HP ArcSight EnterpriseView

For the Windows Operating System

Software Version: 1.0

ArcSight EnterpriseView REST API Developer Guide

Document Release Date: March 2012

Software Release Date: March 2012

Legal Notices

Warranty

The network information used in the examples in this document (including IP addresses and host names) is for illustration purposes only.

Restricted Rights Legend

Copyright Notice

Follow this link to see a complete statement of copyrights and acknowledgements for all ArcSight products: http://www.arcsight.com/copyrightnotice.

Trademark Notices

This document is confidential.

Documentation Updates

The title page of this document contains the following identifying information:

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

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

http://h20230.www2.hp.com/selfsolve/manuals

This site requires that you register for an HP Passport and sign in. To register for an HP Passport ID, go to:

http://h20229.www2.hp.com/passport-registration.html

Or click the **New users - please register** link on the HP Passport login page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details.

Support

Visit the HP Software Support Online web site at:

http://www.hp.com/go/hpsoftwaresupport

This web site provides contact information and details about the products, services, and support that HP Software offers.

HP Software online support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support web site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract. To register for an HP Passport ID, go to:

http://h20229.www2.hp.com/passport-registration.html

To find more information about access levels, go to:

http://h20230.www2.hp.com/new_access_levels.jsp

Contents

Chapter 1

REST API Overview

Using the EnterpriseView REST API, you can import assessments on assets from any Security Automation system or tool, eliminating the need to manually assess assets in EnterpriseView. Importation of assessment information on assets is only available for EnterpriseView assets of type **IP**; this assessment information is aggregated to top-level assets as in manual assessments.

Prerequisites:

Make sure that the following entities and parameters are identical in both EnterpriseView and the Security Automation tool:

- Assets of type IP
- Policy names
- Control paragraph numbers

The following HTTP request methods are used for importing assessment information into EnterpriseView**:**

- **GET** for reading one of the following resources from the EnterpriseView server:
  - **Asset ID**
  - P5 control maturity scores for **multiple** (1-5) factors
  - **Compliance score** of an asset with a control
  - **Note** on a control that is applied to an asset
- **PUT** for updating/creating the following resources on the EnterpriseView server:
  - P5 control maturity score for a **single** factor
  - P5 control maturity scores for **multiple** (1-5) factors
  - **Compliance score** of an asset with a control
- **POST** for inserting notes on a control that is applied to an asset on the EnterpriseView server.

The first step of a client application is authenticating the user, as described in "Authenticate" (on page 12). After a client is authenticated, it can work with EnterpriseView resources.

Chapter 2

HTTP Return Codes

Unless otherwise specified, these HTTP return codes are used:

Code	Cause
200	Successful operations
204	No content
400	Bad request
401	Unauthorized
403	Unauthorized operations
404	Resource not found
409	Conflict
500	Internal server error

Chapter 3

Error Messages

These are the application error messages returned by the REST API.

Code	Example	Exception
400	Trying to retrieve an asset ID by the asset's IP address, MAC address or DNS name. Parameters are not passed.	Cannot execute your request. Parameters are missing.

404	Trying to retrieve an asset ID by the asset's IP address, MAC address or DNS name.	Cannot find asset IP Address: {IP_Address}
		or
		Cannot find asset MAC Address: {MAC_Address}
		or
		Cannot find asset DNS Name: {DNS_Name}

	Trying to update or retrieve P5 score/compliance score/note for a specific control.	Cannot find policy: {Policy_Name}
		Cannot find control: {Control_Paragraph_Number}
		Cannot find asset: {Asset_ID}
409	Trying to retrieve an asset ID by the asset's IP address, MAC address or DNS name.	More than one asset matches the request. You can add additional parameters to filter the results.
500	Trying to input a compliance score that is out of range.	Score is out of range. Range should be between 0-100.
	Trying to input a P5 control maturity factor score that is out of range.	Score is out of range. Range should be between 0-5.
	Trying to input a P5 control maturity factor score or a compliance score that is not numeric or "NotAssessed".	Score value must be either numeric or "NotAssessed".
	Trying to input a P5 control maturity factor score or a compliance score that is not an integer.	Score value must be an integer.
	Trying to input a P5 control maturity factor score or a compliance score for a control that is not applied to an asset.	Control {Control_Paragraph_Number} is not applied to asset {Asset_ID}. Scores cannot be updated.
	Trying to input one or more P5 control maturity factor that are not applied to the control.	The following P5 factors are not applied to control {Control_Paragraph_Number}: {p5-1}, {p5-2}…

Chapter 4

Example Application

The following examples are based on Spring framework version 3.0.

