CA Enterprise Log Manager

API Programming Guide

r12.5

This documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the “Documentation”) is for your informational purposes only and is subject to change or withdrawal by CA at any time.

This Documentation may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA. This Documentation is confidential and proprietary information of CA and may not be disclosed by you or used for any purpose other than as may be permitted in (i) a separate agreement between you and CA governing your use of the CA software to which the Documentation relates; or (ii) a separate confidentiality agreement between you and CA.

Notwithstanding the foregoing, if you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all CA copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION “AS IS” WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is CA.

Provided with “Restricted Rights.” Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

CA Technologies Product References

This document references the following CA Technologies products:

- CA Access Control
- CA Audit
- CA ACF2™
- CA Directory
- CA Embedded Entitlements Manager (CA EEM)
- CA Enterprise Log Manager
- CA Identity Manager
- CA IT Process Automation Manager (CA IT PAM)
- CA NSM
- CA Security Command Center (CA SCC)
- CA Service Desk
- CA SiteMinder®
- CA Spectrum®
- CA Top Secret®

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information you need for your Home Office, Small Business, and Enterprise CA Technologies products. At http://ca.com/support, you can access the following:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Provide Feedback

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

If you would like to provide feedback about CA Technologies product documentation, complete our short customer survey, which is available on the CA Support website at http://ca.com/docs.

Documentation Changes

The following documentation updates have been made since the last release of this documentation:

- getObject - This existing topic contains a description of the getIncidentModel command.
- getIncidentModel - This new topic contains an example of the getIncidentModel command.

Contents

Chapter 1: About This Guide

The *CA Enterprise Log Manager API Programming Guide* provides instructions for using the CA Enterprise Log Manager API to access data from the event repository using the query and report mechanism, and display it in a web browser. You can also use the API to embed CA Enterprise Log Manager queries or reports in a CA or third-party product interface.

The guide is designed for administrators or web designers familiar with basic API structure and usage, CA Enterprise Log Manager queries, federation, and event refinement. They need administrator access to CA Enterprise Log Manager and other required third-party or CA products.

Chapter 2: About the CA Enterprise Log Manager API

The CA Enterprise Log Manager API uses a web application that accepts HTTPs post commands to return the query or report information you want. The web application consists of a dedicated iGateway spindle.

You use specific URLs that include arguments to control what data is returned and how it is filtered. Each available URL / API command verifies whether the caller is authenticated by validating the session id or certificate credentials. Each HTTPs request must contain one of these types of authentication information.

CA Enterprise Log Manager API features include:

- Authenticated, secure APIs
- Product registration for single sign-on (SSO)
- Retrieval of a query or report list, filtered by tag
- Display of a query or report in the interactive CA Enterprise Log Manager interface, allowing filtering, and embedding in a user interface

To use CA Enterprise Log Manager API calls effectively, be familiar with the federation structure of your environment, the queries and reports available, and user roles and their access rights.

More information:

CA Enterprise Log Manager API Structure (see page 12)
API Authentication (see page 15)
CA Enterprise Log Manager API Examples (see page 21)

API Call Returns

All API commands, with the exception of getQueryViewer and getReportViewer, return an element in the XML that describes whether the command was successful and if it was not, the reason why.

API Return Example

```
<?xml version="1.0" encoding="UTF-8" ?>
<Result>
<Value>true</Value>
<Description>Get Object Successful. Type [getQueryList]</Description>
<Items>
<Item edit="false">
        <Panel id="Subscription/panels/Unclassified_Event_Detail" name="Unclassified Event Detail" shortname="Detail"
subscription="true" type="EventViewer" version="12.0.46.5">
        <Description>Provides event details for unclassified event activity</Description>
```

In this case, the result value is “true” indicating success, and the description contains the command executed.

CA Enterprise Log Manager API Structure

A CA Enterprise Log Manager API call uses the HTTPS protocol to contact the event log store. The call returns results in XML, or in the form of a graphical query or report display, depending on the call you use.

Each call has a defined URL structure, consisting of several common elements. For example, an API login call appears this way:

```
https://ELMSERVER:5250/spin/calmapi/calmapi_login.csp?username=xx&password=xx
```

The first element defines the target server:

```
https://ELMSERVER:5250/spin/calmapi/
```

To use the call in your environment, replace the "ELMSERVER" portion of the URL with the hostname or IP address of the server where the data you want is stored. Port 5250 is the default port used by CA Enterprise Log Manager. The "/spin/calmapi/" text remains the same for any call.

The second element defines the API call itself, and provides any authentication details:

```
calmapi_login.csp?username=xx&password=xx
```

"calmapi_login.csp" is the login call. The second portion, "?username=xx&password=xx" defines the credentials that are used to log in. In this case it is a CA Enterprise Log Manager user name and password.

More information:

API Authentication (see page 15)
API Registration (see page 50)

Chapter 3: API Authentication

Your API calls must be authenticated to access the CA Enterprise Log Manager event log store. Here are several ways to set up authentication:

- Using a valid CA Enterprise Log Manager user name and password as part of the authentication URL. When constructing a call, verify that the information you want is available to the user account you want to use to authenticate.
- Using a certificate name and certificate password as part of the authentication URL. You can create a certificate from the API product registration interface. See the *CA Enterprise Log Manager API Online Help* for more information about creating a certificate.
- Using a session ID as part of the authentication URL. This session ID is a unique ID returned as part of the XML response after a successful authentication call. Use one of the other authentication methods to derive a session ID, which you can then use to create another session.

User Name and Password Example

```
https://ELMSERVER:5250/spin/calmapi/getObject.csp?type=getQueryList&username=xx&password=xx
```

This example uses the getQueryList command, and authenticates using a CA Enterprise Log Manager user name and password.

Certificate Name and Password Example

```
https://ELMSERVER:5250/spin/calmapi/getObject.csp?type=getELMServers&certname=xx&password=xx
```

This example uses the getELMservers command, and authenticates using a certificate name and password.

Session ID Example

```
https://ELMSERVER:5250/spin/calmapi/getObject.csp?type=getQueryViewer&objectId=Subscription/panels/Syst
em_Event_Count_By_Event_Action&sessionID=xxxxx
```

This example uses the getQueryViewer command, and authenticates using a session ID.

More information:

API Login (see page 16)
Register a Product (see page 54)
API Certificate Creation (see page 50)

API Login

This call authenticates a user using a set of CA EEM credentials, a certificate, or a session ID.

Because you can include authentication information in any API call URL, you do not need a separate login call in most cases. The login call is most useful for returning a Session ID, which can then be used to authenticate another call, such as getReportViewer.

The arguments used for this call are as follows:

username

> Defines the valid CA Enterprise Log Manager user name for authentication.

certname

> Defines the certificate name for authentication, if you have registered the product you want to access CA Enterprise Log Manager.

password

> Defines either the CA Enterprise Log Manager user password, or the certificate password for authentication, depending on which method you have used for authentication.

sessionid

> Defines the session ID from an existing authenticated session, which you can use to authenticate a new session.

API Login Examples

Command:

```
https://ELMSERVER:5250/spin/calmapi/calmapi_login.csp&username=xx&password=xx
```

Success Response:

```
  <?xml version="1.0" encoding="UTF-8" ?>
 <Result>
  <Value>true</Value>
  <Description>Authentication Successful.</Description>
  <SessionId>spin=62e39751-computername.domain.com49b8a97e-9bfd318-1</SessionId>
  </Result>
```

The session ID opened by the login appears in the <SessionId> tag.

