerIncidentDetail extension fields *(continued)*

Extension field	Field type	Occurrences	Description
`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.
`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.

Table C-15 DiscoverExchangeCrawlerIncidentDetail extension fields *(continued)*

Extension field	Field type	Occurrences	Description
`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.

Endpoint component detail types

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

- `EndpointLocalFileSystemIncidentDetail` (Table C-16)
- `EndpointPrintFaxIncidentDetail` (Table C-17)
- `EndpointClipboardIncidentDetail` (Table C-18)
- `EndpointNetworkIncidentDetailType` (Table C-19)

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-20)
- `EndpointHTTPIncidentDetail` (Table C-21)
- `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-16 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.

Table C-17 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-18 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-19 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-20 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-21 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.

Index

F

H

I

J

K

L

M

N

O

P

R

S

T

U

V

W

X

Y