Application Context

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:context="http://www.springframework.org/schema/context"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
       http://www.springframework.org/schema/beans/spring-beans-
3.0.xsd
       http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-3.0.xsd">

    <context:annotation-config/>

    <context:component-scan base-package="com.hp.redcat.restsample"/>

    <bean id="credentials"
class="org.apache.commons.httpclient.UsernamePasswordCredentials">
        <constructor-arg value="admin"/>
        <constructor-arg value="admin"/>
    </bean>

    <bean id="secureHttpClient" factory-bean="httpClientFactoryBean"
factory-method="getHttpClient"/>

    <bean id="httpClientFactory"
class="org.springframework.http.client.CommonsClientHttpRequestFactory">
        <constructor-arg ref="secureHttpClient"/>
    </bean>

    <bean id="restTemplate"
class="org.springframework.web.client.RestTemplate">
        <constructor-arg ref="httpClientFactory"/>
    </bean>

</beans>
```

HTTP Client Factory Bean

```
package com.hp.redcat.restsample;

import org.apache.commons.httpclient.Credentials;
import org.apache.commons.httpclient.HttpClient;
import org.apache.commons.httpclient.auth.AuthScope;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Component;

@Component
public class HttpClientFactoryBean {

    private HttpClient httpClient;

    @Autowired
    public HttpClientFactoryBean(Credentials credentials) {
        this.httpClient = new HttpClient();
        this.httpClient.getState().setCredentials(AuthScope.ANY,
credentials);
    }

    public HttpClient getHttpClient() {
        return httpClient;
    }
}
```

Example

```
package com.hp.redcat.restsample;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Component;
import org.springframework.web.client.RestTemplate;

import java.util.HashMap;
import java.util.Map;

@Component
public class RestSample {

    @Autowired
    private RestTemplate restTemplate;

/**

*  @param assetIdInEnterpriseView is the asset ID in EnterpriseView.

*   Follow the instructions in "Asset Reconciliation" (on page 13) to
get it.

*

**/

    public void putP5Assessment(String assetIdInEnterpriseView) {
        Map<String, String> variables = new HashMap<String, String>();
        variables.put("assetId", assetIdInEnterpriseView);
        variables.put("policyKey", "Cobit 4.1");
        variables.put("controlParagraph", "PO1.1");
        variables.put("p5type", "People");

        final String url = "http://hostname:8080/redcat/rest/" +

"assets/{assetId}/policy/audit/{policyKey}/{controlParagraph}/p5/{p5type}"
;
        restTemplate.put(url, "1", variables);
    }
```

Chapter 5

Authenticate

The application must perform HTTP basic authentication.

The user must have the following permissions:

- Login
- Read Policy Assessment
- Edit Policy Assessment
- Read Assets

Chapter 6

Resource References

%20 is used to represent a space in parameters such as Policy Names and control Paragraph Numbers

Asset Reconciliation

Description

Reconciliation is the process of identifying and matching entities from different data repositories, for example, in HP Server Automation (SA) and EnterpriseView. This process is designed to assure unique identification of assets in EnterpriseView.

The asset ID is passed as a parameter in all of the other methods, therefore, retrieving the asset ID is always the first action.

Parameters

The asset ID can be retrieved with one or any combination of the following parameters:

- IP Address
- MAC Address
- Host Name

URL

```
http://{host}:{port}/redcat/rest/reconcile/
asset?ipAddress={ipAddress}
```

```
http://{host}:{port}/redcat/rest/reconcile/
asset?macAddress={macAddress}
```

```
http://{host}:{port}/redcat/rest/reconcile/
asset?hostname={hostname}
```

Or any combination of parameters. For example:

```
http://{host}:{port}/redcat/rest/reconcile/
asset?hostname={hostname}&macAddress=
{macAddress}&ipAddress={ipAddress}
```

> **Note:** The IP Address, MAC Address and Host Name of an asset can change. In this case, using these parameters may return a different asset ID. If your connector has a unique key for this asset that is made up of the **Connector Asset ID** and the **Connector Name**, then it is recommended that you use these parameters to make sure that the same asset is always returned.
>
> **URL**

```
http://{host}:{port}/redcat/rest/reconcile/
asset?connectorName={connectorName}&connectorAssetId=
{connectorAssetId}&hostname={hostname}
```

```
http://{host}:{port}/redcat/rest/reconcile/
asset?connectorName={connectorName}&connectorAssetId=
{connectorAssetId}&ipAddress={ipAddress}
```

```
http://{host}:{port}/redcat/rest/reconcile/
asset?connectorName={connectorName}&connectorAssetId=
{connectorAssetId}&macAddress={macAddress}
```

Or any combination of parameters. For example:

```
http://{host}:{port}/redcat/rest/reconcile/
asset?connectorName=MyConnector&connectorAssetId=17&ipAddress=
192.168.0.1&hostname=assetHost&macAddress=assetMAC
```

Remark

EnterpriseView can return only one asset ID. If more than one asset is found with this method, then 404 error code is returned. None of the parameters (IP address, MAC address, host name) are mandatory in EnterpriseView. Using your knowledge of EnterpriseView, construct your queries to match a unique asset.

GET	
Action	Returns one asset ID.
Request Body	None
Returns	See below
Example	`GET http://127.0.0.1:8080/redcat/rest/reconcile/asset?ipAddress=192.168.0.1`

Returns:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
        <asset>
                <id>{asset ID}</id>
        </asset>
```

Compliance Score

Description

A score between 0 and 100 (integer) representing the asset compliance with a specific control.