Failure Response:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Result>
<Value>false</Value>
                <Description> EE_AUTHFAILED Authentication Failed</Description>
</Result>
```

More information:

API Authentication (see page 15)
Query and Report Viewer Calls (see page 34)

API Logout

This call ends an API session by logging out a user, ends a certificate session, or ends a session created through session ID. The call accepts no arguments.

API Logout Examples

```
https://ELMSERVER:5250/spin/calmapi/calmapi_logout.csp
```

Success Response:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Result>
<Value>true</Value>
        <Description>Logout Successful</Description>
</Result>
```

Failure Response:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Result>
<Value>false</Value>
        <Description> User is not logged in</Description>
</Result>
UTHFAILED Authentication Failed</Description>
</Result>
```

About API Sessions

CA Enterprise Log Manager creates a session each time you use an API call. The persistence of these sessions differs depending on the authentication method you use:

- User name and password, or session ID-authenticated sessions expire in the same way CA Enterprise Log Manager sessions do, using the Session Timeout value, which is set at 15 minutes by default. You can set the Session Timeout value from the CA Enterprise Log Manager interface.
- Certificate-authenticated sessions do not expire except in certain circumstances. The Session Timeout value is suspended, which lets you integrate CA Enterprise Log Manager with a web portal or outside product more easily. However, it can require additional action to avoid unnecessary use of system resources by persistent sessions.

CA Enterprise Log Manager closes certificate-authenticated sessions under the following circumstances:

- Closing a browser displaying a graphic component such as a query
- Logging out of an outside product
- Allowing the user session of an outside product to expire,

The CA Enterprise Log Manager session timer begins to count down, and ends the session after the timeout value you have configured expires.

If many getQueryViewer or getReportViewer calls are in use, there can be a number of open but idle sessions. To reduce the system resources used by such sessions, use the logout command to end a session when an outside product user logs out, or an outside product session ends.

More information:

Query and Report Viewer Calls (see page 34)
API Authentication (see page 15)
API Certificate Creation (see page 50)
API Login (see page 16)
API Logout (see page 18)

Chapter 4: CA Enterprise Log Manager API Examples

This section contains the following topics:

About API Examples

This chapter contains examples of API calls. Each example describes the URL required, and gives the expected XML return for success or failure, if any. These calls can be tested by entering the URL directly in a browser, and observing the XML response.

The getQueryViewer and getReportViewer calls provide CA Enterprise Log Manager event and query interface displays rather than XML. They are considered in their own section of this guide.

GetObject

You can use this command file to retrieve various types information. You can use it to retrieve a list of queries, reports, or global parameters, and the Common Event Grammar (CEG). The getObject command uses a qualifier or argument named “type” to determine what data to return to the caller as in this example:

```
https://ELMSERVER:5250/spin/calmapi/getObject.csp?type=type&tag=tagname1&tag=tagnamen&taglogic=OR|
AND
```

The following list provides a summary of the types of data returned using the variations of this command:

getQueryList

Returns an XML string showing all the queries in CA Enterprise Log Manager. getQueryList supports many filtering parameters, which let you select and include proper query names in your API calls.

getReportList

Returns an XML string showing all the reports in CA Enterprise Log Manager. getReportList supports many filtering parameters, which let you select and include proper report names in your API calls.

getDataModel

Returns the Common Event Grammar (CEG) in XML format. You select CEG terms you want to include in API call filtering.

getIdealModel

Returns the ideal models defined in the CEG. You select broad product area terms you want to include in API call filtering.

getIncidentModel

Returns the available CEG fields used in Incidents generated by event correlation.

getCombinedModel

Returns the Common Event Grammar (CEG) in XML format for both event and incident fields. You select CEG terms you want to include in API call filtering.

getGlobalSettings

Returns the global settings for the CA Enterprise Log Manager server against which the command is run. You can understand what filtering is already in place for CA Enterprise Log Manager queries so you can build effective API call filters.

getELMServers

Returns a list of CA Enterprise Log Manager servers. This command is useful in a federated environment, because it allows you to target the parent or child servers you want to query.

getTimeZones

Gets a list of time zones that can be used as arguments in running queries.

getVersion

Returns the ELM version, which is the same as the version of the APIs, useful for diagnostic purposes.

getObjectDefinition

Returns the metadata for a report or query given a specific object id. Metadata is all the formatting data that governs how a report or query is presented. Use the metadata when you must use the runQuery call to acquire CA Enterprise Log Manager data for an application that cannot embed query or report viewer directly.

getQueryViewer

Returns the HTML containing the query viewer component preloaded with a specified query.

getReportViewer

Returns the HTML containing the report viewer component preloaded with a specified report.

All the GetObject commands, with the exception of getQueryViewer and getReportViewer, return an error if there is no authenticated session in the API command:

```
Failed Response:
<?xml version="1.0" encoding="UTF-8" ?>
<Result>
<Value>false</Value>
        <Description> User is not logged in</Description>
</Result>
```

In the preceding example, the result value was “false” which indicates a failure, and the description contains the reason, in this case: “User is not logged in”.

More information:

Query and Report Viewer Calls (see page 34)

getQueryList

Use the getQueryList command to list all the queries available in your CA Enterprise Log Manager environment. The XML response also contains the formatting data and any predefined filtering criteria for each query.

You can use the following optional parameters with the getQueryList command

tag

Defines a tag that exists in the system. You can include one or more tags to search for using the getQueryList command. If you specify an unknown tag the command returns an empty list.

tagLogic

Specifies how the getQueryList command treats multiple tags. Supported values are AND and OR. The default value is OR. You can only use one tagLogic value at a time.

Unfiltered Tag Example

https://ELMSERVER:5250/spin/calmapi/getObject.csp?type=getQueryList

Returns all queries and all formatting data associated with each one

OR TagLogic Example

https://ELMSERVER:5250/spin/calmapi/getObject.csp?type=getQueryList&tag=Unknown Category&tag=System

Returns all queries that are associated with the tags “Unknown Category” OR “System”

AND TagLogic Example

https://ELMSERVER:5250/spin/calmapi/getObject.csp?type=getQueryList&tag=Unknown
Category&tag=System&tagLogic=and

Returns all queries that are associated with the tags “Unknown Category” AND “System”

Result Example

This abbreviated example shows only one query, "System Event Count by Event Category".

```
<?xml version="1.0" encoding="UTF-8" ?>
<Result>
        <Value>true</Value>
        <Description>Get Object Successful. Type [getQueryList]</Description>
        <Items>
        <Item edit="false">
                <Panel id="Subscription/panels/System_Event_Count_by_Event_Category" name="System Event Count by
Event Category" subscription="true" version="12.0.46.8">
                        <Description>Ranks system event count activity by event category</Description>
```

