ORACLE®

PRIMAVERA

Integration Interface Guide
Release 15.1

April 2015

Contents

Introduction

The interface to integrate Primavera Unifier and an external system is based on Web Service methodology. This document describes the methods required for the integration and the data being passed in those methods.

Note: The following services are included in the WSDL file, but these services are reserved for internal use (not for customer use): createFundingStructure, createOIMUser, updateOIMUser, createSapBPRecord, getSapBPList, getSapBPRecord, updateSapBPRecord, Ping, getTransactionStatus.

Security

Primavera Unifier Web services are sent and received through a secured server using "https" protocol.
Once https is set up at both ends, the client must accept the server certificate to continue with the Integration procedure.

Primavera Unifier also requires clients to register on Primavera Unifier with the shortname (the company identifier) and authcode (the company authentication code).

Unifier Web Services

Tag Requirements

- All input XML tags must be wrapped within the tags <list_wrapper> </list_wrapper>
- All BP Tag names must start with <_bp> and end with </_bp>
- All BP Line Items must start with <_bp_lineitems> and end with </_bp_lineitems>

Terminology

- Project: This term refers to Projects (Standard), which are projects that are supported by Primavera Unifier without going through the Shell Manager.
- Shells: Shells are designed in uDesigner and can represent virtually anything you want. For example, a shell can represent a building, a project, or a concept such as "innovative ideas." Refer to Primavera Unifier   Admin and User Guide for more information.
- Shell with Cost Code WBS: These are shell types for use with the WBS Cost Manager.
- Shell with Cost Code Generic: These are shell types with for use with the Generic Cost Manager.

Internationalization and Web Services

About Web Services

New records can be created and line items added using Integration through Web Services. Also, the Unifier Schedule Manager integrates with Primavera scheduling software by way of Web Services.

> **Note**: Integration through Web Services must be coordinated with an Oracle Primavera representative.

As Project Administrator, you can receive email notification of the successful creation of a shell instance, for shells that are created manually, through Web Services or a CSV file upload, or through auto-creation. This notification can be set up in email notifications in uDesigner. Also, you can set your User Preferences to control whether you receive these notifications.

Refer to the *Primavera Unifier Integration Interface Guide* for data elements you can use with Web Services.

Web Services and Internationalization

The Web Services support Internationalization (language translation) according to the User's Preference (language).

Since Web Services allows interaction between two systems (machine to machine), the data is transferred from one system to an external system.

Note: If a record (Example: Business Process) is created by using Web Services and the Data Definition (DD) label includes a non-ASCII string, then the record creation will fail.

Number formatting of data

When you enter numeric data in XML, you can use the decimal point (period) and negative sign (dash), only.

Examples

XML Tag: <Committed_Amount>100.99</Committed_Amount>

XML Tag: <Credited_Amount>-1423.99</Credited_Amount>

Get Web Services

You can use the Get Web Services call methods to get various attributes of Shell, WBS, and the list of Business Process records, Shells, and User defined data.

When a Get call is executed, all the input content in the response XML will be in the language of the source strings.

Number formatting does not apply to the numeric data and the decimal point is a period. The negative numbers are displayed with the minus sign before the numeric data, for example, -12345.99.

Note: Number formatting is not supported for Symbols that are based on a right-to-left language such as official languages of Afghanistan or Hebrew.

Methods

WBS Code Methods

Get WBS Structure

Description

This method gets WBS Structure in Primavera Unifier for a company.

Support

This service is available at Company Level

Level	Yes or No
Company Level	Yes
Project and Shell Level	No

Prototype

public XMLObject getWBSStructure(String shortname, String authcode);

Parameters

Parameter	Description
Shortname	identifier of the company, company's short name
Authcode	authentication key for the company, in text string

Return Value

XMLObject (XMLObject is described in Appendix D of this document)

Sample Method

getWBSStructure ("acme", "acme_authcode")

Sample Return XML

```
<ulink>
        <_shortname>acme</_shortname>
        <_authcode>acme_authcode</_authcode>
        <_servicename>createWBS</_servicename>
        <_projectNumber>proj001</_projectNumber>
<List_wrapper>
        <_bp_wbscode>
                <wbs_code>0-0000~~0-08600</wbs_code>
                <wbs_item>New WBS Code 1</wbs_item>
                                <description>New WBS Code 1</description>
                <costattribute></costattribute>
                <external_refid></external_refid>
                <owner></owner>
                <status>Active</status>
                <cost_type>Expense</cost_type>
    </_bp_wbscode>
    <_bp_wbscode>
                <wbs_code>0-0000~~0-08700</wbs_code>
                <wbs_item>New WBS Code 2</wbs_item>
                               <description>New WBS Code 2</description>
                <costattribute></costattribute>
                <external_refid></external_refid>
                <owner></owner>
                <status>Active</status>
                <cost_type>Expense</cost_type>
    </_bp_wbscode>
</List_wrapper>
</ulink>
```

Create WBS Code

Description

This method creates WBS code under Project or Shell (cost code type as WBS) cost sheet.

Support

This method only supports creation of WBS code for ASP and Self host type installation.

Installation: ASP and Self host

Level	Yes or No
Company Level	No
Project Level and Shell Level	Yes

Prototype

public XMLObject createWBS(String shortname,String authcode, string projectNumber, String WBSXML);

Parameters

Parameter	Description
shortname	identifier of the company, company's short name
authcode	authentication key for the company, in text string
projectNumber	identifier of the project or a shell in Primavera Unifier
WBSXML	identifier used for WBS creation XML format

WBSXML Elements

The following is a list of elements that are supported while creating WBS code. Any other codes that are defined as part of Cost Attribute form will be supported except User and BP Picker

Parameter	Description
<status>	Status is not a required tag when you are sending createWBS message. But if you include this tag then following are the valid values Active: Code will be created with Active status Inactive: Code will be created with Inactive status If you do not send this tag then code will be created without any status which is equivalent to Inactive status.
<cost_type>	This is not a required tag when you are sending createWBS message. But if you include this tag then following are the valid values Capital: To create WBS of type Capital Expense: To create WBS of type Expense If you do not send this tag then code will be created with Capital as Cost Type.
<costattribute>	This is not a required tag when you are sending createWBS message. But if you include this tag then send data that is defined as values under Cost Attribute data definition.

Return Value

XMLObject (XMLObject is described in Appendix D of this document)

Sample XML Data

Creates two WBS codes

```
<ulink>
        <_shortname>acme</_shortname>
        <_authcode>acme_authcode</_authcode>
        <_servicename>createWBS</_servicename>
        <_projectNumber>proj001</_projectNumber>
<List_wrapper>
        <_bp_wbscode>
                <wbs_code>0-08600</wbs_code>
                <wbs_item>New WBS Code 1</wbs_item>
                               <description>New WBS Code 1</description>
                <costattribute></costattribute>
                <external_refid></external_refid>
                <owner></owner>
                <status>Active</status>
                <cost_type>Expense</cost_type>
    </_bp_wbscode>
    <_bp_wbscode>
                <wbs_code>00-08700</wbs_code>
                <wbs_item>New WBS Code 2</wbs_item>
                              <description>New WBS Code 2</description>
                <costattribute></costattribute>
                <external_refid></external_refid>
                <owner></owner>
                <status>Active</status>
                <cost_type>Expense</cost_type>
    </_bp_wbscode>
</List_wrapper>
</ulink>
```

Creates WBS code with a tree structure

```
<ulink>
        <_shortname>acme</_shortname>
        <_authcode>acme_authcode</_authcode>
        <_servicename>createWBS</_servicename>
        <_projectNumber>proj001</_projectNumber>
<List_wrapper>
        <_bp_wbscode>
                <wbs_code>0-0000~~0-08600</wbs_code>
                <wbs_item>New WBS Code 1</wbs_item>
                        <description>New WBS Code 1</description>
                <costattribute></costattribute>
                <external_refid></external_refid>
                <owner></owner>
                <status>Active</status>
                <cost_type>Expense</cost_type>
    </_bp_wbscode>
    <_bp_wbscode>
                <wbs_code>0-0000~~0-08700</wbs_code>
                <wbs_item>New WBS Code 2</wbs_item>
                       <description>New WBS Code 2</description>
                <costattribute></costattribute>
                <external_refid></external_refid>
                <owner></owner>
                <status>Active</status>
                <cost_type>Expense</cost_type>
    </_bp_wbscode>
</List_wrapper>
</ulink>
```

Additional Information

Use getWBSStructure service to get the WBS attribute structure. The retrieved structure can be used to create new WBS codes. User can define a Cost Attribute form in uDesigner. In order to send XML message to create new Cost codes user has to send data using schema that is generated as output from getWBSStrucutre service.

If the Project or Shell WBS structure is in tree mode, or a parent / child relationship is required, the WBS code passed via the integration interface must concatenate the parent wbs code and child wbs code with "~~" as the delimiter.

If you have multilevel parent child relationship then you have to pass in "~~" as delimiter at all levels to separate parent and child.

Parent will be automatically created when you create child.

WBS Item element will take value of Description element.

This service will only work for Shells with Cost Code: WBS

Get WBS Codes

Description

This method gets the WBS Codes in Primavera Unifier for a particular project or shell.

Support

This service is available at Company Level

Level	Yes or No
Company Level	No
Project and Shell Level	Yes

Prototype

public XMLObject getWBSCodes(String shortname, String authcode, String projectnumber, String options);

Parameters

Parameter	Description
shortname	identifier of the company, company's short name
authcode	authentication key for the company, in text string
projectnumber	Identifier of the project or shell in Primavera Unifier.
options	Three options are available: <status>: can be set to active or inactive. If status is blank or not provided it will return all status. <cost_type>: can be set to capital or expense. If cost_type is blank or not provided it will return all cost types. <summary_detail> : can be set to true or false. Setting to true, leaving blank or not providing the value will return all summaries and detail WBS codes. Setting to false will return on the detail WBS codes with no summaries.

Return Value

XMLObject (XMLObject is described in Appendix D of this document)

Sample Method

getWBSCodes ("acme", "acme_authcode", "P-00001" status=active)

Sample XML Input

```
<ulink>
            <_shortname>acme</_shortname>
            <_authcode>acme_authcode</_authcode>
            <_servicename> getWBSCodes </_servicename>
                        <_projectnumber>P-00001</_projectnumber>
            <options>
                                    <status>active</status>
                         <cost_type> </cost_type >
                                    <summary_detail></summary_detail >
                        </options>
</ulink>
```

Sample Return XML

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<List_Wrapper>
                        <codestructure>
                                    <treemode>true</treemode>
                                    <segsize>1</segsize>
                                    <seperator>-</seperator>
                        </codestructure>
            <costcodes>
                        <costcode>
                            <wbscode>Wbs1</wbscode>
                            <wbsitem>02-12-2012 17:00:00</wbsitem>
                            <costtype>capital</costtype>
                            <status>Active</status>
                            <wbscodeid>2</wbscodeid>
                            <parentid>1</parentid>
                            <indentlevel>1</indentlevel>
                            <orderid>0</orderid>
                        </costcode>
                        <costcode>
                            <wbscode>Wbs2</wbscode>
                            <wbsitem>02-12-2012 17:00:00</wbsitem>
                            <costtype>expense</costtype>
                            <status>Active</status>
                            <wbscodeid>3</wbscodeid>
                            <parentid>1</parentid>
                            <indentlevel>1</indentlevel>
                            <orderid>1</orderid>
                        </costcode>
                        <costcode>
                            <wbscode>Wbs2.1</wbscode>
                            <wbsitem>02-12-2012 17:00:00</wbsitem>
                            <costtype>expense</costtype>
                            <status>Active</status>
                            <wbscodeid>4</wbscodeid>
                            <parentid>3</parentid>
                            <indentlevel>2</indentlevel>
                            <orderid>2</orderid>
                        </costcode>
                        <costcode>
                            <wbscode>Wbs3</wbscode>
                            <wbsitem>02-12-2012 17:00:00</wbsitem>
                            <costtype>capital</costtype>
                            <status>Active</status>
                            <wbscodeid>5</wbscodeid>
                            <parentid>1</parentid>
                            <indentlevel>1</indentlevel>
                            <orderid>3</orderid>
                        </costcode>
            </costcodes>
</List_Wrapper>
```

Business Process (BP) Methods

Create BP Record

Description

This method can be used to create new business process records in Primavera Unifier based on Integration design that has been defined in uDesigner. For more information refer to *uDesigner Administrator Guide*.

If you want to create a BP record with line item information, then you should use this service. If this is a cost related BP, line items will be rolled up to the cost sheet.

Support

This process supports all type of BPs

Installation: ASP and Self host

Level	Yes or No
Company Level	Yes
Project and Shell Level	Yes

Prototype

public XMLObject createBPRecord(String shortname, String authcode, string projectNumber, String BPName, String BPXML);

Parameters

Parameter	Description
shortname	identifier of the company, company's short name
authcode	authentication key for the company, in text string
projectNumber	Identifier of the project or shell in Primavera Unifier. If you want to create a BP record at Company level then pass null for this parameter.
BPName	Identifier of the business process in Primavera Unifier, for example: Invoice. This is case sensitive, and should be the name as given in uDesigner
BPXML	identifier for BP content in XML format

BPXML Elements

The elements that include as part of BPXML parameters is dependent on the Integration interface design of your BP in uDesigner. XML format can be exported (downloaded) from uDesigner. Elements with direction marked as "Input" or "Both" will be part of XML format.

Note that elements lists should be send based on the type of BP and your design in uDesigner.

Following a list of elements that can be included as part of integration design in uDesigner and behavior when you include them. Note that elements lists should be send based on the type of BP for which you are creating a record.

Tag Name	BP Type	Description
<amount>	Cost	Amount is a required field if you are sending line item information under _bp_lineitems. You will see same value on Primavera Unifier.
<wbs_code>	Cost	WBS Code is a required field if you are sending line information under _bp_lineitems.
<status>	Any	a) If BP uses a status field then you can send status tag. b) If the BP is a workflow based then the record will be created and will stay in the initial status that is valid. User has to go through remaining workflow process. c) If it is non-workflow based then the message can send a valid status and the record will take that status. If there is only one status defined for the BP then by default that status will be picked even if user sends a wrong status value.
<record_no>	Any	You can send record number by using this tag. If you do not specify a record_no tag as part of message then a new record will be created. Primavera Unifier will assign a record number. If you specify a record_no then system will create a record with the given record number. If the record number is a duplicate number then an error message will be send back as response. If the record_no is specified and if it does not exists then a new record will be created.

Return Value

XMLObject or XMLFileObject (XMLObject and XMLFileObject is described in Appendix D of this document). This is dependent on which WSDL you are going to use.

If you use WSDL –1 then you should expect XMLObject as return value.

If you use WSDL – 2 then you should expect XMLFileObject as return value.

Sample XML Data

Sample XML Data for Cost type business process

