Primavera Portfolio Management

API Addendum

G48459-01
July 2026

Primavera Portfolio Management API Addendum,

G48459-01

Primary Author: Oracle Corporation

Contents

1

API Addendum

Use this document in conjunction with the API Reference Guide (api.chm). The Primavera Portfolio Management API.chm is no longer being updated. Changes, additions, and corrections to the API.chm are documented here.

2
WSDL

WSEventSink

> **Note**
>
> This topic is updated to replace the sample code found in the api.chm.

A Web Service Event Sink must be implemented in order to receive the notifications when the Programmatic Alerts are triggered.

The Web Service Sink is a web service that implements .

It should publish the following methods:

- actionItemModified
- categoryConditionMet
- deliverableModified
- phaseModified
- multipleCategoryConditionMet
- itemAddedToPortfolio

Each one of the psPortfoliosWebServiceAlert methods accepts a parameter (webServiceURL) that holds the URL where the Web Service Event Sink is hosted.

The system will process events one by one, and call the registered Event Sink URLs with the relevant method which correlates to the event being triggered.

The easiest way to implement the Web Service Sink is to start with the fix code examples WsEventSinkSample. (for Microsoft Visual Studio 2015).

For Example: AddItemAddedAlert was called to add a Programmatic Alert on portfolio “Customers” and with webServiceURL set to "http://MyServer/ProSightEventSink/EventSink.asmx".

Once the alert is added, whenever an item is added to the “Customers” portfolio, the System's Alerts Engine will call the Web Service method itemAddedToPortfolio at the above mentioned URL, sending it all the event details.

Implementation Best Practices:

1. When implementing the Web Service Event Sink, It is recommended that the methods that implement the WsEventSinkSample return in a very short length of time (less than a second). This is to make sure that the hosting application is always responsive and that the PPM Alerts Engine processing is not delayed. If there is a need for longer processing, it is

advised to use some sort of static queue to hold all alerts dispatched and to process them using a separate thread.

2. Remember to delete Programmatic Alerts which are not in use, or use the ExpirationTime property to set the Alert to expire after a certain amount of time.
3. In some special cases, Alert notifications can be sent more than once. Protect the Event Sink code by “remembering” alerts that were previously received, using the AlertGuid and EventDateTime properties.

WSEventSinkSample

> **Note**
>
> - This topic is updated to replace the sample code found in the api.chm.
> - When upgrading the old sample code application (V18 and its prior versions) to V19 version or later, modify the WebService namespace attribute for the `Service.cs` class from [WebService(Namespace = "http://prosight.com/wsdl/7.0/EventSink/")] to [WebService(Namespace = "http://prosight.com/wsdl/8.0/EventSink/")].

The easiest way to implement the the Web Service Sink is to start with the sample code (for Microsoft Visual Studio 2015).

This sample application writes the information of each message received to a log file.

The location of the log file is configurable via the Web.Config file (configuration / system.diagnostics / trace / listeners / add). Note that the Windows user running the service (by default NETWORK SERVICE) needs permission to write to the directory you select.

There are two samples available, depending on the selection of sWebServiceNameSpace (this selection affects how CategoryChangedAlert works):

WsEventSinkSample for sWebServiceNameSpace = http://prosight.com/wsdl/7.0/EventSink

WsEventSinkSample for sWebServiceNameSpace = http://prosight.com/wsdl/8.0/EventSink

3

Main Objects

psPortfoliosvalueList

psPortfoliosValueList Class

APIs for handling the definition of Value Lists.

Methods

GetValues Method

Get Value List Values. May raise exception.

Return Type

A variant value containing an ARRAY of VARIANT, each containing a COM object of type psPortfoliosValueListValueInfo.

> **Note**
>
> The response of GetValues method will contain both ‘Weight’ and ‘WeightDouble’ elements in the object of type “psPortfoliosValueListValueInfo”.
>
> - If the double(up to 10-digit precision) value is saved, ‘Weight’ element will return rounded float type value and ‘WeightDouble’ element will return double type value. For example, for a value of 32.1234567890, the value of the ‘Weight’ element will be 32.123455 when rounded as a float, and the ‘WeightDouble’ element will be 32.1234567890 when rounded as a double type value.
> - If the saved value falls in the range of a float number, then the value of both “Weight” and “WeightDouble” will be same. For example, for a value of 32.12345, the value of Weight and WeightDouble is 32.12345.

Syntax

```
object.GetValues(sName)
```

`object` is an expression evaluating to an object of type psPortfoliosvalueList.

Arguments

Argument	Data Type	Description
sName	String	Name of the ValueList.

Return Value

For the Return Error Code refer to *Error Handling for APIs of type A*.

Code Examples in Different Programming Languages

See example of psPortfoliosValueList.GetValue

Update Values Method

Updates List of values in a Value List.

Return Type

A psRETURN_VALUES value.

Syntax

```
object.UpdateValues(sName, vaValues)
```

`object` is an expression evaluating to an object of type psPortfoliosvalueList.

Arguments

Argument	Data Type	Description
sName	String	The Value List to update is identified by this.