```
            <Tags>
                <Tag name="System" />
            </Tags>
            <Query id="">
                <Table>view_event</Table>
                <Args unique="false" />
                <Column columnname="event_datetime" datatype="T" displayname="Date"
resultname="event_datetime" visible="true" />
                <Column columnname="event_category" datatype="S" displayname="Category"
grouporder="1" notnull="true" resultname="event_category" sortdesc="" visible="true" />
                <Column columnname="event_count" datatype="I" displayname="Count"
functionname="sum" resultname="event_count" sortdesc="true" sortorder="1" visible="true" />
            </Query>
            <Display>
                <X name="Category" resultname="event_category" />
                <Y name="Count" resultname="event_count" />
                <Visualization type="VizBarChart" />
                <Visualization type="VizPieChart" />
                <Visualization type="VizTable" />
            </Display>
        </Panel>
    </Item>
    <Item edit="false">
```

"Panel id=" shows that it is a subscription report, and its name.

Note: If the query is a prompt query, the "Prompt id=" tag appears rather than the "Panel id=" tag, "Prompt id=HostPrompt" for example.

"Tag Name=" indicates that it has the System Tag.

The "Column columname=" elements specify the event columns searched by the query, and how they are grouped and ordered.

The "Display" elements specify how the events are shown graphically.

More information:

getQueryViewer (see page 35)
Prompt Queries (see page 46)
runQuery (see page 49)

getReportList

Use the getReportList command to list of all reports available in your CA Enterprise Log Manager environment. The XML response also contains the formatting data and the ID of each query used in the report.

You can use the following optional parameters with the getReportList command:

tag

Defines a tag that exists in the system. You can include one or more tags to search for using the getReportList command. If you specify an unknown tag the command returns an empty list.

tagLogic

Specifies how the getReportList command treats multiple tags. Supported values are AND and OR. The default value is OR. You can only use one tagLogic value at a time.

Unfiltered Tag Example

```
https://ELMSERVER:5250/spin/calmapi/getObject.csp?type=getReportList
```

Returns all reports and all the formatting and display data associated with each one.

OR TagLogic Example

```
https://ELMSERVER:5250/spin/calmapi/getObject.csp?type= getReportList&tag=Unknown
Category&tag=System
```

Returns all reports that are associated with the tags “Unknown Category” OR “System”.

AND TagLogic Example

```
https://ELMSERVER:5250/spin/calmapi/getObject.csp?type= getReportList&tag=Unknown
Category&tag=System&tagLogic=and
```

Returns all reports that are associated with the tags “Unknown Category” AND “System”.

getObjectDefinition

Use the getObjectDefinition command to show the formatting and layout data specific to a query or report in XML format. You can view formatting data in existing reports to create custom formatting, especially using the runQuery command. You can use getObjectDefinition to return data on both subscription and custom user-defined reports or queries.

getObjectDefinition Example

```
https://ELMSERVER:5250/spin/calmapi/getObject.csp?type=getObjectDefinition&objectId=Subscription/panels/Unclassified_Event_Trend
```

Returns the following XML:

```
?xml version="1.0" encoding="UTF-8" ?>
<Result>
        <Value>true</Value>
        <Description>Get Object Successful. Type [getObjectDefinition]</Description>
        <Panel id="Subscription/panels/Unclassified_Event_Trend" name="Unclassified Event Trend" shortname="Trend" subscription="true" version="12.0.46.5">
                <Description>Provides Trending for unclassified event activity</Description>
                <Tags>
                        <Tag name="Unclassified Event" />
                        <Tag name="Unknown Category" />
                </Tags>
                <Params />
                <Query>
```

This example shows the formatting data for the Unclassified Event Trend query. The “objectId” parameter in the call specifies which query or report formatting to display. In this case, it is the Unclassified Event Trend query in the Subscription queries folder.

More information:

runQuery (see page 49)

getDataModel

Use the getDataModel command to show formatting data specific to the Common Event Grammar (CEG). The CEG contains all the possible event fields contained in the schema, a description of each field and possible values for each field (if applicable). You can correctly identity CEG fields for any filtering you want to include in a call.

getDataModel Example

https://ELMSERVER:5250/spin/calmapi/getObject.csp?type=getDataModel

Returns the following XML:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Result>
<Value>true</Value>
<Description>Get Object Successful. Type [getDataModel]</Description>
<CommonEventGrammar version="12.0.45.4">
                                ....
<field name="event_logname" type="S" class="" category="event" index="y" desc="The name of the log expressed in the event
information.">
<values>
                                <value>ACF2</value>
                                <value>Apache</value>
                                <value>AuditEngine</value>
```

The "field name=" element displays the CEG field, in this case event_logname.

Each CEG field has a type, which is displayed in the "type=" element.

getCombinedModel

Use the getCombinedModel command to show formatting data specific to the Common Event Grammar (CEG) for both events and incidents. The CEG contains all the possible event fields contained in the schema, a description of each field and possible values for each field (if applicable). You can correctly identity CEG fields for any filtering you want to include in a call.

getCombinedModel Example

```
https://ELMSERVER:5250/spin/calmapi/getObject.csp?type=getCombinedModel
```

Returns the following XML:

```
                              <?xml version="1.0" encoding="UTF-8" ?>

- <Result>

  <Value>true</Value>

  <Description>Get Object Successful. Type [getCombinedModel]</Description>

- <CEGFields>

+ <events>

- <incidents>

- <CommonEventGrammar version="12.1.5109.0">

  <SchemaVersion value="1" desc="Incident Schema version, integer, incremented starting at 1 (no version=0)" />

  <field internal="true" name="" type="" class="" category="" index="" desc="" dbtable="version" dbname="value"
dbtype="INTEGER" dbindex="NOT NULL" />

  <field internal="true" name="" type="" class="" category="" index="" desc="" dbtable="version"
dbname="timestamp" dbtype="INTEGER" dbindex="NOT NULL" />

  <field name="incident_id" type="S" class="" category="" index="y" desc="" dbtable="incidents"
```

The "field name=" element displays the CEG field, in this case incident_id.

The "dbtable=" element identified the database type, in this case the incident database.

getIncidentModel

Use the getIncidentModel command to show Common Event Grammar (CEG) fields specific to incidents in your environment. The CEG contains all the possible event fields, a description of each field and possible values for each field (if applicable). You can correctly identity CEG fields for any filtering you want to include in a call.

getIncidentModel Example

```
https://ELMSERVER:5250/spin/calmapi/getObject.csp?type=getIncidentModel
```

Returns the following XML:

```
<?xml version="1.0" encoding="UTF-8" ?>
- <Result>
  <Value>true</Value>
  <Description>Get Object Successful. Type [getIncidentModel]</Description>
- <CommonEventGrammar version="12.1.5109.0">
  <SchemaVersion value="1" desc="Incident Schema version, integer, incremented starting at 1 (no version=0)" />
  <field internal="true" name="" type="" class="" category="" index="" desc="" dbtable="version" dbname="value" dbtype="INTEGER" dbindex="NOT NULL" />
  <field internal="true" name="" type="" class="" category="" index="" desc="" dbtable="version" dbname="timestamp" dbtype="INTEGER" dbindex="NOT NULL" />
  <field name="incident_id" type="S" class="" category="" index="y" desc="" dbtable="incidents" dbname="producer_msg_id" dbindex="UNIQUE NOT NULL" SnmpOID="1.3.6.1.4.1.791.9845.2.1001" />
  <field name="incident_createtime_gmt" type="T" class="" category="" index="y" desc="" dbtable="incidents" dbname="createtime" SnmpOID="1.3.6.1.4.1.791.9845.2.1002" />
  <field name="incident_name" type="S" class="" category="" index="y" desc="" dbtable="incidents" dbname="name" SnmpOID="1.3.6.1.4.1.791.9845.2.1003" />
  <field name="incident_rule_id" type="S" class="" category="" index="y" desc="" dbtable="incidents" dbname="rule_id" SnmpOID="1.3.6.1.4.1.791.9845.2.1004" />
  <field name="incident_rule_version" type="S" class="" category="" index="y" desc="" dbtable="incidents" dbname="rule_version" SnmpOID="1.3.6.1.4.1.791.9845.2.1005" />
  <field name="incident_rule_grouppath" type="S" class="" category="" index="y" desc="" dbtable="incidents" dbname="rule_grouppath" SnmpOID="1.3.6.1.4.1.791.9845.2.1006" />
```