```
<ulink>
        <_shortname>acme</_shortname>
        <_authcode>acme_authcode</_authcode>
        <_servicename>createBPRecord</_servicename>
<_projectNumber>Proj-001</_projectnumber>
<_BPName>Bpname</_BPName>
        <_projectNumber>proj001</_projectNumber>
<list_wrapper>
   <_bp>
<title>New title-1</title>
<vendorName>OFFICEMAX INCORPORATED</vendorName>
<recvDate>2002-07-09</recvDate>
<_bp_lineitems>
   <wbs_code>010_PRE_PROJ~~OVRHD-PERMT</wbs_code>
   <short_desc>E22962GR DRAFTING TABLE BASE</short_desc>
   <amount>120000</amount>
</_bp_lineitems >
<_bp_lineitems >
   <wbs_code>010_PRE_PROJ~~OVRHD-PERMT</wbs_code>
   <short_desc>E22962GR DRAFTING TABLE BASE</short_desc>
   <amount>120000</amount>
</_bp_lineitems>
   </_bp >
</list_wrapper>
</ulink>
```

Example of Cost BP with Fund Information

```
<ulink>
        <_shortname>acme</_shortname>
        <_authcode>acme_authcode</_authcode>
        <_servicename>createBPRecord</_servicename>
        <_projectNumber>PROJ-0001</_projectNumber>
        <_BPName>bpname</_BPName>
        <_recordNumber></_recordNumber>
<List_Wrapper>
<_bp>
        <amount>6000</amount>
        <title>Integration Testing Funds</title>
        <Fund_Total_Amount>6000</Fund_Total_Amount>
        <status>Approved</status>
           <_bp_lineitems>
                <fund_code>Federal-Bond-Infrastructure Bond-2006</fund_code>
                <amount>6000</amount>
                <short_desc>Integration Testing</short_desc>
          </_bp_lineitems>
</_bp>
</List_Wrapper>
</ulink>
```

Example of General Spends Business Process (SOV = Group by commit codes)

```
<ulink>
        <_shortname>acme</_shortname>
        <_authcode>acme_authcode</_authcode>
        <_servicename>createBPRecord</_servicename>
        <_projectNumber>PROJ-0001</_projectNumber>
        <_BPName>bpname</_BPName>
        <_recordNumber></_recordNumber>
<List_Wrapper>
        <_bp>
           <record_no></record_no>
           <amount>100.00</amount>
           <title>Spends 1</title>
           <refid>PO-GS-0003</refid>
           <creator_id>Pixar Admin</creator_id>
                <_bp_lineitems>
                   <bItemID>1-1000~~1-1125</bItemID>
                   <amount>100.00</amount>
                   <uuu_effective_date>03-20-2012 15:00:00</uuu_effective_date>
                   <short_desc>Line 1 Payment</short_desc>
                   <_breakdown></_breakdown>
                </_bp_lineitems>
        </_bp>
</List_Wrapper>
</ulink>
```

Example of Payment Application Business Process (SOV = Individual Line Items)

```
<ulink>
        <_shortname>acme</_shortname>
        <_authcode>acme_authcode</_authcode>
        <_servicename>createBPRecord</_servicename>
        <_projectNumber>PROJ-0001</_projectNumber>
        <_BPName>bpname</_BPName>
        <_recordNumber></_recordNumber>
<List_Wrapper>
        <_bp>
           <record_no></record_no>
           <amount>100.00</amount>
           <title>Spends 1</title>
           <refid>PO-GS-0003</refid>
           <creator_id>Pixar Admin</creator_id>
                <_bp_lineitems>
                   <uuu_tab_id>Line Item List</uuu_tab_id>
                   <bItemID>1-1000~~1-1001</bItemID>
                   <_refnum>1</_refnum>
                   <amount>100.00</amount>
                   <uuu_effective_date>03-20-2012 15:00:00</uuu_effective_date>
                   <short_desc>Line 1 Payment</short_desc>
                </_bp_lineitems>
        </_bp>
</List_Wrapper>
</ulink>
```

Additional Information

To get the BP XML structure, first send a message to get BP record. Refer to the Web Service getBPRecord for more information.

This interface can be extended with additional data elements. In order for Primavera Unifier to recognize those elements, the data element name must match those created in uDesigner for this business process.

YTB/AFC: If the tags YTB or AFC are present in the line items, Primavera Unifier processes these tags. If they are not present, Primavera Unifier will check the uDesigner BP setup configuration for Cost Adjustment. If this field has been set, the line item amount is treated as YTB/AFC and processed.

Single Record BP:

Record Exists: If you send a message for single record BP and a record already exists then Primavera Unifier will not send back and error but if you send an invalid record number then Primavera Unifier will send back an error.

Record does not exist: If you send a message for single record BP and a record does not exist then Primavera Unifier will create a record for you. If the BP type is of Line item then any line items included in the message will be added to the BP record. If the BP type is of Simple then any line items added as part of message will be ignored.

If your BP has **uuu_creation_date** then it will be populated when you send a message

If your Cost BP has fund related information then you have to send Fund code details. If you are using multi segment fund code then send the code segments with delimiters.

Ex: Fundseg1-Fundseg2-Fundseg3.

The fund information send as part of Cost BP will be rolled up to Funding Sheet or Cost Sheet.

You cannot send fund related information for Cost Type BPs such as General Spends and Payment Applications.

All data element values provided as part of integration message will be validated against the action form selected as part of design in uDesigner. Validation includes required field, form validation etc..,

The following XML tag <_refnum> </refnum> should be added under BP line item tag <_bp_lineitem> </bp_lineitem> to support the ability to import the line items for General Spends and Payment Applications if SOV is individual line items. For Group by commit codes lines are identified by WBS Code.

If the action form used for validation is auto-populating values from a Business Process picker then system will auto-populate these values if a valid value is provided for that Business Process Picker.

This service can be used for both Project (Standard) and Shells of cost code type WBS and Generic.

Create Complete BP Record

Description

This method can be used to create new business process records with attachments in Primavera Unifier based on Integration design that has been defined in uDesigner. It allows attachments to be added to the upper form and line items.

If you want to create a BP record with line item information with attachments, then you should use this service. If this is a cost related BP, line items will be rolled up to the cost sheet.

Support

This process supports all type of BPs

Installation: ASP and Self host

Level	Yes or No
Company Level	Yes
Project and Shell Level	Yes

Prototype

public XMLObject createCompleteBPRecord(String shortname, String authcode, string projectNumber, String BPName, String BPXML, String iszipfile, FileObject[] fileobjects);

Parameters

Parameter	Description
shortname	identifier of the company, company’s short name
authcode	authentication key for the company, in text string
projectNumber	Identifier of the project or shell in Primavera Unifier. If you want to create a BP record at Company level then pass null for this parameter.
BPName	Identifier of the business process in Primavera Unifier, for example: Invoice. This is case sensitive, and should be the name as given in uDesigner
BPXML	identifier for BP content in XML format
fileobjects	Array of FileObjects, each FileObject is made up of two components: String filename (this should be the same name as in the record data <_attachment><filename> tag, and javax.activation.DataHandler as filestream. Files can be sent as a single zipped file or individual files.
isZipFile	Identifies if the file provided under the fileobjects tag is a zip file. The value for this tag is "yes" or "no". Yes to allow zip file as an attachment. Otherwise, select No. **Note:** Only .zip file format is acceptable. When using .zip file as an attachment, you should attach one zip file at a time that includes a single attachment of each file in the zip file.

BPXML Elements

The elements that include as part of BPXML parameters is dependent on the Integration interface design of your BP in uDesigner. XML format can be exported (downloaded) from uDesigner. Elements with direction marked as "Input" or "Both" will be part of XML format.

Note that elements lists should be send based on the type of BP and your design in uDesigner.

Following a list of elements that can be included as part of integration design in uDesigner and behavior when you include them. Note that elements lists should be sent based on the type of BP for which you are creating a record.

Tag Name	BP Type	Description
<amount>	Cost	Amount is a required field if you are sending line item information under _bp_lineitems. You will see same value on Primavera Unifier.

Tag Name	BP Type	Description
<status>	Any	a) If BP uses a status field then you can send status tag. b) If the BP is a workflow based then the record will be created and will stay in the initial status that is valid. User has to go through remaining workflow process. c) If it is non-workflow based then the message can send a valid status and the record will take that status. If there is only one status defined for the BP then by default that status will be picked even if user sends a wrong status value.
<record_no>	Any	You can send record number by using this tag. If you do not specify a record_no tag as part of message then a new record will be created. Primavera Unifier will assign a record number. If you specify a record_no then system will create a record with the given record number. If the record number is a duplicate number then an error message will be send back as response. If the record_no is specified and if it does not exists then a new record will be created.
attachment	Simple BP, Text BP, Line Item BP, Cost BP, Document Type BPs	Multiple attachment elements can be added to the upper form and to line items (if applicable). The attachment element contains sub elements that identify: File name Title Revision Number Issue Date Version Reference Refer to the Sample XML below.

Return Value

XMLObject or XMLFileObject (XMLObject and XMLFileObject is described in Appendix D of this document). This is dependent on which WSDL you are going to use. If you use WSDL –1 then you should expect XMLObject as return value and if you use WSDL – 2 then you should expect XMLFileObject as return value.

Sample BP XML

```
<ulink>
        <_shortname>acme</_shortname>
        <_authcode>acme_authcode</_authcode>
        <_servicename>createCompleteBPRecord</_servicename>
<_BPName>bp name</_BPName>
        <_projectNumber>proj001</_projectNumber>
<List_wrapper>
 <_bp>
        <record_no> </record_no>
        <title>title</title>
        <status>status</status>
        <amount>amount</amount>
        <_attachment>
                <file_name>Specs_PDF_1.pdf</file_name>
                <title>Specifications</title>
                <revision_no></revision_no>
                <issue_date></issue_date>
                <file_size></file_size>
                <version></version>
                <reference>No </reference>
        </_attachment>
        <_bp_lineitems>
                <amount>22.0</amount>
                <short_desc>line-2</short_desc>
                <uom>day</uom>
                <_attachment>
                        <file_name>Specs_PDF_1.pdf</file_name>
                        <title>Specifications</title>
                        <revision_no></revision_no>
                        <issue_date></issue_date>
                        <file_size></file_size>
                        <version></version>
                        <reference>No </reference>
                </_attachment>
        </_bp_lineitems>
</_bp>
</List_Wrapper>
</ulink>
```

Additional Information

Only .zip file format is acceptable. When using a Zip file as an attachment, you should attach one zip file at a time that includes the single attachment for each file in the zip file. To get the BP XML structure, first send a message to get BP record. Refer to the Web Service getBPRecord for more information.

This interface can be extended with additional data elements. In order for Primavera Unifier to recognize those elements, the data element name must match those created in uDesigner for this business process.

YTB/AFC: If the tags YTB or AFC are present in the line items, Primavera Unifier processes these tags. If they are not present, Primavera Unifier will check the uDesigner BP setup configuration for Cost Adjustment. If this field has been set, the line item amount is treated as YTB/AFC and processed.

Single Record BP:

Record Exists: If you send a message for single record BP and a record already exists then Primavera Unifier will not send back and error but if you send an invalid record number then Primavera Unifier will send back an error.

Record does not exist: If you send a message for single record BP and a record does not exist then Primavera Unifier will create a record for you. If the BP type is of Line item then any line items included in the message will be added to the BP record. If the BP type is of Simple then any line items added as part of message will be ignored.

If your BP has **uuu_creation_date** then it will be populated when you send a message.

If your Cost BP has fund related information then you have to send Fund code details. If you are using multi segment fund code then send the code segments with delimiters.

Ex: Fundseg1-Fundseg2-Fundseg3.

The fund information send as part of Cost BP will be rolled up to Funding Sheet or Cost Sheet.

All data element values provided as part of integration message will be validated against the action form selected as part of design in uDesigner. Validation includes required field, form validation etc..,

If the action form used for validation is auto-populating values from a Business Process picker then system will auto-populate these values if a valid value is provided for that Business Process Picker.

Attachments can be added to upper form and line items, and multiple records

Files can be sent as a single zipped file or individual files for the complete transaction record, the same file can be referenced multiple times in the same transaction. However, files contained in the zip file should still be described by separate attachment elements.

This API requires Integration Design to be published in uDesigner for the business process.

Attachment file cross references is not supported. The reference sub element under attachments should always have the value ‘No’.

Adding attachments is optional. If attachment elements are not provided, then the API behaves exactly like createBPRecord API.

This API is not compatible with non Java web services client platforms.

Total attachment size per request across all files is limited to 250 MB. It is recommended that files are compressed into 1 zip file for submission to the API.

This service can be used for both Project (Standard) and Shells of cost code type WBS and Generic.

The following XML tag <_refnum> </refnum> should be added under BP line item tag <_bp_lineitem> </bp_lineitem> to support the ability to import the line items for General Spends and Payment Applications.

Creating Complete BP Record for Document type BP with Folder Structure

The Web Services calls support creating new records and allows users to include the line items assigned to a particular folder.

- The tag <folder_path> specifies the path of the folder for the line item. This tag already exists in the current input XML for Document Type BP with folder structure.
- The folder path (<folder_path>) contains the names of the folders.
  - If the folders mentioned in the folder path do not exist, then the folders are created and the line items are added to the last folder mentioned in the folder path.
  - If the folder names already exist, then the line items are created under the last folder mentioned in the path.
- The Input integration XML contains the tags, as part of bp_lineitems elements.
- The syntax of the entry for the <folder_path> field is: “/<Folder Name1>/”
- If you create the line items under the root node, then the valid inputs within the tag are either “/” or blank.
- If the folder path is blank, then the line item is created at the root node “Attachments” and the folder name will be “Attachments”.
- If the line items have attachments in the <_attachment> elements, then the line items are added to the respective folder.

When the folders are being created through Web Services, the validations (in web interface of Unifier) are completed during the naming of the folders. You can enter non-ASCII characters as folder names. For example, you can enter Chinese characters as folder names in the folder path.

Sample XML Data