URL

```
http://{host}:{port}/redcat/rest/assets/{asset_
id}/policy/audit/{policy_name}/{control_paragraph}/compliance
```

Remark

Attempting to create/update a compliance score that is not in the range of 0 and 100, results in a 500 exception.

All parameters are case sensitive.

GET	
Action	Returns the compliance score for a specific control in a specific policy for a specific asset. **Note:** If the compliance is not assessed, GET returns the parameter "NotAssessed" instead of a score.
Request Body	None

Returns

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
   <compliance>
      <complianceScore>{compliance Score}</complianceScore>
  </compliance>
```

Example	`GET http://127.0.0.1:8080/redcat/rest/assets/189/policy/audit/Cobit%204.1/PO1.1/compliance`

PUT	
Action	Creates/updates the compliance score for a specific control in a specific policy for a specific asset. **Note:** If the compliance that you want to create/update is not assessed, you need to pass the parameter "NotAssessed" instead of a score.
Request Body	The compliance score

Example	`PUT http://127.0.0.1:8080/redcat/rest/assets/189/policy/audit/Cobit%204.1/PO1.1/compliance`
	`(Request body) 100`

P5 Control Maturity Score Factor (Single)

Description

A score between 1-5 representing the control maturity of a single P5 factor in reference to a specific asset.

URL

```
http://{host}:{port}/redcat/rest/assets/{asset_
id}/policy/audit/{policy_name}/{control_paragraph}/p5/{p5 factor}
```

P5 factors: people, proof, procedure, process, product

Remark

All parameters are case sensitive.

PUT	
Action	Creates/updates the P5 control maturity score for one factor on a specific control in a specific policy for a specific asset. **Note:** If the P5 factor that you want to create/update is not assessed, pass the parameter "NotAssessed" instead of a score.
Request Body	The P5 control maturity score

Example	`PUT http://127.0.0.1:8080/redcat/rest/assets/189/policy/audit/Cobit%204.1/PO1.1/p5/people`
	`(Request body) 5`

P5 Control Maturity Score Factors (Multiple)

Description

A score between 1-5 representing the control maturity of one or more (maximum five) P5 factors in reference to a specific asset.

URL

```
http://{host}:{port}/redcat/rest/assets/{asset_
id}/policy/audit/{policy_name}/{control_paragraph}/p5
```

Remark

All parameters are case sensitive.

GET	
Action	Returns the P5 control maturity score for multiple factors on a specific control in a specific policy for a specific asset. If a specific P5 factor is not applied to the control, then it will not be returned. **Note:** If a P5 factor is not assessed, GET returns the parameter "NotAssessed" instead of a score.
Request Body	None
Returns	`<?xml version="1.0" encoding="UTF-8" standalone="yes"?>`

GET

```
        <p5>
                <people>{people}</people>
                <procedure>{procedure}</procedure>
                <process>{process}</process>
                <product>{product}</product>
                <proof>{proof}</proof>
        </p5>
```

Example

```
GET http://127.0.0.1:8080/redcat/rest/assets/189/policy/
audit/Cobit%204.1/PO1.1/p5/
```

PUT

Action	Creates/updates the P5 control maturity score for multiple factors on a specific control in a specific policy for a specific asset. **Note:** • If the P5 factor that you want to create/update is not assessed, you need to pass the parameter "NotAssessed" instead of a score. • If you try to create/update a P5 factor that is not applied to the control, you get a 500 error code.
Request Header	`content-type: application/xml`

Request Body

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
        <p5>
                <people>{people}</people>
                <procedure>{procedure}</procedure>
                <process>{process}</process>
                <product>{product}</product>
                <proof>{proof}</proof>
        </p5>
```

Example

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
        <p5>
                <people>3</people>
                <procedure>3</procedure>
```

PUT	

```
                <process>4</process>

                <product>5</product>

                <proof>NotAssessed</proof>

        </p5>
```

Notes

Description

Any number of notes can be attached to a control in a specific policy for a specific asset.

URL

```
http://{host}:{port}/redcat/rest/assets/{asset_id}/policy/audit/{policy_name}/{control_paragraph}/notes
```

Remark

POST creates a single note. GET returns all notes for the control.

GET	
Action	Returns all of the notes for a specific control in a specific policy for a specific asset.
Request Body	None
Returns	

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

        <notes>

                <note>

                   <createTime>{creation date and time}</createTime>

                   <creator>

                        <uniqueId>{user name}</uniqueId>

                   </creator>

                   <message>{note text}</message>

                </note>

                …

        </notes>
```

Example	`GET http://127.0.0.1:8080/rest/assets/189/policy/audit/Cobit%204.1/PO1.1/notes`

POST	
Action	Create a new note for a specific control in a specific policy for a specific asset.
Request Body	The note text
Returns	HTTP code 200 or an error code

Example	`http://127.0.0.1:8080/redcat/rest/assets/189/policy/audit/Cobit%204.1/PO1.1/note`
	`{note text}`