The "field name=" element displays the incident CEG field.

getELMServers

Use the getELMServers command to return a list of available CA Enterprise Log Manager servers to run queries against.

getELMServers Example

https://ELMSERVER:5250/spin/calmapi/getObject.csp?type=getELMServers

Returns the following XML:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Result>
        <Value>true</Value>
        <Description>Get Object Successful. Type [getELMServers]</Description>
        <service type="service" name="Event Log Store" id="/CALM_Configuration/Modules/logDepot/Config" edit="true"
updated="1232571794" global_config="true">
        <service type="host" name="machinename" id="/CALM_Configuration/Modules/logDepot/machinename/Config"
edit="true" service_name="Event Log Store" updated="1232571795" />
        </service>
</Result>
```

This example shows only one server, where the attribute “type=host” indicates a CA Enterprise Log Manager server host name, in this case "machinename". There can be one or more hosts specified. Each XML "service" element represents a single CA Enterprise Log Manager server.

getGlobalSettings

Use the getGlobalSettings command to show global settings for the target CA Enterprise Log Manager server. You can view the global settings and decide if they are suitable for any API query or report calls you want to construct. The settings are controlled from the CA Enterprise Log Manager interface.

getGlobalSettings Example

https://ELMSERVER:5250/spin/calmapi/
getObject.csp?type=getIGlobalSetthttps://ELMSERVER:5250/spin/calmapi/getObject.csp?type=getGlobalSetting
sings

Returns the following XML:

```
<?xml version="1.0" encoding="UTF-8" ?>
- <Result>
  <Value>true</Value>
  <Description>Get Object Successful. Type [getGlobalSettings]</Description>
- <iSponsor>
  <Name>CALM</Name>
  <Version>12.1.xxx.1</Version>
  <EEMServer>etr851l1-blade3</EEMServer>
  <EEMAdmin>EiamAdmin</EEMAdmin>
  <Certificate>/opt/CA/SharedComponents/iTechnology/CAELMCert.p12</Certificate>
  <Password>BhUXVFhQCFxEDA==</Password>
  <DisplayName>Global Configuration</DisplayName>
  <CalmType>service</CalmType>
  <AppInstance>CAELM</AppInstance>
  <ELMPath>/opt/CA/LogManager</ELMPath>
  <Updated>1269421754</Updated>
  <KeyFile>@APP_NAME@Cert.key</KeyFile>
  <UpdateInterval label="Update Interval (seconds)" def="300" prompt="Update interval in seconds at which
components checks for updated configurations" type="number" min="30" max="86400"
global="true">30</UpdateInterval>
  <SessionTimeout label="Session Timeout (minutes)" def="15" prompt="Session timeout in minutes"
type="number" min="10" max="600">15</SessionTimeout>
  <AutoRefreshAllowed type="bool" label="Allow Auto Refresh" prompt="Allow users to set auto refresh of reports"
def="false">true</AutoRefreshAllowed>
  <AutoRefreshFrequency type="number" label="Auto Refresh Frequency (minutes)" prompt="Auto refresh
frequency in minutes" min="1" max="60" def="10">10</AutoRefreshFrequency>
  <AutoRefreshEnabled type="bool" label="Enable Auto Refresh" prompt="Enable auto refresh of reports"
def="false">false</AutoRefreshEnabled>
  <AlertAuthentication def="true" label="Viewing Action Alerts Requires Authentication" prompt="Requires
authentication for Viewing action alerts" type="bool" global="true">false</AlertAuthentication>
  <DefaultReport EEMDisplay="calmName"
EEMsource="/CALM_Configuration/Content/Reports/Subscription/scorecards,/CALM_Configuration/Content/Rep
orts/User" calmType="scorecard" label="Default Report" prompt="The default report to run"
type="combo">Collection_Monitor_by_Log_Manager</DefaultReport>
  <EnableDefaultReport type="bool" label="Enable default report launch" prompt="Enable automatic launch of
default report" def="true">true</EnableDefaultReport>
  <HiddenReportTags type="shuttle" prompt="Hide selected report tags view in the application." icon="tagIcon"
label="Hide Report Tags" EEMsource="/CALM_Configuration/Content/Reports/Tags/Report" orderedlist="false" />
  <HiddenQueryTags type="shuttle" prompt="Hide selected query tags view in the application." icon="tagIcon"
label="Hide Query Tags" EEMsource="/CALM_Configuration/Content/Reports/Tags/Panel" orderedlist="false"
global="true" />
```

```
 <EnableDefaultProfile group="Profiles" type="bool" label="Enable default profile" prompt="Enable automatic
launch of default profile" def="false">false</EnableDefaultProfile>
 <DefaultProfile group="Profiles" EEMDisplay="calmName"
EEMsource="/CALM_Configuration/Content/Profiles/Subscription,/CALM_Configuration/Content/Profiles/User"
calmType="profile" label="Default Profile" prompt="The default profile to run" type="combo"
global="true">CA_Access_Control</DefaultProfile>
 <HiddenProfiles group="Profiles" EEMDisplay="calmName" type="shuttle" prompt="Hide selected profiles view in
the application." icon="profileIcon" label="Hide Profiles"
EEMsource="/CALM_Configuration/Content/Profiles/Subscription,/CALM_Configuration/Content/Profiles/User"
orderedlist="false" global="true">CA_Identity_Manager</HiddenProfiles>
 </iSponsor>
 </Result>
```

getTimeZones

Use the getTimeZones command to show time zones that are supported as a query parameter. You can use it to get a list of time zones so that the query data is returned using the proper time zone formatting.

Note: If you do not provide a valid time zone for getQueryViewer, getReportViewer, and runQuery the data is returned in the CA Enterprise Log Manager server time zone.

getTimeZones Example

https://ELMSERVER:5250/spin/calmapi/getObject.csp?type=getTimeZones

Returns the following XML:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Result>
        <Value>true</Value>
        <Description>Get Object Successful. Type [getTimeZones]</Description>
        <tz>
                <TimeZone isDefault="false">Etc/GMT+12</TimeZone>
                <Offset>720.0</Offset>
        </tz>
        <tz>
                <TimeZone isDefault="false">Etc/GMT+11</TimeZone>
                <Offset>660.0</Offset>
        </tz>
        ....
```

getVersion

You can use the getVersion command to show the APIs version running on the target CA Enterprise Log Manager server. The versions need not be the same. Use this command for troubleshooting purposes.