```
 <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
- <ulink>
  <_shortname>U90All</_shortname>
  <_servicename>CreateCompleteBPRecord</_servicename>
  <_bpname>92_Doc type With folder_Project_WF</_bpname>
  <_projectnumber>Proj-11</_projectnumber>
  <record_no />
- <List_wrapper>
- <_bp>
  <project_projectnumber />
  <project_projectname />
  <uuu_creation_date />
  <uuu_name>CC1</uuu_name>
  <record_no />
  <status />
  <uuu_issue_date />
  <uuu_revision_no />
  <uuu_latitude />
  <uuu_longitude />
  <uuu_address_code />
  <uuu_user_city />
  <uuu_user_country />
  <uuu_user_state />
- <_bp_lineitems>
  <folder_path>/Rev2</folder_path>
  <uuu_user_id />
  <short_desc />
  <uuu_name>abc.txt</uuu_name>
  <uuu_issue_date />
  <uuu_line_item_status />
- <_attachment>
  <file_name>xyz.txt</file_name>
  <title>Specifications</title>
  <revision_no>rev2</revision_no>
  <issue_date>02-10-2012</issue_date>
  <reference>No</reference>
  </_attachment>
  </_bp_lineitems>
- <_bp_lineitems>
  <folder_path>/Rev3</folder_path>
  <uuu_user_id />
  <short_desc />
  <uuu_name>pqr.txt</uuu_name>
  <uuu_issue_date />
  <uuu_line_item_status />
- <_attachment>
  <file_name>pqr.txt</file_name>
  <title>Specifications</title>
  <revision_no>rev2</revision_no>
```

```
<issue_date>02-10-2012</issue_date>
<reference>No</reference>
</_attachment>
</_bp_lineitems>
</_bp>
</List_wrapper>
</ulink>
```

Add BP Line Item

Description

This method adds a BP line item to an existing business process record based on Integration design that has been defined in uDesigner. For more information refer to uDesigner admin guide.

This method can also be used to create new BP records with line items, check LineXML elements sub-topic to find out more.

Support

This process supports all type of BPs.

Installation: ASP and Self host

Level	Yes or No
Company Level	Yes
Project or Shell Level	Yes

Prototype

public XMLObject addBPLineItem(String shortname, String authcode, string projectNumber, string BPName, String ItemXML);

Parameters

Parameter	Description
shortname	identifier of the company, company's short name
authcode	authentication key for the company, in text string
projectNumber	identifier of the project or shell in Primavera Unifier
BPName	identifier of the business process in Primavera Unifier, for example: Invoice. This is case sensitive, and should be the name as given in uDesigner
ItemXML	identifier for line item content in XML format

ItemXML Elements

The elements that include as part of LineXML parameter are dependent on the Integration interface design of your BP in uDesigner. ItemXML format can be exported (downloaded) from uDesigner. Elements with direction marked as "Input" or "Both" will be part of XML format.

The following is a list of elements that can be added as part of integration design and their behavior when you include them. Note that elements lists should be sent based on the type of BP for which you are creating a record.

Tag Name	BP Type	Description
<amount>	Cost	Amount is a required field if you are sending line item information under _bp_lineitems. You will see same value on Primavera Unifier.
<wbs_code>	Cost	WBS Code is a required field if you are sending line information under _bp_lineitems.
<record_no>	Any	You can send record number by using this tag. If you do not specify a record_no tag as part of message then a new record will be created with the line item information that is provided as part of ItemXML If you specify a record_no then system will add new line item(s) based on the data provided as part of ItemXML If the record_no is specified and if it does not exists then a new record will be created with the line item information that is provided as part of ItemXML.
<status>	Any	You can send record status by using this tag. This tag is valid only for multi-record BPs. If the record_no is specified and if it does not exists then a new record will be created with the line item information that is provided as part of ItemXML. If the BP is a workflow based then the record will be created and will be created and will stay in the initial status that is valid. User has to go through remaining workflow process. If it is a non-workflow based then the message can send a valid status and the record will take that status. If there is only one status defined for the BP then by default that status will be picked even if user sends a wrong status value.

Return Value

XMLObject (XMLObject is described in Appendix D of this document)

Sample XML Data

```
<ulink>
        <_shortname>acme</_shortname>
        <_authcode>acme_authcode</_authcode>
        <_servicename>addBPLineItem</_servicename>
        <_projectNumber>proj001</_projectNumber>
<_BPName>bp name</_BPName>
<_recordNumber>record number</_recordNumber>
<list_wrapper>
 <_bp>
  <record_no>1234<_record_no>
    <_bp_lineitems >
      <effective_date>2005-11-20</effective_date>
      <status>Open</status >
    </_bp_lineitems >
  </_bp>
</ list_wrapper >
</ulink>
```

Additional Information

This interface can be extended with additional data elements. In order for Primavera Unifier to recognize those elements, the data element name has to match those created in uDesigner for this business process.

YTB/AFC: If the tags YTB or AFC are present in the line items, Primavera Unifier processes these tags. If they are not present, Primavera Unifier will check the uDesigner BP setup configuration for Cost Adjustment. If this field has been set, the line item amount is treated as YTB/AFC and processed.

To add a line item to company level BP record then send null for projectNumber parameter. If you send a project number for a BP that is at Company level then you will get an error that BP name is not correct.

This service can be used for both Project (Standard) and Shells of cost code type WBS and Generic.

The following XML tag <_refnum> </refnum> should be added under BP line item tag <_bp_lineitem> </bp_lineitem> to support the ability to import the line items for General Spends and Payment Applications.

This service can be used to add lines to Payment Applications and General Spends type of business process. When adding lines to Payment Applications type of business process, existing lines if any will be removed and new lines will be added that are provided as part of the service call. For General spends line will be appended like any other business process.

Add Complete BP Line Item

Description

This method adds a BP line item to an existing business process record based on Integration design that has been defined in uDesigner. For more information refer to uDesigner admin guide. It allows multiple files to be attached to the BP line item.

This method can also be used to create new BP records with line items, check LineXML elements sub-topic to find out more.

Support

This process supports all type of BPs.

Installation: ASP and Self host

Level	Yes or No
Company Level	Yes
Project and Shell Level	Yes

Prototype

public XMLObject addCompleteBPLineItem(String shortname, String authcode, string projectNumber, string BPName, String ITemXML , FileObject[] fileobjects);

Parameters

Parameter	Description
shortname	identifier of the company, company's short name
authcode	authentication key for the company, in text string
projectNumber	identifier of the project or shell in Primavera Unifier
BPName	identifier of the business process in Primavera Unifier, for example: Invoice. This is case sensitive, and should be the name as given in uDesigner
ItemXML	identifier for line item content in XML format
fileobjects	Array of FileObjects, each FileObject is made up of two components: String filename (this should be the same name as in the record data <_attachment><filename> tag, and javax.activation.DataHandler as filestream. Files can be sent as a single zipped file or individual files

ItemXML Elements

The elements that include as part of LineXML parameter are dependent on the Integration interface design of your BP in uDesigner. ItemXML format can be exported (downloaded) from uDesigner. Elements with direction marked as "Input" or "Both" will be part of XML format.

The following is a list of elements that can be added as part of integration design and their behavior when you include them. Note that elements lists should be send based on the type of BP for which you are creating a record.

Tag Name	BP Type	Description
<amount>	Cost	Amount is a required field if you are sending line item information under _bp_lineitems. You will see same value on Primavera Unifier.
<record_no>	Any	You can send record number by using this tag. If you do not specify a record_no tag as part of message then a new record will be created with the line item information that is provided as part of ItemXML If you specify a record_no then system will add new line item(s) based on the data provided as part of ItemXML If the record_no is specified and if it does not exists then a new record will be created with the line item information that is provided as part of ItemXML.
<status>	Any	You can send record status by using this tag. This tag is valid only for multi-record BPs. If the record_no is specified and if it does not exists then a new record will be created with the line item information that is provided as part of ItemXML. If the BP is a workflow based then the record will be created and will be created and will stay in the initial status that is valid. User has to go through remaining workflow process. If it is a non-workflow based then the message can send a valid status and the record will take that status. If there is only one status defined for the BP then by default that status will be picked even if user sends a wrong status value.
attachment	Any	Multiple attachment elements can be added to the upper form and to line items (if applicable). The attachment element contains sub elements that identify: ▸ File name ▸ Title ▸ Revision Number ▸ Issue Date ▸ Version ▸ Reference Refer to the Sample XML below.

Return Value

XMLObject (XMLObject is described in Appendix D of this document)

Sample XML

```
<ulink>
    <_shortname>acme</_shortname>
    <_authcode>acme_authcode</_authcode>
    <_servicename>addCompleteBPLineItem</_servicename>
    <_projectNumber>proj001</_projectNumber>
<_BPName>bp name</_BPName>
<_recordNumber>record number</_recordNumber>
<list_wrapper>
<_bp>
    <record_no>ul-0022</record_no>
    <title>Test custom with & special care</title>
    <status>Pending</status>
    <amount>166.0</amount>
    <short_desc>Test custom & and more</short_desc>
    <_bp_lineitems>
        <record_id>22</record_id>
        <li_num>2</li_num>
        <amount>22.0</amount>
        <short_desc>two-2</short_desc>
        <UnitPrice>11.0</UnitPrice>
        <Quantity_1>11.0</Quantity_1>
        <_attachment>
            <file_name>Specs_PDF_1.pdf</file_name>
            <title>Specifications</title>
            <revision_no></revision_no>
            <issue_date></issue_date>
            <file_size></file_size>
            <version></version>
            <reference>No </reference>
        </_attachment>
    </_bp_lineitems>
</_bp>
</list_wrapper>
</ulink>
```

Additional Information

This interface can be extended with additional data elements. In order for Primavera Unifier to recognize those elements, the data element name has to match those created in uDesigner for this business process.

YTB/AFC: If the tags YTB or AFC are present in the line items, Primavera Unifier processes these tags. If they are not present, Primavera Unifier will check the uDesigner BP setup configuration for Cost Adjustment. If this field has been set, the line item amount is treated as YTB/AFC and processed.

To add a line item to company level BP record then send null for projectNumber parameter. If you send a project number for a BP that is at Company level then you will get an error that BP name is not correct.

Attachments can be added to line items

Files can be sent as a single zipped file or individual files for the complete transaction record, the same file can be referenced multiple times in the same transaction. However, files contained in the zip file should still be described by separate attachment elements.

This API requires Integration Design to be published in uDesigner for the business process.

Attachment elements specified as part of upper form data will be ignored by this API.

Attachment file cross references is not supported. The reference sub element under attachments should always have the value ‘No’.

Adding attachments is optional. If attachment elements are not provided, then the API behaves exactly like addBPLineItem API.

This API is not compatible with non Java web services client platforms.

Total attachment size per request across all files is limited to 250 MB. It is recommended that files are compressed into 1 zip file for submission to the API.

This service can be used for both Project (Standard) and Shells of cost code type WBS and Generic.

The following XML tag <_refnum> </refnum> should be added under BP line item tag <_bp_lineitem> </bp_lineitem> to support the ability to import the line items for General Spends and Payment Applications.

This service can be used to add lines to Payment Applications and General Spends type of business process. When adding lines to Payment Applications type of business process, existing lines if any will be removed and new lines will be added that are provided as part of the service call. For General spends line will be appended like any other business process.

Adding Complete BP Line Item for Document type BP with Folder Structure

You can add additional line items (including a document) into a particular folder in the Document Type BP with folder structure for an existing, or new, BP record.

- The tag <folder_path> specifies the path of the folder for the line item. This tag already exists in the current input XML for Document Type BP with folder structure.
- The folder path (<folder_path>) contains the names of the folders.
  - If the folders mentioned in the folder path do not exist, then the folders are created and the line items are added to the last folder mentioned in the folder path.
  - If the folder names already exist, then the line items are created under the last folder mentioned in the path.
- The Input integration XML contains the tags, as part of bp_lineitems elements.
- The syntax of the entry for the <folder_path> field is: “/<Folder Name1>/”
- If you create the line items under the root node, then the valid inputs within the tag are either “/” or blank.
- If the folder path is blank, then the line item is created at the root node “Attachments” and the folder name will be “Attachments”.
- If the line items have attachments in the <_attachment> elements, then the line items are added to the respective folder.

When the folders are being created through Web Services, the validations (in web interface of Unifier) are completed during the naming of the folders. You can enter non-ASCII characters as folder names. For example, you can enter Chinese characters as folder names in the folder path.

Get BP Record

Description

This method gets a specific Business Process record in Primavera Unifier based on Integration design that has been defined in uDesigner. For more information refer to uDesigner admin guide. This method works for both company level and project or shell level BPs.

Support

This process supports all type of BPs.

Installation: ASP and Self host

Level	Yes or No
Company Level	Yes
Project and Shell Level	Yes

Prototype

public XMLObject getBPRecord(String shortname, String authcode, string projectNumber, String BPName, String recordNumber);

Parameters

Parameter	Description
shortname	identifier of the company, company's short name
authcode	authentication key for the company, in text string
projectNumber	Identifier of the project or shell in Primavera Unifier. Pass null for Company Level BP
BPName	identifier of the business process in Primavera Unifier, for example: Invoice
recordNumber	identifier of the business process example: inv-0001

Return Value

XMLObject (XMLObject is described in Appendix D of this document)

Sample Method

getBPRecord("acme", "acme_authcode", "proj-01", "Invoice","innv-0001") will return the XML Record of the BP.

Sample XML INPUt