Argument	Data Type	Description
vaValues	Variant	List of objects of type psPortfoliosValueListValueInfo. Each object in the list represents a value in the Value List. Some values will get updated (possibly renamed or assigned a new weight), while others will be added. Values that exist in the value list but not specifed in vaValues will be deleted. The relevant fields in each psPortfoliosValueListValueInfo object are: • ID An integer used as the internal representation of the value in the PPM database (different than the number shown in the Value List wizard). If specified, the value to update will be identified by this. If no Value with this ID exists in the value list, such a value will be added. To rename a value, specify the ID. • Text Unless ID is specified, this string identifies the value to update. If no Value with this Text exists in the value list, such a value will be added. If ID is specified, this sets the Text that is displayed for the Value. • Weight This is the Weight as in the Value List wizard. If it is not specified, weights will be automatically assigned to the values using consecutive numbers in the order that the values appear in the list. Use this element to send a zero (0) value The data type of the Weight element is Float. • WeightDouble This is the Weight as in the Value List wizard. Use this element instead of the Weight element to specify a non-zero value with a precision of up to 10 digits. When an API request contains both “Weight” and “WeightDouble” elements, the value of the WeightDouble element supersedes and is saved in the database.

Argument	Data Type	Description
		Note The data type of the WeightDouble element is Double.

Return Value

For the Return Error Code refer to *Error Handling for APIs of type A*.

Code Examples in Different Programming Languages

Set the 'Domains' value list to show only Items(1; 1.7), Portfolios, Gizmos(3;) and Wizards (; 3.14). Gizmos(3) will replace Applications(3).

See Example 2 of psPortfoliosValueList.GetValue

psPortfoliosUser

psPortfoliosUser Class

APIs for handling users.

Methods

GetUserInfo Method

Get details of user identified by ID.

Available in:

COM: Yes Web Services (SOAP Document/Literal): Yes SOAP RPC: No

Return Type

A psPortfolioUserInfo object value.

Syntax

```
object.GetUserInfo(sID As Long)
```

(object is an expression evaluating to an object of type psPortfoliosUser)

Arguments

Argument	Data Type	Description
sId	Long	The ID of the user.

Return Value

Object of type psPortfolioUserInfo object. The object represents a User in Portfolios

GetUserInfoByLogin Method

Returns information about a User in the system.

Available in:

COM: Yes Web Services (SOAP Document/Literal): No SOAP RPC: No

Return Type

A psPortfoliosUserInfo value.

Syntax

```
object.GetUserInfoByLogin(sLogin As BSTR)
```

(`object` is an expression evaluating to an object of type psPortfoliosUser.)

Arguments

Argument	Data Type	Description
sLogin	BSTR	

Return Value

Object of type psPortfolioUserInfo object value.

4

Helper Objects

psPortfoliosValueListValueInfo

psPortfoliosValueListValue Class

psPortfoliosValueListValueInfo object contains the following fields:

Field	Data type	Description
ID	Long	An integer used as the internal representation of the value in the PPM database (different from the number shown in the Value List wizard).
Text	String	Name of the ValueList.
Weight	Float	This is the Weight as in the Value List wizard. A Float data type accepts a precision up to 7 decimal places.
WeightDouble	Double	This is the Weight as in the Value List wizard. A Double data type accepts a precision up to 10 decimal places.

Code Examples in Different Programming Languages

See psPortfoliosValueList.UpdateValues and psPortfoliosValueList.GetValues

Methods

In addition to the methods described in the API Reference Guide, the following methods have been added.

getWeightDouble Method

Get Value List Value Weight

Return Type

A double value.

Syntax

```
object.getWeightDouble
```

object is an expression evaluating to an object of type psPortfoliosValueListValueInfo

Arguments

None

Return Value

For the Return Error Code refer to *Error Handling for APIs of type A*.

setWeightDouble method

Set Value List Value Weight

Syntax

object.setWeightDouble(sText)

object is an expression evaluating to an object of type psPortfoliosValueListValueInfo

Arguments

Argument	Data Type	Description
sText	Double	

Return Value

For the Return Error Code refer to *Error Handling for APIs of type A*.

psPortfolioUserInfo object

psPortfoliosUserInfoClass

psPortfoliosUserInfo object contains the following fields:

Field	Data type	Description
ID	Long	Internal ID. Can be used in conjunction with psPortfoliosCell operations for categories of type User.
FirstName		User's first name.
LastName		User's last name.
JobTitle		User's job title.
Company		User's company name.
Department		User's department within the company.
Phone		User's phone number.
Fax		USer's fax number.
Address		User's postal address.
EMail		User's email address.
IsContact		True if this is a contact. If true, the user cannot log in to the system.

Login		Login of User. Serves as a (case-insensitive) unique identifier within an installation.
LicenseType		License Type assigned to the user. This is relevant only if `IsContact`=false.
AlertImportanceFilter		Value from psEMAIL_IMPORTANCE. Accept alerts with an importance of at least this value.
ReceiveAlerts		If false, no alert emails will be sent to user/contact.
IsEnabled		True if the user is enabled in the system, false if not.