Note: The API version can differ from the version of other CA Enterprise Log Manager components, such as agents, depending on the update choices made by your administrator.

getVersion Example

```
https://ELMSERVER:5250/spin/calmapi/ getObject.csp?type=getVersion
```

Returns the following XML:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Result>
        <Value>true</Value>
        <Description>Get Object Successful. Type [getVersion]</Description>
        <Version>v12.0.48.14</Version>
</Result>
```

Query and Report Viewer Calls

GetQueryViewer and getReportViewer, return a graphical viewer interface window similar to the CA Enterprise Log Manager interface. You can perform many of the tasks associated with reports or queries from this window. For more information about the available tasks, consult the *CA Enterprise Log Manager API Online Help*.

These calls provide external integration points with third-party portals and other applications. When using them, consider the following:

- Using certificate authentication means that the report or query viewer session does not time out as a CA Enterprise Log Manager session does. The application from which you call the event or query viewer controls the time-out, rather than the CA Enterprise Log Manager application.
- For security reasons, if you have not registered a third-party product with CA Enterprise Log Manager, these calls redirect to the login page. You can avoid the redirect to log in by using one of the following techniques:
    - Include the credentials attributes as a hidden field in every command. The API spindle automatically authenticates, which works with some portals that allow the setting of hidden fields.
    - Perform a command such as getVersion before launching or embedding the UI component and take appropriate action (such as reauthenticating behind the scenes) as needed.

More information:

About API Sessions (see page 18)
getQueryViewer (see page 35)
getReportViewer (see page 47)
API Authentication (see page 15)

getQueryViewer

Use this call to display the graphical viewer for a specific query. The viewer is a fully functional CA Enterprise Log Manager query viewer delivered as a standalone component. You can embed specific queries in an outside application interface or external portal, by embedding the URL in an iFrame.

Note: The solution provided here works with web-based applications such as JSPs, JavaScript, and HTML. The solution might *not* work in C++ or Java Swing applications depending on the availability and support for an embedded HTML page and the necessary FLASH plug-in support of the applications. For applications where FLASH cannot be supported, we recommend that you use runQuery to retrieve the raw data and then render it using a method appropriate to your environment.

getQueryViewer Example

```
https://ELMSERVER:5250/spin/calmapi/getObject.csp?type=getQueryViewer&objectId=Subscription/panels/Syst
em_Event_Count_By_Event_Action
```

Displays the System Event Count by Event Action query.

"getObject.csp?type=getQueryViewer" specifies the type of getObject call, in this case the query viewer.

"&objectId=Subscription/panels/System_Event_Count_By_Event_Action" identifies the specific query, in this case the Subscription query named System Event Count by Event Action. Any query name can be specified by entering the title as it appears in the interface, separated by underscores.

More information:

getQueryList (see page 24)
Prompt Queries (see page 46)
runQuery (see page 49)

Query Specifications

You can prequalify the results of a getQueryViewer, run getReportViewer, or runQuery call by adding specifications. You can set present detailed information that can be a subset of an existing query, or relevant to particular consumers. For example, you can use specifications to query one server for a certain type of events in the past day only.

You can set the following specifications:

server

Specifies the CA Enterprise Log Manager server queried. The default is localhost, the server named in the getQuery call. You can use this specification to target a different server.

timezone

Defines the time zone in which the query appears. The default is the time zone in which the CA Enterprise Log Manager server is running. You can use this specification to set your results in a different time zone.

federated

Specifies (using true or false) whether the query is applied to the appropriate federated servers. The default value is true, which applies the query across federated servers. This behavior applies the normal CA Enterprise Log Manager rules for querying federation hierarchies.

filterXml

Defines the data filters applied to the query, in XML format. You could use this specification to filter on hostname or other CEG fields.

IncidentFilterXml

Defines the data filters applied to an incident query included in a report, in XML format. You could use this specification to filter on the incident creation time or other CEG fields. This specification applies only to the getReportViewer call.

accessfilterXml

Defines the data filters applied to the query, in XML format. You could use this specification to filter a query or report result according to your role when you use the certificate name and password authentication.

params

Defines the result conditions applied to the query, in XML format.

prompt

Controls (using true or false) whether the additional prompt controls are displayed. The default value is false. This value is only valid if the query type is prompt. The value is ignored if the query is not a prompt.

The following specifications are only used if you have set "prompt=true":

promptvalue

Sets the filter value for a prompt query.

col

Lists the event columns the prompt query searches. You can use multiple col terms to identify more than one target column.

More information:

getQueryViewer (see page 35)
Prompt Queries (see page 46)
runQuery (see page 49)

Server Specifications

You can specify the event log store of a non-default CA Enterprise Log Manager server as the target of the query, by name or IP address. The default is localhost, the server named in the API call.

You can use getELMServers to retrieve a list of eligible server names.

Server Name Specification Example

```
https://ELMSERVER:5250/spin/calmapi/getObject.csp?type=getQueryViewer&objectId=Subscription/panels/System_Event_Count_By_Event_Action&server=ELMSERVER2
```

In this example, "&server=" specifies the name of the server for the query. The server name you want replaces "ELMSERVER2". Because the default is the localhost (ELMSERVER) there is no need to use the &server element unless you want to specify a non-default target server.

Note: If you enter an invalid server name, the call returns data from the default CA Enterprise Log Manager server identified by the ELMSERVER value.

More information:

getELMServers (see page 31)
runQuery (see page 49)

Time Zone Specifications

You can add a time zone specification to your getQuery or runQuery call. You can retrieve a list of available time zones using getTimeZones.

Time Zone Specification Example

```
https://ELMSERVER:5250/spin/calmapi/getObject.csp?type=getQueryViewer&objectId=Subscription/panels/System_Event_Count_
By_Event_Action&timezone=TIMEZONENAME
```

In this case, "&timezone=" specifies the name of the time zone you want. Your time zone name replaces "TIMEZONENAME", as displayed in the list returned by the getTimeZones call.

Note: The response to an invalid time zone differs depending on the call you include it in:

- If an invalid time zone is used in the runQuery call, GMT timestamps are returned. If no time zone is passed then the default is the time zone where the server is running.
- If no time zone, or an invalid time zone is used in the getQueryViewer or getReportViewer call then the default is the target server time zone.

More information:

getTimeZones (see page 33)
runQuery (see page 49)

More information:

IncidentFilter XML Specifications (see page 42)

Filter XML Specifications

You can preset CA Enterprise Log Manager filters for your report in XML format and add them to the getQueryViewer, getReportViewer, getIncidentViewer, or runQuery URL using the filterXML term. You can nest multiple filters, using AND and OR terms and parentheses. You are essentially creating CA Enterprise Log Manager advanced filters in XML.

Important! FilterXml terms are complex, and the API performs no validation. Invalid filter terms result in a query error. For this reason, we recommend that you take particular care in constructing your filter terms.

The available filter elements, listed in the order in which they must be used, are as follows:

lparens

Defines the number of left parentheses. The valid values are 0 or more.

logic

Sets the logical term connecting filters; AND or OR. For the first filter term, always leave the logic value empty.

col

Defines the event columns queried. You can get the list of available columns by using getDataModel.

oper

Defines an operator for the filter. The valid case-sensitive values are:

- EQUAL - Equal to
- NEQ - Not Equal to
- LESS - Less than
- GREATER - Greater than
- LEQ - Less than or Equal to
- GREATEQ - Greater than or Equal to
- LIKE - Like
- NOTLIKE - Not Like
- INSET - In Set
- NOTINSET - Not in Set
- MATCH - Matches

- KEYED - Keyed
- NOTKEYED - Not Keyed

val

Defines the value for which the filter searches.

rparens

Defines the number of right parentheses. The valid values are 0 or more. The total number of right parenthesis always matches the number of left parenthesis.

When you view a graphical query or report, you can view or adjust the FilterXML terms you set from the advanced filter section of the Local Filter dialog in the viewer interface.

Filter XML Specification Example

This example shows a getQueryViewer call with a filter statement. The filter terms are shown expanded for clarity.