```
<ulink>
        <_shortname>acme</_shortname>
        <_authcode>acme_authcode</_authcode>
        <_servicename>getBPRecord</_servicename>
        <_projectNumber>PROJ-001</_projectNumber>
        <_BPName>BP Name</_BPName>
        <_recordNumber>record number</_recordNumber>
</ulink>
```

Additional Information

If the recordNumber is left blank or invalid recordNumber then the method will returns the BP's skeletal XML. BP template can also be downloaded from uDesigner.

To get company level BP record information, send null for projectNumber parameter. If you send a project number for a BP that is at Company level then you will get an error that BP name is not correct.

To get Single Record BP you need a record number populated on the record. If you do not have a record number populated then getBPRecord will return back the structure of the BP.

If you getting information for a Cost Type BP then you will always get <bItemID> and <fund_code> elements as part of return XML.

This service can be used for both Project (Standard) and Shells of cost code type WBS and Generic.

Get BP List

Description

This method gets all the records for a specific Business Process in Primavera Unifier based on Integration design that has been defined in uDesigner. For more information see the "uDesigner" section in the *Primavera Unifier Administration Guide*.

Support

This process supports all type of BPs.

Installation: ASP and Self host

Level	Yes or No
Company Level	Yes
Project and Shell Level	Yes

Prototype

public XMLObject getBPList (String shortname, String authcode, string projectNumber, String BPName, String[] fieldnames, String filterCondition, String[] filterValues);

Parameters

Parameter	Description
shortname	identifier of the company, company’s short name
authcode	authentication key for the company, in text string
projectNumber	identifier of the project or shell in Primavera Unifier
BPName	identifier of the business process in Primavera Unifier, for example: Invoice
fieldnames	names of the elements in the BP that you want to retrieve; this is an optional parameter. If not specified, all elements will be returned. Currently this parameter is not supported through Web services.
filterCondition	a list of supported condition, this is an optional parameter. If not specified, no conditions will be applied. Field names that you enter as filterCondition should match with the elements that you have selected while designing Integration Interface in uDesigner.
filterValues	a list of values depending on the filter conditions. Currently this parameter is not supported through Web services.

fieldnames

Values that you are going to send through fieldnames parameter are dependent on your BP design. Only those filed can send through fieldnames that are part of your BP design. You can send a name from line item list.

Return Value

XMLObject (XMLObject is described in Appendix D of this document)

Sample Method

getBPList("acme", "acme_authcode", "proj-01", "Invoice",null,null,null) will return the XML List of all the Records of the BP Invoice.

```
<ulink>
        <_shortname>acme</_shortname>
        <_authcode>acme_authcode</_authcode>
        <_servicename>getBPList</_servicename>
        <_projectNumber>PROJ-001</_projectNumber>
        <_BPName>BP Name</_BPName>
</ulink>
```

Additional Information

To get company level BP record information, do not send projectNumber parameter. If you send project number and the BP exists at company level then you will get an error that BP name is not correct.

To get Single Record BP you need a record number populated on to the record. If you do not have a record number populated then getBPRecord will return back the structure of the BP.

If you cannot send filterCondition and filter values currently then it is recommended that these be set to "null." If you want to filter out the records that you want to retrieve then you can use this element. If you want to include more than one filter condition then following this syntax

<_filterCondition>fieldname=value &amp;&amp; fieldname=value<_filterCondition>

Can add custom field clause through filters

Example: end_date=04-13-2005, &amp;&amp; record_no=uas-1200

If condition value is string (ex: ABC Inc) then value should be entered as follows <_filterCondition>Vendor Name='ABC Inc'</_filterCondition>. Observe that the value has single quotes. Only equal to "=" operation is supported right now.

> **Note:** If errors are present when evaluating the basic filter condition system will ignore it, but if there are errors in custom filter condition field names and values, system will return an empty list. All condition fields are evaluated using AND. If the combination has only basic fields and it has an error in processing all the records will be returned.

filterNames: Filter name will contain list of elements that you want to see as part of output XML. Ex: If you want to see project name and project number as output then include projectname and projectnumber.

filterValues, and filterNames are **not supported** as part of Web services for this release.

This service can be used for both Project (Standard) and Shells of cost code type WBS and Generic.

Update BP Record

Description

This method updates a record in Primavera Unifier.

Support

This service is available at both Company Level and Project or Shell Level BPs of type non-workflow Line Item type and can only update information on upper form. Line items will be ignored. Additionally the service is also available at Project or Shell Level for Cost (Non-Payment type), Simple, Text type and Document Type (Basic and Advanced) business process. In these cases, too, the service can only update information on upper form. Line items will be ignored.

Installation: ASP and Self host

Level	Yes or No
Company Level	Yes
Project and Shell Level	Yes

Prototype

public XMLObject updateBPRecord(String shortname, String authcode, String projectNumber, String BPName, String BPXML);

Parameters

Parameter	Description
shortname	identifier of the company, company’s short name
Authcode	authentication key for the company, in text string
projectNumber	Identifier of the project or shell in Primavera Unifier.
BPName	Identifier of the BP
BPXML	Content of the BP that needs to be updated

Return Value

XMLObject (XMLObject is described in Appendix D of this document)

Sample Method

updateBPRecord("acme", "acme_authcode", "Proj-0001", "Blanket PO", "<XML Content>")

Sample XML Data for <XML Content>

```
<ulink>
        <_shortname>acme</_shortname>
        <_authcode>acme_authcode</_authcode>
        <_servicename>updateBPRecord</_servicename>
        <_projectNumber>PROJ-001</_projectNumber>
        <_BPName>BP Name</_BPName>
<List_Wrapper>
<_bp>
   <record_no>BPO-0000</record_no>
   <title>My First BPO Integration</title>
    <status>Approved</status>
   <description>Testing Integration</description>
   <discipline></discipline>
   <deliverymethods>1</deliverymethods>
</_bp>
</List_Wrapper>
</ulink>
```

Additional Information

BPXML Elements**:** Elements are based on BP design. So in order to use this method user has to first use getBPRecord service. getBPRecord service will provide user with valid XML elements that are part of BP design.

If you want to update record for a BP at company level then do not pass projectnumber parameter.

If you send invalid projectnumber then you will get an error.

<record_no> tag as part of <_bp> element is required while send updateBPRecord message.

This service can be used for both Project (Standard) and Shells of cost code type WBS and Generic.

Sample XML

```
  <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
- <ulink>
  <_shortname>U90All</_shortname>
  <_servicename>updateBPRecord</_servicename>
  <_bpname>92_Doc type With folder_Project_NWF</_bpname>
  <_projectnumber>Proj-11</_projectnumber>
  <record_no />
- <List_wrapper>
- <_bp>
  <Decimal4 />
  <record_no>utwf2-0047</record_no>
  <status>Pending</status>
  <uuu_latitude />
  <uuu_longitude />
  <uuu_address_code />
- <_bp_lineitems>
  <folder_path>/Rev1/Rev2</folder_path>
  <Decimal3 />
  <uuu_contact_company />
  <uuu_creation_date />
  <uuu_name>b.txt</uuu_name>
  <Integer1 />
  <uuu_issue_date />
  <uuu_line_item_status>pending</uuu_line_item_status>
  </_bp_lineitems>
- <_bp_lineitems>
  <folder_path>/Rev4</folder_path>
  <Decimal3 />
  <uuu_contact_company />
  <uuu_creation_date />
  <uuu_name>b.txt</uuu_name>
  <Integer1 />
  <uuu_issue_date />
  <uuu_line_item_status>pending</uuu_line_item_status>
  </_bp_lineitems>
- <_bp_lineitems>
  <folder_path />
  <Decimal3 />
  <uuu_contact_company />
  <uuu_creation_date />
  <uuu_name>b.txt</uuu_name>
  <Integer1 />
  <uuu_issue_date />
  <uuu_line_item_status>pending</uuu_line_item_status>
  </_bp_lineitems>
  </_bp>
  </List_wrapper>
  </ulink>
```

Update Complete BP Record

Description

This method updates a record in Primavera Unifier. It also provides the ability to add attachments to the upper form. It will not modify/delete existing attachments.

Support

This service is available at both Company Level and Project or Shell Level BPs of type non-workflow Line Item type and can only update information on upper form. Line items will be ignored.   Additionally the service is also available at Project or Shell Level for Cost (Non-Payment type), Simple, Text type and Document Type (Basic and Advanced) business process. In these cases, too, the service can only update information on upper form. Line items will be ignored.

Installation: ASP and Self host

Level	Yes or No
Company Level	Yes

Level	Yes or No
Project and Shell Level	Yes

Prototype

public XMLObject updateCompleteBPRecord(String shortname, String authcode, string projectNumber, String BPName, String BPXML, String iszipfile, FileObject[] fileobjects);

Parameters

Parameter	Description
shortname	identifier of the company, company’s short name
Authcode	authentication key for the company, in text string
projectNumber	Identifier of the project or shell in Primavera Unifier.
BPName	Identifier of the BP
BPXML	Content of the BP that needs to be updated
iszipfile	This tag identifies if the file provided under the fileobjects tag is a zip file or not. Values for this tag are "yes" or "no".
fileobjects	Array of FileObjects, each FileObject is made up of two components: String filename (this should be the same name as in the record data <_attachment><filename> tag, and javax.activation.DataHandler as filestream. Files can be sent as a single zipped file or individual files..

Return Value

XMLObject (XMLObject is described in Appendix D of this document)

Sample XML Data for <XML Content>

```
<ulink>
        <_shortname>acme</_shortname>
        <_authcode>acme_authcode</_authcode>
        <_servicename>updateCompleteBPRecord</_servicename>
        <_projectNumber>PROJ-001</_projectNumber>
        <_BPName>BP Name</_BPName>
<_recordNumber>record number</_recordNumber>
<List_Wrapper>
<_bp>
        <record_no>ul-0022</record_no>
        <title>Test custom with & special care</title>
        <due_date/>
        <end_date/>
        <status>Pending</status>
        <amount>166.0</amount>
        <short_desc>Test custom & and more</short_desc>
        <otherCompanyID/>
        <recordDate/>
        <_attachment>
                <file_name>Specs_PDF_1.pdf</file_name>
                <title>Specifications</title>
                <revision_no></revision_no>
                <issue_date></issue_date>
                <file_size></file_size>
                <version></version>
                <reference>No </reference>
        </_attachment>
</_bp>
</List_Wrapper>
</ulink>
```

Additional Information

BPXML Elements**:** Elements are based on BP design. So in order to use this method user has to first use getBPRecord service. getBPRecord service will provide user with valid XML elements that are part of BP design.

If you want to update record for a BP at company level then do not pass projectnumber parameter.

If you send invalid projectnumber then you will get an error.

<record_no> tag as part of <_bp> element is required while send updateBPRecord message.

Attachments can be added only to the upper form.

Files can be sent as a single zipped file or individual files for the complete transaction record, the same file can be referenced multiple times in the same transaction. However, files contained in the zip file should still be described by separate attachment elements.

This API requires Integration Design to be published in uDesigner for the business process.

Attachment file cross references is not supported. The reference sub element under attachments should always have the value ‘No’.

Adding attachments is optional. If attachment elements are not provided, then the API behaves exactly like updateBPRecord API.

This API is not compatible with non Java web services client platforms.

Total attachment size per request across all files is limited to 250 MB. It is recommended that files are compressed into 1 zip file for submission to the API.

This service can be used for both Project (Standard) and Shells of cost code type WBS and Generic.

Updating Complete BP Record for Document type BP with Folder Structure

The existing Web Services call updates the Upper form elements and you can add attachments to the Upper form.

> **Note**: The Document type BP with folder structure does not have record-level attachments.

Update BP Record V2

Description

This option will allow external systems to update a business process record and advance it to next step.

Support

Installation: ASP and Self host

Level	Yes or No
Company Level	Yes
Project and Shell Level	Yes
Program Level	No

Prototype

public XMLObject updateBPRecordV2(String shortname, String authcode, String shellNumber, String BPName, String BPXML, String options).

Parameters

Parameter	Description
Shortname	identifier of the company, company’s short name
Authcode	authentication key for the company, in text string
shellNumber	Identifier of the project or shell in Primavera Unifier.
BPName	Identifier of the BP

BPXML	Content of the BP that needs to be updated and moved to next step
Options	This tag will allow user to send additional information related to workflow and line items. Following are the tags under options tag and their description ▸ LineItemIdentifier: Name of the DE on the detail form / line item to uniquely identify line items. ▸ WFCurrentStepName: Name of the current workflow setup ▸ WFActionName: Action that should be taken to move the WF to next step ▸ SYS Auto Sequence type DE

Return Value

XMLObject (XMLObject is described in Appendix D of this document)

Sample Method

updateBPRecordV2("acme", "acme_authcode", "PRJ-0001", "Contracts", <XML content>,<options>)

Sample uLink XML Input Data