```
https://ELMSERVER:5250/spin/calmapi/getObject.csp?type=getQueryViewer&objectId=Subscription/panels/System_Event_Count_
By_Event_Action&server=ELMSERVER&filterXml=
  <Filter logic="" lparens="1" col="source_username" oper="LIKE" val="su" rparens="0"/>
  <Filter logic="AND" lparens="0" col="event_logname" oper="LIKE" val="CALM" rparens="1"/>
</Scope>
```

"&filterxml=" specifies that a filter statement follows.

The filter statement sets the query to search the source_username column for "su", and the event_logname column for "CALM". Because an AND statement joins the two terms (Filter logic="AND"), only events where each value is found in its respective columns are returned.

Access Filter XML Specifications

You can preset CA Enterprise Log Manager filters for your query or report in XML format when you authenticate using the certificate name and certificate password mechanism. An access filterXML passed in a login call is applied to all the queries and reports run in that session. If you pass a filterXML in the query or report after logging in with an access filterXML, CA Enterprise Log Manager applies both the filters to retrieve results.

The access filter XML elements are similar to the filter XML elements.

Access Filter XML Specification Example Without a Filter XML

This example shows a getQueryViewer call with an access filter XML statement. The filter terms are shown expanded for clarity.

```
https://ELMSERVER:5250/spin/calmapi/getObject.csp?type=getQueryViewer&objectId=Subscription/panels/System_Event_Count_
by_Event_Source&certname=test&password=test&accessFilterXml=<AccessScope><Filter logic="" lparens="0"
col="event_logname"oper="LIKE" val="CALM" rparens="0"/></AccessScope>
```

"&accessFilterXml=" specifies that an access filter statement follows.

Access Filter XML Specification Example With a Filter XML

This example shows a objectId call with a filter and access filter XML statements.

```
https://ELMSERVER:5250/spin/calmapi/runQuery.csp?objectId=Subscription/panels/System_Event_Count_by_Event_Source&filter
Xml=<Scope><Filter logic="" lparens="1"  col="event_logname" oper="INSET" val="'CALM','Unix'"
rparens="1"/></Scope>&certname=test&password=test&accessFilterXml=<AccessScope><Filter logic="" lparens="1"
col="event_logname" oper="LIKE" val="CALM" rparens="1"/></AccessScope>
```

"&filterXml=" specifies that a filter statement follows.

"&accessFilterXml=" specifies that an access filter statement follows.

IncidentFilter XML Specifications

You can preset CA Enterprise Log Manager filters for an incident report in XML format and add them to the getReportViewer URL using the IncidentFilterXML term. You can nest multiple filters, using AND and OR terms and parentheses. IncidentFilter specifications work in the same way as Filter specifications, and share the same elements and operators.

IncidentFilter XML specifications apply only to incident queries contained in reports. However, a report can contain both event and incident queries. To access and filter such a report, your API URL can contain both Filter XML and IncidentFilter XML specifications.

Important! IncidentFilterXml terms are complex, and the API performs no validation. Invalid filter terms result in a query error. For this reason, we recommend that you take particular care in constructing your filter terms.

IncidentFilter XML Specification Example

This example shows a getReportViewer call with a filter statement. The filter terms are shown expanded for clarity.

```
https://ELMSERVER:5250/spin/calmapi/getObject.csp?type=getQueryViewer&objectId=Subscription/panels/Incidents_by_Priority=ELMSERVER&incidentfilterXml=

<Filter logic="AND" lparens="1" col="incident_createtime_gmt" colfunc="" oper="GREATEQ" val="1285854741" rparens="0" filterTag="" substituteValue="false" isDynamic="true"/>

<Filter logic="AND" lparens="0" col="incident_createtime_gmt" colfunc="" oper="LEQ" val="1285876341" rparens="1" filterTag="" substituteValue="false" isDynamic="true"/>
```

"&incidentfilterxml=" specifies that an incident filter statement follows.

The filter specifies all incidents created within a specified time span.

More information:

Filter XML Specifications (see page 39)

Result Condition Specifications

Use param terms to set result conditions for a getQueryViewer, getReportViewer, or runQuery call.

The following param terms are available:

ARG_limit

Sets the number of rows returned by the query

ARG_show_other

Sets whether the show other column appears in a query viewer display, using true or false. This option is used for charts having Top N Queries (Aggregated Queries with Row Limit Set and aggregated based on event_count). If this option is selected, the first N -1 (N being the Row Limit) events are displayed normally. However, the Nth event is the "Other Event" which is an aggregated event based on the rest of the events.

ARG_event_datetime

Sets the granularity level of the time period used in the query display for trend queries. The available values are:

- event_datetime
- event_day_datetime
- event_minute_datetime
- event_hour_datetime
- event_month_datetime
- event_year_datetime

ARG_start

Sets the dynamic start time for the query.

ARG_stop

Sets the dynamic end time for the query.

ARG_minduring

Defines the earliest grouped event dated after a specified dynamic time. Relevant only for grouped queries.

ARG_maxduring

Defines the latest grouped event dated after a specified dynamic time. Relevant only for grouped queries.

ARG_maxbefore

Defines the latest grouped event dated before a specified dynamic time. Relevant only for grouped queries.

ARG_sumatleast

Defines the minimum number of events for grouping. Relevant only for grouped queries.

ARG_sumatmost

Defines the max number of events in the grouping. Relevant only for grouped queries.

Result Condition Specification Example

This example is shown with the params terms expanded for clarity.

```
https://ELMSERVER:5250/spin/calmapi/getObject.csp?type=getQueryViewer&objectId=Subscription/panels/System_Event_Count_
By_Event_Action
<Params>
  <Param id="ARG_limit" val="'200'"/>
</Params>
```

The "ARG_limit" value '200' sets the query to display only the first 200 rows.

More information:

runQuery (see page 49)
Dynamic Time Terms (see page 44)

Dynamic Time Terms

Use dynamic time params terms to specify the time ranges to which a query applies by adding them to certain result conditions specifications.

The available dynamic time params terms are as follows:

Term	Description
now	The current time
start of day	Start of the current day

weekday <number>	Numbered day of the week: ■ Sunday 0 ■ Monday 1 ■ Tuesday 2 ■ Wednesday 3 ■ Thursday 4 ■ Friday 5 ■ Saturday 6
start of month	Start of the current month
start of year	Start of the current year
<number> seconds	Number of seconds
<number> minutes	Number of minutes
<number> hours	Number of hours
<number> days	Number of days

You can specify result conditions for a Query Definition or Report Definition. In this case any time specifications you add to the call override the values specified in the base Query or Report.

In both cases, any values not specified in the URL remain unchanged.

Dynamic Time Terms Specification Example

This example is show with the params terms expanded for clarity.

```
https://ELMSERVER:5250/spin/calmapi/getObject.csp?type=getQueryViewer&objectId=Subscription/panels/Syst
em_Event_Count_By_Event_Action
<Params>
  <Param id="ARG_start" val="'now', '-12 hours'"/>
  <Param id="ARG_stop" val="'now'"/>
</Params>
```

The "ARG_start" values 'now' and '-12 hours' set the query to start 12 hours ago.

The "ARG_stop" value 'now' sets the query to end at the current time, so this query would gather data only from the last 12 hours.

More information

Result Condition Specifications (see page 43)
runQuery (see page 49)

Prompt Queries

Prompts are specialized queries that allow you to enter certain filter values before running the query. You can use getQueryList to view the available prompt queries. The "Prompt id" element identifies a prompt query, which appears in place of the "Panel id" element that identifies standard queries. You can use the prompt, promptvalue, and col terms to define prompt queries you want to call.

You can access the graphical prompt query without the filter values specified, or prespecify them in the URL. If no columns are provided in the URL, all the prompt columns are selected.

Unfiltered Host Prompt Example

https://ELMSERVER:5250/spin/calmapi/getObject.csp?type=getQueryViewer&objectId=Subscription/panels/HostPrompt

Displays the Host Prompt without any filter value entered, but with all prompt columns selected.

Filtered IP Prompt Example

https://ELMSERVER:5250/spin/calmapi/getObject.csp?type=getQueryViewer&objectId=Subscription/panels/IPPrompt&prompt=true&promptvalue=255.255.255.0&col=dest_address

Runs the IP Prompt, searching the destination address column for the IP address 255.255.255.0

"&prompt=true" shows the prompt controls, which lets you modify the prompt query values after its run, and run the query again as needed.

"&promptvalue=" specifies the IP address you want.

"&col=dest_address" selects the event column you want.

More information:

getQueryList (see page 24)
runQuery (see page 49)

getReportViewer

You can use the getReportViewer command to display the graphical viewer for a specific report. The report viewer is similar to the CA Enterprise Log Manager interface report viewer, delivered as a standalone component. You can embed specific reports in an outside application interface or external portal, usually by embedding the URL in an iFrame or portlet.

Note: The solution provided here works with web-based applications such as JSPs, JavaScript, and HTML. The solution might *not* work in C++ or Java Swing applications depending on the availability and support for an embedded HTML page and the necessary FLASH plug-in of those applications. For those applications where FLASH cannot be supported, we recommend that you use getReportList to determine which queries are included in the report, and then use runQuery for each report to retrieve the raw data and then render it using a method appropriate to your environment.

getReportViewer Example

This example calls the Collection Monitor by Log Manager Report.

```
https://ELMSERVER:5250/spin/calmapi/getObject.csp?type=getReportViewer&objectId=Subscription/scorecards/Collection_Monitor_by_Log_Manager
```

You can use filter and other specifications for getReportViewer in the same way as you do for getQueryViewer.

Any report name can be specified by entering the title as it appears in the interface, separated by underscores.

More information:

getReportList (see page 26)
runQuery (see page 49)

getIncidentViewer

You can use the getIncidentViewer command to display a graphical incident viewer. The viewer is similar to the CA Enterprise Log Manager interface incident viewer, delivered as a standalone component. Management functions available from within the CA Enterprise Log Manager interface, such as merging, or deleting incidents cannot be performed from this viewer.

Note: The solution provided here works with web-based applications such as JSPs, JavaScript, and HTML. The solution might *not* work in C++ or Java Swing applications depending on the availability and support for an embedded HTML page and the necessary FLASH plug-in of those applications.

getIncidentViewer Example

```
https://elmserver:5250/spin/calmapi/getObject.csp?type=getIncidentViewer
```

This call displays the Incident Viewer, showing incidents created in the last six hours.

You can use time terms, filters and other specifications for getIncidentViewer in the same way as you do for getQueryViewer.

runQuery

Use runQuery to run a query and return the results in XML, rather than in the graphical query viewer. You can use this method to acquire CA Enterprise Log Manager data for an application that cannot embed the query or report viewer directly, such as those that cannot support Flash.

Add query specifications to the URL to filter the base query, as you do for getQueryViewer.

After using runQuery, format the XML data for display in an appropriate way for your environment. For example, you could embed a runQuery call in a web portal, and apply a style sheet to display the data.

runQuery Example

https://ELMSERVER:5250/spin/calmapi/runQuery.csp?objectId=Subscription/panels/Collection_Monitor_by_Log_Manager_By_Log_Name

Returns the following XML:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Result>
        <Value>true</Value>
        <Description>Query run successful</Description>
        <QueryResults>
                <Version>1</Version>
                <Row number="1">
                        <event_logname>CALM</event_logname>
                        <event_count>581</event_count>
                </Row>
                <Row number="2">
                        <event_logname>EiamSdk</event_logname>
                        <event_count>131</event_count>
                </Row>
                <Result totalrows="2" returnedrows="2" startrow="1" endrow="2" executems="2382" mstofirst="2382" mstolast="2382" />
                <DbResult numberdbsqueried="1" numberdbsresponding="1" numberdbsnotresponding="0" listdbsresponding="../../LogManager/data/hot/machinename_1232571874.hot" listdbsnotresponding="" />
                <HostResult numbberhostsqueried="0" numberhostsresponding="0" numberhostsnotresponding="0" listhostsresponding="" listhostsnotresponding="" />
        </QueryResults>
        SQL ServerSELECT event_logname , SUM(event_count) AS FUNC_SUM_event_count FROM view_event WHERE ( ( datetime(event_time_gmt, 'unixepoch') >= datetime('now', '-6 hours') and datetime(event_time_gmt, 'unixepoch') < datetime('now') ) AND ( event_category = ? ) ) GROUP BY event_logname ORDER BY FUNC_SUM_event_count DESC LIMIT 10 ; [Operational Security]</Sql>
</Result>
```

More information:

getQueryViewer (see page 35)
getReportViewer (see page 47)

API Registration

This section contains information about registering products with CA Enterprise Log Manager. You can use the API product registration page to create registration certificates, allowing single sign-on from outside products. You can register multiple products, using a single interface, rather than having to construct individual registration calls. The product registration pages allow certificate creation in nearly all cases.

This section also contains calls allowing registration where use of the product registration page, or simple authentication is not preferable or possible.

More information:

API Certificate Creation (see page 50)
Register a Product (see page 54)
Unregister a Product (see page 55)

API Certificate Creation

You can access the API product registration interface page to create single sign-on registration certificates, view a list of registered products, or unregister products by deleting existing certificates.

You can add authentication information to the URL. If you are not authenticated, you are redirected to the CA Enterprise Log Manager log in page. This behavior matches all other API calls that return a user interface.

Note: Use the EiamAdmin username and password to create a registration certificate. To list or unregister products, you c*an* use the EiamAdmin credentials, but Administrator credentials are sufficient.

Certification Page Display Example

URL: https://ELMSERVER:5250/spin/calmapi/products.csp

Displays the CA Enterprise Log Manager log in page. When you enter the appropriate credentials, the product registration page appears.

See the *CA Enterprise Log Manager API Help*, accessible from the product registration page, for more information about creating certificates.

More information:

Register a Product (see page 54)
Query and Report Viewer Calls (see page 34)
API Authentication (see page 15)

How to Register a Product with CA Enterprise Log Manager

You can register a product with CA Enterprise Log Manager to allow single sign-on. You can access CA Enterprise Log Manager queries and reports from a password management, access management, or other application depending on your needs. The registration process has two steps:

1. Create a registration certificate in CA Enterprise Log Manager.
2. Use the registration certificate name and password from your outside product to complete the single sign-on registration.

   The exact procedure for this step differs depending on the specific product you want to register with CA Enterprise Log Manager. However, have the following information available to complete the registration:

   - The CA Enterprise Log Manager server hostname or IP address where you want to register.
   - The certificate name you create in Step 1.
   - The certificate password you create in Step 1.

More information:

Create a Registration Certificate (see page 52)

Create a Registration Certificate

You can create a registration certificate to allow single sign-on from other CA or third-party products.

To create a registration certificate

1. Open a web browser and enter the following URL:

   https://calmserver:5250/spin/calmapi/products.csp

   Replace "calmserver" with the server name or IP address of the CA Enterprise Log Manager server where you want to register products.

   Unless you are already authenticated as the EiamAdmin user, the login screen appears. If you are already authenticated, the product registration page appears.

2. Enter the Eiamadmin user name and password.

   A list of all current registration certificates appears.

   Note: You must have EiamAdmin user credentials to create a certificate. Administrator credentials are sufficient to list or unregister products.

3. Click the Register link above the Registered Products list in the left pane.

4. Enter a name for the product you want to register, and a password.

   Note: Be sure to record the certificate name and password. You need them for the completion of the registration process from the outside product.