```
<ulink>
        <_shortname>acme</_shortname>
        <_authcode>acme_authcode</_authcode>
        <_servicename>updateBPRecordv2</_servicename>
        <_projectnumber>PROJ-00001</_projectnumber>
        <_bpname>Purchase Requisition</_bpname>
            <_options>
                 <LineItemIdentifier>sysAutoSeqDet</LineItemIdentifier>
                 <WFCurrentStepName>Step 3</ WFCurrentStepName>
                <WFActionName>Line 2</WFActionName>
            </_options>
          <List_Wrapper>
          <_bp>
              <record_no>ACT-5119</record_no>
              <status>Approved</status>
             <_bp_lineitems>
                 <sysAutoSqDet>0001</ sysAutoSqDet> </line_no_de>
                 <wbs_code>1-1</wbs_code>
                 <amount>1000</amount>
             </_bp_lineitems>
             <_bp_lineitems>
                <line_no_de>LI2</line_no_de>
                <wbs_code>1-2</wbs_code>
                <amount>1000</amount>
             </_bp_lineitems>
             <_delete_bp_lineitems>LI3, LI4</_delete_bp_lineitems>
            </_bp>
           </List_Wrapper>
</ulink>
```

Additional Information

If the shell number is null then the record will be assumed for company level record.

Step to which record will be "Send To" should be pre-assigned

options.WFCurrentStepName: Value of this tag identifies the current step of the record. This is to ensure that web service call is called for correct step with correct action.

options.WFActionName: Value of this tag determines the action that should be taken. This action name will be validated against the WFCurrentStepName

If options.WFActionName tag is empty then system will just update record. WFCurrentStepName in this case is not validated.

If options.WFActionName tag has a value then system will push workflow to next step based on the value of WFActionName tag. All form level validations will be triggered before pushing workflow to the next step. In this scenario WFCurrentStepName will be validated to make sure that the step is valid and action is valid.

Options.WFActionName tag will be ignored for Non-workflow business process

Validations while processing WFCurrentStepName and WFActionName tags

If WFActionName tag **IS** empty then do not validate WFCurrentStepName

If WFActionName tag **IS NOT** empty then validate WFCurrentStepName

Record is still on the step identified by WFCurrentStepName. If not system should error out.

Step identified by WFCurrentStepName value has "Enable Step for Integration" option is set to "Yes". If not system should error out.

Validate WFActionName is valid action that can be taken from step identified by the value of WFCurrentStepName.

After going through all validations rules, system should move record to next step. Creator selected on the Admin → BP Setup → Workflow (node) → Auto Creation (tab) → Creator should be the person who takes action on the step.

It is possible that someone would have accepted the task on step identified by the value of WFCurrentStepName. (This can happen because "Assignees" option for this step under BP Setup is not disabled). There are three possible scenarios under this condition

No one accepted the task: In this scenario system will automatically accepts the task, update record and line items and move record to next step. System will remove task from all other users list.

Someone accepted the task: In this scenario system will update record and move it to next step and will remove task from other users list including the use who accepted the task.

Someone accepted the task and move workflow to next step. System will generate an error message as the task is already accepted and WF is moved out of that step

Options.LineItemIdentifier: There are two scenarios

LineItemIdentifier value is empty:

In this scenario system will add all line items.

<_delete_bp_lineitems> tag will be ignored.

LineItemIdentifier value is not empty:

In this scenario value provided as part of LineItemIdentifier should be a tag under _bp_lineitems. If not system will error out.

If LineItemIdentifier value is an element under _bp_lineitems then System will try to query up a line item based on the value provided under this element.

If the line is found then it will be updated.

If two lines are found then system will error out saying that there are duplicate rows.

If the line is not found then it will be added.

Above mentioned behavior is applicable to all lines on all tabs.

System will not update / remove or add line items of cost type of business process records that has reached terminal status (not just end step) using this new service.

Options.LineItemIdentifier will be ignored for following type of business processes

Simple

Document → with detail form

Text

Options.WFActionName and WFCurrentStepName will be ignored for non workflow business process

The following XML tag <_refnum> </refnum> should be added under BP line item tag <_bp_lineitem> </bp_lineitem> to support the ability to import the line items for General Spends and Payment Applications.

This service can be used to add lines to Payment Applications and General Spends type of business process. When adding lines to Payment Applications type of business process, existing lines if any will be removed and new lines will be added that are provided as part of the service call. For General spends line will be appended like any other business process.

> **Note:** *updateBPRecordv2* cannot be used to update existing line items for Payment Applications type of business process.

Updating BP Record V2 for Document type BP with Folder Structure

The Web Service call supports the adding of line items to a specific folder to an existing Document type BP record.

Sample XML

```
  <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
- <ulink>
  <_shortname>U90All</_shortname>
  <_servicename>updateBPRecordV2</_servicename>
  <_bpname>92_Doc type With folder_Project_NWF</_bpname>
  <_projectnumber>Proj-11</_projectnumber>
  <record_no />
- <List_wrapper>
- <_bp>
  <Decimal4 />
  <record_no>utwf2-0047</record_no>
  <status>Pending</status>
  <uuu_latitude />
  <uuu_longitude />
  <uuu_address_code />
- <_bp_lineitems>
  <folder_path>/Rev1/Rev2</folder_path>
  <Decimal3 />
  <uuu_contact_company />
  <uuu_creation_date />
  <uuu_name>b.txt</uuu_name>
  <Integer1 />
  <uuu_issue_date />
  <uuu_line_item_status>pending</uuu_line_item_status>
  </_bp_lineitems>
- <_bp_lineitems>
  <folder_path>/Rev4</folder_path>
  <Decimal3 />
  <uuu_contact_company />
  <uuu_creation_date />
  <uuu_name>b.txt</uuu_name>
  <Integer1 />
  <uuu_issue_date />
  <uuu_line_item_status>pending</uuu_line_item_status>
  </_bp_lineitems>
- <_bp_lineitems>
  <folder_path />
  <Decimal3 />
  <uuu_contact_company />
  <uuu_creation_date />
  <uuu_name>b.txt</uuu_name>
  <Integer1 />
  <uuu_issue_date />
  <uuu_line_item_status>pending</uuu_line_item_status>
  </_bp_lineitems>
  </_bp>
  </List_wrapper>
  </ulink>
```

Get Complete BP Record

Description

Methods returns back complete BP record information. This includes any attachments, general comments etc.

Support

This service is available at both Company Level and Project and Shell Level BPs

Installation: ASP and Self host

Level	Yes or No
Company Level	Yes
Project and Shell Level	Yes

Prototype

public XMLObject getCompleteBPRecord(String shortname, String authcode, String projectNumber, String BPName, String record_no);

Parameters

Parameter	Description
shortname	identifier of the company, company's short name
Authcode	authentication key for the company, in text string
projectNumber	Identifier of the project or shell in Primavera Unifier.
BPName	Identifier of the BP
record_no	A valid record number.

Return Value

XMLObject (XMLObject is described in Appendix D of this document)

Sample Method

getCompleteBPRecord("acme", "acme_authcode", "Proj-0001", "Invoices", "Inv-0001")

Additional Information

If you want to get company level BP record information then pass null for projectnumber element.

You will get a zip file as an output. The zip file contains any attachments and a PDF file that contains BP record information.

<record_no> tag as part of <_bp> element is required while send getCompleteBPRecord message.

This service can be used for both Project (Standard) and Shells of cost code type WBS and Generic.

Getting Complete BP Record for Document type BP with Folder Structure

The existing Get Complete BP record displays the <folder_path> in <bp_lineitems> elements and <attachments_folder> elements. The folder path seen in the line items elements matches the folder path seen in the <attachments_folder>. The <folder_name> is displayed with the <folder_path> in the <attachments_folder> section.

Sample XML

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
- <ulink>
  <_shortname>U90All</_shortname>
  <_projectnumber>Proj-11</_projectnumber>
  <_bpname>15.1 Doc</_bpname>
  <_servicename>getCompleteBPRecord</_servicename>
  <_recordNumber>ux12pg-0006</_recordNumber>
  </ulink>
```

Object Methods

Object methods are used for creating or retrieving data from objects such as Standard Projects or Users, which have not been modified based on an integration design defined in uDesigner.

Create Object

Description

This method creates a specific object in Primavera Unifier.

Support

This method only supports creation of user.

Installation: ASP and Self host

Level	Yes or No
Company Level	Yes
Project and Shell Level	Not Applicable

Prototype

public XMLObject CreateObject(String shortname, String authcode, String objectName, String objectXML);

Parameters

Parameter	Description
Shortname	identifier of the company, company's short name
Authcode	authentication key for the company, in text string
objectName	identifier of the object. Names are case sensitive. Valid value: user_info
objectXML	content of the Object including element name and value in XML format

Return Value

XMLObject (XMLObject is described in Appendix D of this document)

Sample Method

createObject("acme", "acme_authcode", "user_info", "<XML Content>")

objectXML Elements

Tag Name	Required	Description	Valid values
<user_info>	Yes	Base tag for new user data.	
<username>	Yes	User name that is going to be used by user to sign in to Primavera Unifier.	
<password>	Yes	User's password	
<firstname>	No	First name of the user.	
<lastname>	No	Last name of the user.	
<title>	No	Title information	
<email>	No	Email of the user	
<workphone>	No	Work phone number of the user	
<homephone>	No	Home phone number of the user	
<fax>	No	Fax number of the user	
<cellphone>	No	Cell phone number of the user	
<pager>	No	Pager number of the user	
<status>	No	Status of the newly created user (Active, Inactive, On-Hold)	Active: 1 Inactive: 0 On-Hold: 2

Tag Name	Required	Description	Valid values
<timezone>	No	Time zone of the user.	Please refer to Appendix → Time zone section for time zone data. You have to send the data from value column. Ex: for time zone GMT -11:00, Samoa Standard Time (Samoa) send → 10
<dateformat>	No	User preference date format.	Please refer to Appendix → Date Format section. You have to send the data from value column. Ex: for MM/DD/YYYY HH:MM AM send → 0

Sample XML Data

```
<list_wrapper>
        <user_info>
                <username>userone</username>
                <firstname>User</firstname>
                <lastname>One</lastname>
                <title>user title</title>
                <email>userone@acme.com</email>
                <workphone>111-11-11111</workphone>
                <homephone>222-22-22222</homephone>
                <fax></fax>
                <cellphone></cellphone>
                <pager></pager>
                <timezone>10</timezone>
                <dateformat>0</dateformat>
                <password>useronepassword</password>
        </user_info>
</list_wrapper>
```

Additional Information

The Web Services call for user administration includes a XML tag to support the Proxy login setup. The Data Element (DE) for the XML tag is: "`uuu_user_proxy_config`."

This is not a required field, so the default value remains unchecked.

Users can be created using the CreateOject WS method, and the XML tag is: "`allowproxy`."

To map the values for checkbox:

- If `uuu_user_proxy_config` or `allowproxy` is set to Yes, then the Check box "Do not allow Proxies" in Proxy tab should be unchecked.

- If `uuu_user_proxy_config` or `allowproxy` is set to No, then the Check box “Do not allow Proxies” in Proxy tab should be checked.

Update Object

Description

This method updates a specific object in Primavera Unifier.

Support

This method can only support project and user information.

Installation: ASP and Self host

Level	Yes or No
Company Level	Yes
Project and Shell Level	Not Applicable

Prototype

public XMLObject UpdateObject(String shortname, String authcode, String objectName, String objectXML);

Parameters

Parameter	**Description**
Shortname	identifier of the company, company’s short name
Authcode	authentication key for the company, in text string
ObjectName	name of the object. Valid values: project_info or user_info
ObjectXML	content of the Object including element name and value in XML format

Return Value

XMLObject (XMLObject is described in Appendix D of this document)

Sample Method

updateObject("acme", "acme_authcode", "project_info", "<XML Content>")

Sample XML Data for <XML Content>

```
<list_wrapper>
    <project_info>
        <projectnumber>project_001</projectnumber>
        <extended_id>1</extended_id>
    </project_info>
</list_wrapper>

<list_wrapper>
    <user_info>
        <firstname>sam</firstname >
        <lastname>lee</lastname >
        <email>cc@cc.com</email>
    </user_info>
</list_wrapper>
```

Additional Information

If a new project is created by copying from a Template or another existing Project then users and groups will be copied by default along with permissions.

Address information provided through XML tags is of type Project Address, Billing, Shipping and Billing and Shipping.

List of valid countries refer to Country List under Appendix

If a company is configured to have auto numbering for projects then any information send through integration for projectnumber tag will be ignored and system will provide a number automatically based on sequence.

This service can only be used to update Project (Standard).

Get Object List

Description

This method gets object list from Primavera Unifier.

Support

This process supports retrieval of project information and user information only.