5. Click the Register button in the right pane.

   A confirmation message appears, and the certificate name appears in the Registered Products list.

Unregister a Product

You can unregister a product by deleting the registration certificate.

To unregister a product

1. Open a web browser and enter the following URL:

   ```
   https://calmserver:5250/spin/calmapi/products.csp
   ```

   Replace "calmserver" with the server name or IP address of the CA Enterprise Log Manager server where you want to unregister products.

   The login screen appears.

2. Enter an Administrator role user name and password.

   A list of all current registration certificates appears.

3. Click the registration certificate you want to delete.

4. Click Unregister.

   A confirmation dialog appears.

5. Click OK.

   A confirmation message appears, and the certificate name is removed from the Registered Products list.

Register a Product

You can use the registerProduct call to register a product for single sign-on purposes. Registering a product creates a certificate which is stored in the management database. You can use this call where it is not possible or preferable to access the product registration interface.

For example, if you are integrating a third-party product, you might not want to distribute the EiamAdmin password widely to allow certificate creation. In that case, you can create a certificate and password and distribute them to those product users to set up integrations.

registerProduct Examples

```
https://ELMSERVER:5250/spin/calmapi/calmapi/registerProduct.csp?action=register&certname=YourProductName&certpassword=CertPassword&certname=xxxxx&password=xxxxxx
```

In this case "&certname=YourProductName" defines the product you want to register. Replace "YourProductName" with the product name you want to register.

"&certname=xxxxx" specifies valid certificate name and password.

Success Response:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Result>
<Value>true</Value>
        <Description>The product has been registered successfully. The default access rights on the ELM application have been provided.</Description>
</Result>
```

Failure Response:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Result>
<Value>false</Value>
        <Description> EE_POZERROR Repository Error</Description>
</Result>
```

Note: An error often occurs when the certificate with the name given in the URL has already been created. Another common error description is “EE_AUTHFAILED Authentication failed” which indicates the password was incorrect.

Unregister a Product

You can use the unregister command to unregister a product. You can use this call where it is not possible or preferable to access the product registration interface to remove a registration certificate.

Unregister Product URL Examples:

```
https://ELMSERVER:5250/spin/calmapi/calmapi/registerProduct.csp?action=unregister&certname=YourProductN
ame&username=Administrator&password=adminpassword
```

In this case "&username=Administrator" specifies a CA Enterprise Log Manager user with the Administrator role. Replace "Administrator" with an appropriate administrator-privileged user.

"&password=adminpassword" specifies password for the Administrator user. Replace "adminpassword" with the password for the user you specified in "&username=".

Note: Use the EiamAdmin username and password to register a product. To list or unregister products, you *can* use the EiamAdmin credentials, but Administrator credentials are sufficient.

Success Response:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Result>
<Value>true</Value>
        <Description>The product has been unregistered successfully. The default access rights have been
revoked. </Description>
</Result>
```

Failure Response:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Result>
<Value>false</Value>
        <Description> EE_POZERROR Repository Error</Description>
</Result>
```

Note: An error often results when the product has already been unregistered or does not exist. Another common error description is “EE_AUTHFAILED Authentication failed” which indicates the password was incorrect.

Chapter 5: How to Embed CA Enterprise Log Manager in a Web Portal

You can embed CA Enterprise Log Manager queries or reports in a web portal, allowing you to display the content you want. The process is as follows:

1. Identify the CA Enterprise Log Manager content you want to display, and construct the API call to identify and return it.
2. Embedding the selected content in the web portal.

More information:

Identifying Content (see page 58)
Embed Content in a Liferay Portal (see page 59)
Query and Report Viewer Calls (see page 34)

Identifying Content

Begin the process of embedding CA Enterprise Log Manager content by deciding what content you want to display. Review the CA Enterprise Log Manager interface to find the report or query that contains the information that meets your needs.

To display CA Enterprise Log Manager queries or reports in a web portal, use the getQueryViewer or getReportViewer calls to display interactive reports and queries with most of the functionality available from within the CA Enterprise Log Manager interface.

You can also use the runQuery report to return XML content, and display it by applying a style sheet. The display is non-interactive, and allows you to display the data without using Flash.

This example calls the System All Events Detail report, using getQueryViewer to show an Event Viewer table of all events. The API call syntax for this report is as follows:

```
https://ELMSERVER:5250/spin/calmapi/getObject.csp?type=getQueryViewer&objectId=Subscription/panels/System_All_Events_Detail&username=xxx&password=xxx
```

- To use the call in your environment, replace the "ELMSERVER" portion of the URL with the hostname or IP address of the server where the data you want is stored.
- This example authenticates using a CA Enterprise Log Manager user name and password: "&username=xxx&password=xxx". Using this authentication method is recommended for embedding CA Enterprise Log Manager content. Replace the 'xxx' with an appropriate CA Enterprise Log Manager username and password. If you do not want the username and password to be visible in the URL, you can set them as hidden values, if your web portal allows.

You can test the final syntax by entering the URL you have built in a browser and confirming that the report or query you want appears.

More information:

API Authentication (see page 15)
Query and Report Viewer Calls (see page 34)
getQueryViewer (see page 35)
getReportViewer (see page 47)
runQuery (see page 49)

Embed Content in a Liferay Portal

After you have an API call that returns the query or report you want, embed it in your web portal, using an iFrame or portlet to contain and display the CA Enterprise Log Manager content.

This example uses the Liferay Portal, and assumes that you have created a portal, using Liferay installation and configuration instructions. Your own web portal may have similar controls. Consult your web portal documentation for creating iFrames or portlets.

To embed content in a Liferay Portal

1. Create a page, or open a page you want to modify in Liferay
2. Click the tools icon in the top right of the page, next to the Welcome message.
3. Select Add Application from the menu.

   The Add Application dialog appears, displaying application categories
4. Expand the Sample category, and click Add next to the iFrame application.

   A new iFrame portlet appears in the page.
5. Click the configuration link in the portlet, and add enter the API call text in the Source URL field.
6. Click Save.

   The content you selected appears in the iFrame.
7. Configure other iFrames, or publish the web portal according to Liferay documentation.

Chapter 6: API Troubleshooting

If your API calls are not working as you expect, take the following steps to troubleshoot, testing after each to see if the appropriate results appear.

1. Verify the URL call syntax:

   a. Compare your syntax to the example in the guide, verifying that you have used your own correct CA Enterprise Log Manager server name or IP address.

   b. If you have added query or report specifications, check that the main part of the call (before the specification parameters) ends with the question mark (?) character, before any parameters are added. For example:

   ```
   ?param1=val1&param2=val2
   ```

2. If the URL syntax is correct, and no data appears, verify the filters. If you are using getQueryViewer or getReport Viewer, review the filters and result condition settings in the interface. If you are using runQuery, review the parameter specifications you added to the URL:

   a. **Check Filters** - Verify that the base filters show the data you want. For example, the event source name you are filtering is entered correctly.

   b. **Syntax** - Verify that the filter syntax is correct, especially if you have constructed filters using specification parameters.

   c. **Time Filters** - Verify that your time range is broad enough, and that the timezone of your operating system is same as the timezone of CA Enterprise Log Manager.

   d. **Access Filter XML filter** - Verify that you log out of a session successfully.

   e. **LogDepot log** - Verify that events are being received, and appear in the logDepot_sponsor.log file.

3. Review the logging settings for the API component. Verify that the following files and settings are in place:

   - Property File : epSIM_logging.properties
   - Default Level is WARN
   - Logger : logmanager.ui.calmapi
   - Log File : calm.log