Installation: ASP and Self host

Level	Yes or No
Company Level	Yes
Project and Shell Level	Not Applicable

Prototype

public XMLObject getObjectList(String shortname, String authcode, string objectName, String[] fieldNames , String filterCondition, String[] filterValues);

Parameters

Parameter	Description
Shortname	identifier of the company, company's short name
Authcode	authentication key for the company, in text string
objectName	identifier of the object. Only supports project_info, user_info; this name is case sensitive
fieldnames	names of the elements in the Object that is requested; this is an optional parameter. If not specified, all elements will be returned.
filterCondition	a list of supported condition, this is an optional parameter. If not specified, no conditions will be applied.
filterValues	a list of values depending on the filter conditions. Currently this parameter is not supported through Web services.

fieldnames

Following list will give you a list of valid values that can be part of fieldnames parameter. These are the fields that can be retrieved as part of output if they are included in fieldnames parameters.

fieldnames for project_info

Values	Description
setupdate	Setup date of the project
constructiontype	Type of construction
Projectsite	Project site information
Projectnumber	Number of the project
Projectname	Name of the project
Description	Description of the project
Status	Status of the project
Startdate	Start date of the project
Plannedcompletion	Planned Completion date of the project
Revisedcompletion	Revised completion date of the project
DesignComplete	Design Complete % of the project
Constrcomplete	Construction complete % of the project
Notes	Notes information of the project
Projectphase	Project phase of the project
Createdate	Creation date of the project

fieldnames for user_info

Values	Description
Username	User name of the user
Firstname	First name of the user
Lastname	Last name of the user
Email	Email of the user
Workphone	Work phone number of the user
Homephone	Home phone number of the user
Fax	Fax number of the user
Cellphone	Cell phone number of the user
Pager	Pager of the user
Status	Status of the user. Return value can be ▸ 1 for Active ▸ 0 for Inactive ▸ 2 for On-Hold
Title	Title of the user
Createdate	Creation date of the user
Timezone	Timezone information of the user. Return value can be a number from Time Zone list. Please check Appendix for Time Zone information.
Dateformat	Dateformat information of the user. Return value can be a number from Date Format list. Please check Appendix for Date Format information.

Return Value

XMLObject (XMLObject is described in Appendix D of this document)

Sample Method

```
getObjectList("user", null, null, null);
```

Sample XML Data

```
<ulink>
        <_shortname>shortname</_shortname>
        <_authcode>acme_authcode</_authcode>
        <_servicename>getObjectList</_servicename>
        <_objectName>project_info</_objectName>
        <_fieldnamelist>
                <_fieldName>projectnumber</_fieldName>
                <_fieldName>projectname</_fieldName>
        </_fieldnamelist>
        <_filterCondition>status</_filterCondition>
        <_filtervaluelist>
                <_filterValues>Active</_filterValues>
        </_filtervaluelist>
</ulink>
```

Additional Information

If you cannot send filterCondition and filter values currently then it is recommended that these be set to "null."

filterCondition and filterValues: Only condition that is supported in current release is status element of project.

The following syntax should be used to filter project information based on status.

```
        <_filterCondition>status</_filterCondition>
        <_filtervaluelist>
                <_filterValues>Active</_filterValues>
        </_filtervaluelist>
```

filternames: Filter name will contain list of elements that you want to see as part of output XML. Ex: If you want to see project name and project number as output then include projectname and projectnumber.

The following syntax should be used to filter elements that you want to see in output XML.

```
        <_fieldnamelist>
                <_fieldName>projectnumber</_fieldName>
                <_fieldName>projectname</_fieldName>
        </_fieldnamelist>
```

This service can be used only to retrieve information related to Project (Standard).

Project Methods

Create Project

Description

This method creates a project in Primavera Unifier either based on a template or an existing project.

Support

This method can only support creation of project.

Installation: ASP and Self host

Level	Yes or No
Company Level	Yes
Project and Shell Level	Not Applicable

Prototype

public XMLObject createProject(String shortname, String authcode, String copyFromProject, String projectXML);

Parameters

Parameter	Description
Shortname	identifier of the company, company's short name
Authcode	authentication key for the company, in text string
copyFromProject	this can be a project number or a template number.
projectXML	content of the project that needs to be created.

projectXML Elements

Tag Name	Require d	Description	Valid values
<project_info>	Yes	Base tag for new project data. Only one project can be created at a time.	
<projectnumber>	Yes	Project number of the new project that is getting created. If this project number already exists you will get an error.	User defined number.
<projectname>	Yes	Project name of the new project that is getting created. If this project name already exists you will get an error.	User defined name.
<constructiontype>	No	Type of construction (New Construction or Retrofit/Remodel).	New Construction: 1 Retrofit/Remodel: 0

Tag Name	Require d	Description	Valid values
<status>	No	Status of the Project (Active, Inactive, On-Hold)	Active: 1 Inactive: 0 On-Hold: 2
<typeofproject>	No	Type of Project. Values are based on the user dataset for "Project Type" data definition	
<projectsite>	No	Project site. Values are based on the user dataset for "Project Site" data definition	
<startdate>	No	Start Date of the project.	
<plannedcompletion >	No	Planned Completion date	
<revisedcompletion>	No	Revised Completion date	
<designcomplete>	No	Design % complete	
<constrcomplete>	No	Construction % complete	
<notes>	No	Notes	
<schedulestatus>	No	Schedule status of the project	On Schedule and on Budget: 0 Potential impact on schedule or both: 1 Impact to schedule or Budget or both: 2
<projectphase>	No	Project phase information. Values are based on the user dataset for "Project Phase" data definition	

Note: If you do not provide values for elements that are not required then the values from template or project that is used to create new project are used.

Return Value

XMLObject (XMLObject is described in Appendix D of this document). Message will contain new Project number and name that is created within Primavera Unifier.

Sample Method

createProject ("acme", "acme_authcode", "Proj-001", "<XML Content>")

Sample XML Data for <XML Content>

```
<List_Wrapper>
<project_info>
   <projectnumber>Proj-002</projectnumber>
   <projectname>Project 1</projectname>
    <status>2</status>
   <description>Project created through Integration</description>
    <constructiontype>1</ constructiontype>
   <typeofproject>Retrofit</typeofproject>
    <projectsite>Site 1</projectsite>
   <startdate>01-01-06</startdate>
   <plannedcompletion>01-01-08</plannedcompletion>
   <revisedcompletion>01-01-09</revisedcompletion>
   <designcomplete>10.0</designcomplete>
   <constrcomplete>1</constrcomplete>
   <notes>testing integration</notes>
    <schedulestatus>1</<schedulestatus>
     <projectphase>Planning</projectphase>
</project_info>
</List_Wrapper>
```

Additional Information

If you are sending a message to update user information then you have to send a user name that already exists in Primavera Unifier.

If a new project is created by copying from a Template or another existing Project then users and groups will be copied by default along with permissions.

This service can be used to create Project (Standard).

Create Project with additional options

Description

This method creates a project in Primavera Unifier either based on a template or an existing project. This method is enhanced to have additional options to support copying of modules and address information from template or project.

Support

This method can only support creation of project.

Installation: ASP and Self host

Level	Yes or No
Company Level	Yes
Project Level	Not Applicable

Prototype

public XMLObject createProject(String shortname, String authcode, String copyFromProject, String projectXML);

Parameters

Parameter	Description
Shortname	identifier of the company, company's short name
Authcode	authentication key for the company, in text string
copyFromProject	this can be another project number or a template number.
projectXML	content of the project that needs to be created.

projectXML Elements

Tag Name	Required	Description	Valid values
<project_info>	Yes	Base tag for new project data. Only one project can be created at a time.	
<projectnumber>	Yes	Project number of the new project that is getting created. If this project number already exists you will get an error.	User defined number.
<projectname>	Yes	Project name of the new project that is getting created. If this project name already exists you will get an error.	User defined name.
<constructiontype>	No	Type of construction (New Construction or Retrofit/Remodel).	New Construction: 1 Retrofit/Remodel: 0
<status>	No	Status of the Project (Active, Inactive, On-Hold)	Active: 1 Inactive: 0 On-Hold: 2
<typeofproject>	No	Type of Project. Values are based on the user dataset for "Project Type" data definition	
<projectsite>	No	Project site. Values are based on the user dataset for "Project Site" data definition	

Tag Name	Required	Description	Valid values
<startdate>	No	Start Date of the project.	
<plannedcompletion>	No	Planned Completion date	
<revisedcompletion>	No	Revised Completion date	
<designcomplete>	No	Design % complete	
<constrcomplete>	No	Construction % complete	
<notes>	No	Notes	
<schedulestatus>	No	Schedule status of the project	On Schedule and on Budget: 0 Potential impact on schedule or both: 1 Impact to schedule or Budget or both: 2
<projectphase>	No	Project phase information. Values are based on the user dataset for "Project Phase" data definition	
<description>	No	Description of the project	
<location>	No	Base tag for project location information	This is a new tag added for this enhancement. This tag is optional. If this tag is not provided then Primavera Unifier will create Project by copying "Project Address" information from template.
<address>	No	Base tag for address information	This tag can be repeated to provide multiple address types
<address1>	Yes	Address 1 field of the project	Is child element of <address> element. This element is required if address tag is provided
<address2>	No	Address 2 field of the project	Is child element of <address> element

Tag Name	Required	Description	Valid values
<city>	Yes	City field of the project	Is child element of <address> element. This element is required if address tag is provided
<state>	Yes	State field of the project	Is child element of <address> element. This element is required if address tag is provided
<country>	Yes	Country field of the project	Is child element of <address> element. This element is required if address tag is provided Value should be coming from the list provides under Appendix → Country List section. User should send in Label as a valid value
<phone>	No	Phone field of the project	Is child element of <address> element
<fax>	No	Fax filed of the project	Is child element of <address> element
<addresstype>	Yes	Type of address	Is child element of <address> element Following are the valid values for this element ▸ Project Address ▸ Billing ▸ Shipping ▸ Billing and Shipping
<options>	No	Base tag for options.	This is a new tag added for this enhancement. This tag is optional. If this tag is not provided then Primavera Unifier will create Project by copying user / groups from another project or template.

Tag Name	Required	Description	Valid values
<copyModules>	No	Option to either copy modules or not.	This tag can only go as a child element under <options> tag. This tag is identify either to create a new project by copying all modules including user / group or just creates a new project by copying user / group from another project or template. Only valid values are Yes / No. Yes means copy all modules **including user / groups. No means copy only user / groups. If tag is not provided or value is empty then default behavior will be to create project by copying user / groups.**
<projectCurrency>	No	Option to specify the project currency	A new child element of <projectCurrency> will be available within the existing <options> tag. Valid values for this tag are the code for the currency rather than the names. For example, USD is valid, United States Dollars is not. If the code is not a valid code then the web service will fail. If the tag is provided and the value is empty then template code will be selected.

Note: If you do not provide values for elements that are not required then the values from template will be used to create new project.

Return Value

XMLObject (XMLObject is described in Appendix D of this document). Message will contain new Project number and name that is created within Primavera Unifier.

Sample Method

createProject ("acme", "acme_authcode", "Proj-001", "<XML Content>")

Sample XML

```
<List_Wrapper>
<project_info>
   <projectnumber>Ulink-0004</projectnumber>
   <projectname>Ulink Project-004</projectname>
   <description>FromUlink-9999999</description>
   <typeofproject>Retrofit</typeofproject>
    <projectsite></projectsite>
    <startdate>05-05-2008</startdate>
    <plannedcompletion>05-05-2009</plannedcompletion>
     <revisedcompletion>05-05-2009</revisedcompletion>
    <designcomplete>10.0</designcomplete>
     <constrcomplete>10.0</constrcomplete>
     <notes>testing integration with address_info</notes>
     <projectphase>Planning</projectphase>
    <location>
       <address>
        <address1>111 Independence Drive</address1>
         <address2> </address2>
         <city>Menlo Park</city>
        <state>California</state>
        <zip>95321</zip>
         <country>United States</country>
        <phone>222-222-2222</phone>
        <fax>555-555-5555</fax>
        <addresstype>Project Address</addresstype>
       </address>
       <address>
        <address1>111 Independence Drive</address1>
         <address2> </address2>
         <city>Menlo Park</city>
        <state>California</state>
        <zip>95321</zip>
         <country>United States</country>
        <phone>222-222-2222</phone>
        <fax>555-555-5555</fax>
        <addresstype>Billing</addresstype>
       </address>
    </location>
     <options>
       <copyModules>no</copyModules>
           <projectCurrency>USD</projectCurrency>
   </options>
</project_info>
</List_Wrapper>
```

Additional Information

If a new project is created by copying from a Template or another existing Project then users and groups will be copied by default along with permissions.

Address information provided through XML tags is of type Project Address, Billing, Shipping and Billing and Shipping.

List of valid countries refer to Country List under Appendix

If a company is configured to have auto numbering for projects then any information send through integration for projectnumber tag will be ignored and system will provide a number automatically based on sequence.

This service can be used create Project (Standard).

Cost Sheet Methods

Get Column Data

Description

This method gets data from a project or shells (with cost codes WBS) cost sheet column. getColumnData does not support Assigned Budget column tough it is a manual column.

Support

This method can only support for project or shell level cost sheet column that is defined as Manual Entry Method (either Direct Entry or Line Item Content)

Installation: ASP and Self host

Level	Yes or No
Company Level	No
Project and Shell Level	Yes
Program Level	No

Prototype

public XMLObject getColumnData(String shortname, String authcode, string projectNumber, String ColumnName);

Parameters

Parameter	Description
shortname	identifier of the company, company’s short name
authcode	authentication key for the company, in text string
projectNumber	Identifier of the project or shell in Primavera Unifier.
columnName	Identifier of the cost sheet column in Primavera Unifier. This is case sensitive, and should be the name of Column that is defined in Primavera Unifier.

Return Value

XMLObject (XMLObject is described in Appendix D of this document). Response will contain following data elements.

Tag Name	Description
<wbs_code>	WBS Code is a required field.
<short_description>	Short Description
<long_description>	Long Description
<work_package>	Work Package Name
<spends_category>	Spends Category Label
<quantity>	Quantity
<unit_of_measure>	Unit of Measure
<unit_cost>	Unit Cost Label
<amount>	Amount

Sample Method

getColumnData("acme", "acme_authcode", "project_info","my column name")

Sample XML Return Data

```
<List_Wrapper>
<column>
<wbs_code>1-0-0~~1-1-0~~1-1-1</wbs_code>
<short_description>Undistributed balance</short_description>
<long_description></long_description>
<work_package></work_package>
<spends_category>Labor</spends_category>
<quantity>1.0</quantity>
<unit_of_measure>wk</unit_of_measure>
<unit_cost>100000.0</unit_cost>
<amount>100000.0</amount>
</column>
<column>
<wbs_code>5-0-0~~5-1-0</wbs_code>
<short_description>Testing Again</short_description>
<long_description></long_description>
<work_package></work_package>
<spends_category>Labor</spends_category>
<quantity>1.0</quantity>
<unit_of_measure>wk</unit_of_measure>
<unit_cost>10000.0</unit_cost>
<amount>10000.0</amount>
</column>
</List_Wrapper>
```

Additional Information

Following are the failure reasons and description

Reason	Description
Column is not valid	Invalid Column Name
Column is not manual	Column is not defined as manual entry

If the cost sheet column entry method is "Line Item Content" then this web service call will only return lines for WBS Code which has valid lines. For example, if there are 2 WBS Codes on cost sheet and one has line item content and other does not then only WBS Code with line item content will be returned.

If the cost sheet column entry method is "Direct Entry" then this web service call will return a line for a WBS Code even if there is no data manually entered against that WBS Code in project or shell cost sheet. For example, if there are 2 WBS Codes on cost sheet and one has a value entered directly on cost sheet and other does not have any value, even then this service call will return 2 lines, one for each WBS Code. Following are the elements that will be returned as part of each WBS Code.

```
<wbs_code></wbs_code>
<short_description>Direct Entry Line Item</short_description>
<long_description></long_description>
<work_package></work_package>
<spends_category></spends_category>
<quantity>1.0</quantity>
<unit_of_measure></unit_of_measure>
<unit_cost>0.0</unit_cost>
<amount>0.0</amount>
```

Observe that short_description element will automatically get a value "Direct Entry Line Item". This is to indicate that the value was entered directly on cost sheet.

This service can be used on Project (Standard) and Shells of cost code type WBS.

Update Column Data

Description

This method updates data of a project or shells (cost code type WBS) cost sheet column. This method replaces entire data.

Support

This method can only support for project or shell level cost sheet column that is defined as Manual Entry Method. This column does not support Assigned Budget column. This method does not support incremental or delta changes to a column.

Installation: ASP and Self host

Level	Yes or No
Company Level	No
Project and Shell Level	Yes
Program Level	No

Prototype

public XMLObject updateColumnData(String shortname, String authcode, string projectNumber, String ColumnName, String DataXML);

Parameters

Parameter	Description
shortname	identifier of the company, company’s short name
authcode	authentication key for the company, in text string
projectNumber	Identifier of the project or shell in Primavera Unifier.

Parameter	Description
columnName	Identifier of the cost sheet column in Primavera Unifier. This is case sensitive, and should be the name of Column that is defined in Primavera Unifier.
columnXML	identifier for Column content in XML format

Return Value

DataXML Elements

The elements that include as part of DataXML parameter are:

Tag Name	Description
<wbs_code>	WBS Code is a required field.
<short_description>	This is a required field.
<long_description>	This is an optional field
<work_package>	This is an optional field. But use should specify a work package name which is valid under a project or shell. Data entered under this element will be ignored for company level cost column
<spends_category>	This is an optional field. Data sent through this data element should be validated against the data set values
<quantity>	This is an optional field. Data sent through this data element should be numeric. Any other data will be considered as illegal and error should be returned back to user. If user did not send a value for this then "0" will be assumed as value.
<unit_of_measure>	This is an optional field. Data sent through this data element should be validated against the data set values
<unit_cost>	This is an optional field. Data sent through this data element should be numeric. Any other data will be considered as illegal. If user did not send a value for this then "0" will be assumed as value.
<amount>	This is an optional field. User cannot enter any value for this element. Value of this data element should be automatically calculated as (Quantity * Unit Cost)

Project or Shell → Administrator will be used as user who is performing this integration. Audit log will track and trace all changes done through this web services method.

Sample Method

updateColumnData("acme", "acme_authcode", "project_info","my column name", "DataXML")

Sample XML Data

```
<column>
        <wbs_code>1-0-0~~1-1-0~~1-1-1</wbs_code>
        <short_description>Undistributed balance</short_description>
        <long_description></long_description>
        <work_package></work_package>
        <spends_category></spends_category>
        <quantity>1.0</quantity>
        <unit_of_measure></unit_of_measure>
        <unit_cost>100000.0</unit_cost>
        <amount>100000.0</amount>
</column>
<column>
        <wbs_code>5-0-0~~5-1-0</wbs_code>
        <short_description>Testing Again</short_description>
        <long_description></long_description>
        <work_package></work_package>
        <spends_category>Labor</spends_category>
        <quantity>1.0</quantity>
        <unit_of_measure>wk</unit_of_measure>
        <unit_cost>10000.0</unit_cost>
        <amount>10000.0</amount>
</column>
```

Additional Information

Following are the failure reasons and description

Reason	Description
Column is not valid	Invalid Column Name
Column is not manual	Column is not defined as manual entry
Administrator is inactivated	Administrator is inactive. Cannot create column data.

This service can be used for both Project (Standard) and Shells of cost code type WBS.

User Defined Reports (UDR) Methods

Get UDR Data

Description

This method gets XML data from UDR output.

Support

This process is supported under both Company and Project or Shell level reports. Only reports that are marked for Integration are eligible for integration.

Installation: ASP and Self host

Level	Yes or No
Company Level	Yes
Project and Shell Level	Yes

Prototype

public XMLObject getUDRData(String shortname, String authcode, string projectNumber, String reportName);

Parameters

Parameter	Description
shortname	identifier of the company, company's short name
authcode	authentication key for the company, in text string
projectNumber	identifier of the project or shell in Primavera Unifier. If you want to get company level report then pass null for this parameter.
reportName	name of the report user defined report

Return Value

Response will contain following data elements

Tag Name	Description
<report>	Main report tag
<report_header>	Element that carries all report header column names.
<report_row>	Element that carries data of all rows. This element will also show following information **Group By**: If report has group by condition then the data value will show under this element **Subtotal:** If report is configured to show sub-total then the data value will show under this element **Count:** If report is configured to show count then the data value will show under this element.

Sample Method

getUDRData ("acme", "acme_authcode", "project_info", "Funding Report")

Sample XML Data

XML output of an UDR will be based on report design. Following are some samples.

Sample 1 - Tabular UDR with no Group by:

```
<report>
        <report_header>
                <c1>Record No</c1>
                <c2>Send For</c2>
                <c3>Allowed Duration</c3>
                <c4>Workflow Status</c4>
                <c5>Start Date</c5>
        </report_header>
        <report_row>
                <c1>ucli-0001</c1>
                <c2>Approval</c2>
                <c3>14</c3>
                <c4>Pending</c4>
                <c5>10/1/2007</c5>
        </report_row>
        <report_row>
                <c1>ucli-0002</c1>
                <c2>Approval</c2>
                <c3>14</c3>
                <c4>Pending</c4>
                <c5>10/1/2007</c5>
        </report_row>
</report>
```

Observe that report always starts with tag <report>.

<report_header> is the first level child element of <report> element. This element will list all columns defined in the report. Columns for a user defined report are the data element (including formulas) added to Report → Data Elements tab window. This list will also show columns that are marked hidden. Observe that child elements of <report_header> are marked c1, c2, c3...... c1 represents column 1, c2 represents column 2 of the user defined report and are listed in order they are defined in report definition. Value of the element will be the column header defined in report definition.

<report_row> is the child element of <report> element. This element will list data related to all columns that are listed under <report_header>. Number of elements under <report_row> will be same as the number of elements under <report_header>. Observe that child elements of <report_row> are similar to that of <report_header>. C1 representing column 1, c2 representing column 2 etc. Value of the element will be the data retrieved by user defined report.

<report_row> will list each row of report output, so it will be repeated as many times as number of rows retrieved by the report.

Sample 2 - Tabular UDR with Group by:

```
<report>
        <report_header>
                <c1>Record No</c1>
                <c2>Send For</c2>
                <c3>Allowed Duration</c3>
                <c4>Workflow Status</c4>
                <c5>Start Date</c5>
                <c6>Amount</c6>
        </report_header>
        <report_row>
                <c1>Approved</c1>
        </report_row>
        <report_row>
                <c1>ucli-0001</c1>
                <c2>Approval</c2>
                <c3>14</c3>
                <c4>Approved</c4>
                <c5>10/1/2007</c5>
                <c6>1000.00</c6>
        </report_row>
        <report_row>
                <c1>ucli-0002</c1>
                <c2>Approval</c2>
                <c3>14</c3>
                <c4>Approved</c4>
                <c5>10/1/2007</c5>
                <c6>2000.00</c6>
        </report_row>
        <report_row>
                <c1>Pending</c1>
        </report_row>
        <report_row>
                <c1>ucli-0003</c1>
                <c2>Approval</c2>
                <c3>14</c3>
                <c4>Pending</c4>
                <c5>10/1/2007</c5>
                <c6>12345.67</c6>
        </report_row>
</report>
```

Observe that report always starts with tag <report>.

<report_header> is the first level child element of <report> element. This is similar to Sample 1.

<report_row> is the child element of <report> element. Observe that there is only 1 child element (c1). This element will list first group by value. In this example it is "Approved" and "Pending" (marked bold). If report has more than one group by condition then xml output will list that many <report_row> data elements under <report_header>.

Observe that there is another <report_row> showing a list of columns. This element will show data from the rows generated by report output. Number of elements under <report_row> will be same as the number of elements under <report_header>. Observe that child elements of <report_row> are similar to that of <report_header>. C1 representing column 1, c2 representing column 2 etc.., Value of the element will be the data retrieved by user defined report. <report_row> will list each row of report output, so it will be repeated as many times as number of rows retrieved by the report.

If UDR XML integration is used to integrate Primavera Unifier with external systems then it is recommended not to define user defined report with a group by condition.

Sample 3 - Tabular UDR with Sub-Total:

```
<report>
        <report_header>
                <c1>Record No</c1>
                <c2>Send For</c2>
                <c3>Allowed Duration</c3>
                <c4>Workflow Status</c4>
                <c5>Start Date</c5>
                <c6>Amount</c6>
        </report_header>
        <report_row>
                <c1>Approved</c1>
        </report_row>
        <report_row>
                <c1>ucli-0001</c1>
                <c2>Approval</c2>
                <c3>14</c3>
                <c4>Approved</c4>
                <c5>10/1/2007</c5>
                <c6>1000.00</c6>
        </report_row>
        <report_row>
                <c1>ucli-0002</c1>
                <c2>Approval</c2>
                <c3>14</c3>
                <c4>Approved</c4>
                <c5>10/1/2007</c5>
                <c6>2000.00</c6>
        </report_row>
        <report_row>
                <c1>Subtotal:</c1>
                <c6>3000.00</c6>
        </report_row>
        <report_row>
                <c1>Pending</c1>
        </report_row>
        <report_row>
                <c1>ucli-0003</c1>
                <c2>Approval</c2>
                <c3>14</c3>
                <c4>Pending</c4>
                <c5>10/1/2007</c5>
                <c6>12345.67</c6>
        </report_row>
        <report_row>
                <c1>Subtotal:</c1>
                <c6>12345.67</c6>
        </report_row>
</report>
```

Observe that report always starts with tag <report>.

<report_header> is the first level child element of <report> element. This is similar to Sample 1.

<report_row> is the child element of <report> element. Observe that there is only 1 child element (c1). This element will list first group by value. In this example it is "Approved" and "Pending" (marked bold). If report has more than one group by condition then xml output will list that many <report_row> data elements under <report_header>.

Observe that there is another <report_row> showing a list of columns. This element will show data from the rows generated by report output. Number of elements under <report_row> will be same as the number of elements under <report_header>. Observe that child elements of <report_row> are similar to that of <report_header>. C1 representing column 1, c2 representing column 2 etc.., Value of the element will be the data retrieved by user defined report. <report_row> will list each row of report output, so it will be repeated as many times as number of rows retrieved by the report.

Observe that there is another <report_row> showing subtotal. This will show if report is defined to show sub-total.

Sample 4 - Tabular UDR with Count:

This is similar to Sample 5.

Sample 5 - Cross Tab UDR:

User will be allowed to mark a Cross-Tab type report for integration. Following is sample XML format:

```
<report>
        <report_header>
                <c1>Record No</c1>
                <c2>Approved</c2>
                <c3>Pending</c3>
                <c4>Rejected</c4>
                <c5>Closed</c5>
        </report_header>
        <report_row>
                <c1>ucli-0001</c1>
                <c2>123456.78</c2>
                <c3>123456.78</c3>
                <c4>123456.78</c4>
                <c5>123456.78</c5>
        </report_row>
        <report_row>
                <c1>ucli-0002</c1>
                <c2>123456.78</c2>
                <c3>123456.78</c3>
                <c4>123456.78</c4>
                <c5>123456.78</c5>
        </report_row>
</report>
```

<report_header> is the first level child element of <report> element. Column 1 in this scenario will be the header name of 1st Column selected in the report design. Column 2 to Column n will be based on Body Columns (Data Element) that is selected as part of report design.

<report_row> is the child element of <report> element.

This element will be showing data from the rows generated by report output.Number of elements under <report_row> will be same as the number of elements under <report_header>. Observe that child elements of <report_row> are similar to that of <report_header>. C1 representing column 1, c2 representing column 2 etc.., Value of the element will be the data retrieved by user defined report. <report_row> will list each row of report output, so it will be repeated as many times as number of rows retrieved by the report.

Sample 6 - Summary UDR:

Similar to Sample 1 except there will only be 2 columns.

Sample 7 - Alert UDR:

Similar to Sample 1 except there will only be 7 columns and 1 row.

Additional Information

User defined report is always executed upon receiving the message. This ensures that the latest data is returned.

Web Services call does not need any additional permission to execute an UDR that is marked for Integration

If UDR has query parameters defined then the Web Services method will honor that as long as the query parameters are pre-populated during report design

Following are the failure reasons and description

Reason	Description
Report name is not valid. Check if report exists or is enabled for Integration	Invalid Report Name (If report does not exist or is not marked for Integration)
Unable to run report. Contact System Administrator	While running report if there is a system error or for any reason Primavera Unifier cannot run this report

This service can be used for both Project (Standard) and Shells of cost code type WBS and Generic

Asset Manager Methods

Create Asset

Description

This method creates assets under an asset class.

Support

This method can only support creation of assets at company level.

Installation: ASP and Self host

Level	Yes or No
Company Level	Yes
Project and Shell Level	No
Program Level	No

Prototype

public XMLObject createAsset(String shortname, String authcode, string assetClassName, String copyFromAsset, String assetXML);

Parameters

Parameter	Description
shortname	identifier of the company, company's short name
authcode	authentication key for the company, in text string
assetClassName	Name of asset class (Buildings, IT Equipments etc..,)
copyFromAsset	This can be another asset code or a template asset code. User can provide a null value if user does not want to copy from an asset or template. If user provides a value, system should first look under templates and if template does not exists then it should check under real assets
assetXML	Content of the asset that needs to be created. This is based on Integration interface design for asset class in uDesigner.

Return Value

XMLObject (XMLObject is described in Appendix D of this document)

Sample Method

createAsset("acme", "acme_authcode", "Buildings","Buildings Temp 1","XML")

Sample XML Return Data

```
<List_Wrapper>
   <_bp>
      <uuu_asset_name />
      <uuu_asset_acquisition_cost />
      <uuu_asset_acquisition_date />
      <uuu_asset_salvage_value />
      <uuu_asset_depreciation_prd />
      <status />
  </_bp>
  </List_wrapper>
```

Additional Information

Same logic while creating new assets from UI will be used while creating an asset from integration. This includes validation logic, setting status of an asset, depreciation definitions etc..,

If user tries to create an asset without copying from another asset or template, then new asset will be created based on the data provided as part of web services payload. Note: No depreciation setup will be available for these newly created assets as payload does **not take any depreciation setup information.**

If user tries to create an asset by copying from another asset or template then new asset will be created based on the data provided as part of web services payload. Template or asset name provided as part of web services call will be used to copy depreciation setup only (depreciation data will not be copied). That means only deprecation information will be copied from the template or asset and not any data on the detail form of template.

All validations on asset detail form including required fields and user defined form validations will be honored while creating new assets through web services.

If depreciation is available for newly created assets then the depreciation schedule will be calculated, asset detail form will be updated as part of web services call.

Once asset is created, asset detail form will be in Finish Editing mode and data including depreciation information will be rolled up to asset class sheets, asset summary sheet and accounts sheet.

Data Element which is based on data definition viz.., SYS Numeric Logical Datasource, SYS Date Logical Datasource, SYS Business Process Datasource and SYS Project Cost Datasource will not be available as part of Integration.

Following data element are only available as part of createAsset

Asset Code and Asset Navigation Code: These two data elements are built automatically based on the definition

Net Book Value, Cumulative Depreciation, Current Period Depreciation and Calculations as of: Calculated based on asset depreciation setup

Schedule Manager Methods

Create Schedule Sheet Activities from Primavera P5 and P6 XML

Description

This method creates activities into an existing Primavera Unifier project or shell (cost code type WBS) level schedule sheet using default data mapping.

Support

This method can only support creation of activities in project or shell (cost code type WBS) level schedule sheets.

Installation: ASP and Self host

Level	Yes or No
Company Level	No
Project and Shell Level	Yes
Program Level	No

Prototype

public XMLObject createScheduleActivities(String shortname, String authcode, String projectNumber, String scheduleSheet, String scheduleXML);

Parameters

Parameter	Description
Shortname	Identifier of the company, company's short name
authcode	Authentication code for the company, in text string
projectNumber	Identifier of project or shell in Primavera Unifier.
scheduleSheet	Name of Schedule Sheet to import activities into.
scheduleXML	Content of activities to import. This is obtained from the Primavera P5/P6 xml content. The Primavera XML content should be generated by exporting the full Primavera project.

Return Value

XMLObject (XMLObject is described in Appendix D of this document)

Sample Method

createScheduleActivities("acme","acme_authcode","Proj-0001","Schedule Sheet 001", "<XML Content>");

Sample XML Data

```
<list_wrapper>
-- complete xml of primavera P5 or P6 (starting with <BusinessObjects>)
</list_wrapper>
```

Additional Information

You must perform appropriate data mapping setups on the target schedule sheet. This includes mapping Primavera Unifier Schedule Sheet columns to Primavera XML elements, and setting appropriate XML Import options.   Only elements that are mapped will be copied.

This method can only be used to create new activities not for updating existing activities.

If invoked on an empty schedule sheet, this method will insert activities into the schedule sheet. If invoked on a non-empty schedule sheet, it will overwrite all existing activities (delete and recreate) in the non-empty schedule sheet if this function is allowed in the data mapping setup.

If the method is called on a non-empty schedule sheet and the data mapping setup does not allow overwrite, then the method will return an error in the response.

This service can be used for both Project (Standard) and Shells of cost code type WBS.

Activity identified with this creation method will be GUID.

Data Mapping Marked as default will be used.

Create Schedule Sheet Activities V2 from Oracle® Primavera P6™ XML

Description

This method creates activities into an existing Primavera Unifier project or shell (cost code type WBS) level schedule sheet based on the data mapping parameter and schedule options provided as part of Web Service call.

Support

This method can only support creation of activities in project or shell (cost code type WBS) level schedule sheets.

Installation: ASP and Self host

Level	Yes or No
Company Level	No
Project and Shell Level	Yes

Prototype

public XMLObject createScheduleActivitiesV2(String shortname, String authcode, String projectNumber, String sheetName, String sheetXML, String scheduleOptions);

Parameters

Parameter	Description
Shortname	Identifier of the company, company’s short name
Authcode	Authentication code for the company, in text string
projectNumber	Identifier of project or shell in Primavera Unifier.
sheetName	Name of Schedule Sheet to import activities into.
sheetXML	Content of activities to import. This is obtained from the Primavera P5/P6 xml content. The Primavera XML content should be generated by exporting the full Primavera project.
scheduleOptions	This element will allow message to carry optional information. Following are the tags under this element ▸ <MapName/> ▸ <ActivityIdentifier/> ▸ <WBS_NoOfLevelsToMerge/> ▸ <WBS_StartMergeFrom/>
MapName	Name of the data mapping that should be used while processing XML file. If there is no map name then the one marked "Default" in Primavera Unifier on the schedule sheet will be used.
ActivityIdentifier	Possible values are GUID, ID. ID is the tag of Activity Id in Primavera. This tag will be used if you want to update an activity through udpateScheduleActivities webservice call. If Id tag is used for create and update then the value provided under Id tag will be used as task / activity identifier.
WBS_NoOfLevelsToMerge	This parameter will take a numeric value. This parameter can be used by user to specify how many levels (WBS Levels) in P6 XML file should be merged together. EXAMPLE: Assume that P6 has following structure (5.ON.DA.DA03.F350.U11.095.ABC20.CAct04) If you specify 3 then two levels will be merged together. But which 3 segments will be based on the next parameter (wbs_startmergefrom). Value of this parameter can be 0.
WBS_StartMergeFrom	This parameter will take a numeric value. Value of this parameter will be used to determine the level from which this merge should start. EXAMPLE: In the above example, if a user specifies 2, then system will start from level 2 and merge ON, DA and DA03. So after the merge the WBS Code build by import program will be 5.ONDADA03.F350.U11.095.ABC20.CAct04. Value of this parameter can be 0.

Return Value

XMLObject (XMLObject is described in Appendix D of this document)

Sample Method

createScheduleActivitiesV2("acme","acme_authcode","Proj-0001","Schedule Sheet 001", "<XML Content>","<scheduleoptions>");

Sample XML Data

```
<ulink>
        <_shortname>acme</_shortname>
        <_authcode>acme_authcode</_authcode>
        <_servicename>createScheduleActivitiesV2</_servicename>
        <_projectNumber>Prj-0001</_projectNumber>
        <_sheetname> Schedule Sheet 001</_sheetname>
        <scheduleOptions>
                <MapName>Create</MapName>
                <ActivityIdentifier>Id</ActivityIdentifier>
                <WBS_StartMergeFrom>Merge start number</WBS_StartMergeFrom>
                <WBS_NoOfLevelsToMerge>No of levels
number</WBS_NoOfLevelsToMerge>
        </scheduleOptions>
        <List_Wrapper>
                <APIBusinessObjects
xmlns="http://xmlns.oracle.com/Primavera/P6/V7/API/BusinessObjects"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.oracle.com/Primavera/P6/V7/API/BusinessObjects
http://xmlns.oracle.com/Primavera/P6/V7/API/p6apibo.xsd">
<!—XML tags from P6 ->
                              </APIBusinessObjects>
               </List_Wrapper>
</ulink>
```

Additional Information

You must perform appropriate data mapping setups on the target schedule sheet. This includes mapping Primavera Unifier Schedule Sheet columns to Primavera XML elements, and setting appropriate XML Import options.   Only elements that are mapped will be copied.

This method can only be used to create new activities not for updating existing activities.

If invoked on an empty schedule sheet, this method will insert activities into the schedule sheet. If invoked on a non-empty schedule sheet, it will overwrite all existing activities (delete and recreate) in the non-empty schedule sheet if this function is allowed in the data mapping setup.

If the method is called on a non-empty schedule sheet and the data mapping setup does not allow overwrite, then the method will return an error in the response.

This service can be used for both Project (Standard) and Shells of cost code type WBS.

- Activity identified with this creation method can be GUID or ID
- WBS Code will be derived automatically based on the XML file. For more information refer to Appendix A.

Create Schedule Activities from file V2

Description

This option will allow user to process a P6 XML file and add activities and resources to a Primavera Unifier schedule sheet. Along with activity information this process will also take activity spread information. Since the activity spread information data can be a lot, this method allows user to send in information in a .zip file.

Support

Installation: ASP and Self host

Level	Yes or No
Company Level	No
Project and Shell Level	Yes

Prototype

public XMLObject createScheduleActivitiesFromFileV2(String shortname, String authcode, String projectNumber, String sheetname, String scheduleoptions, String iszipfile, FileObject[] files);

Parameters

Parameter	Description
Shortname	identifier of the company, company's short name
Authcode	authentication key for the company, in text string
projectNumber	Identifier of the project or shell in Primavera Unifier.
sheetName	Identify the name of the schedule sheet in Primavera Unifier
iszipfile	This tag identifies if the file provided under files tag is a zip file or .xml file with just activity and resource information. If user wants to send activity spread information then the value of this tag should be yes and the file provided under files tag should be a zip file containing activity/ resource information and the spread information.
files	FileObject containing the Activity / Resource information and the activity spread information. FileObject is made up of two components String filename (this should be the name of the zip file) and the zip file with two files, one for activity / resource information and one with activity spread information. Name of the activity/ resource information should be same as the one given under <scheduleoptions>.<ActivityFileName>tag and the name of the name of the activity spread information should be same as the one given under <scheduleoptions>.<SpreadFileName>.

Parameter	Description
scheduleOptions	This element will allow message to carry optional information. Following are the tags under this element <MapName/> <ActivityIdentifier/> <WBS_NoOfLevelsToMerge/> <WBS_StartMergeFrom/> <ActivityFileName/> <SpreadFileName/>
MapName	Name of the data mapping that should be used while processing XML file
ActivityIdentifier	Possible values are GUID, ID. ID is the tag of Activity Id in Primavera.
WBS_NoOfLevelsToMerge	This parameter will take a numeric value. This parameter can be used by user to specify how many levels (WBS Levels) in P6 XML file should be merged together. ▸ EXAMPLE: Assume that P6 has following structure (5.ON.DA.DA03.F350.U11.095.ABC20.CAct04) If you specify 3 then two levels will be merged together. But which 3 segments will be based on the next parameter (wbs_startmergefrom)
WBS_StartMergeFrom	▸ This parameter will take a numeric value. Value of this parameter will be used to determine the level from which this merge should start. ▸ EXAMPLE: In the above example, if a user specifies 2, then system will start from level 2 and merge ON, DA and DA03. So after the merge the WBS Code build by import program will be 5.ONDADA03.F350.U11.095.ABC20.CAct04
ActivityFileName	Name of the file that contains the activity and resource information. Please refer to files tag
SpreadFileName	Name of the activity spread file name. Please refer to files tag

Return Value

XMLObject (XMLObject is described in Appendix D of this document)

Sample Method

createScheduleActivitiesV2("acme","acme_authcode","Proj-0001","Schedule Sheet 001", "<XML Content>","yes","<scheduleoptions>",files);

Sample XML Data

```
<ulink>
        <_shortname>acme</_shortname>
        <_authcode>acme_authcode</_authcode>
        <_servicename>createScheduleActivitiesFromFileV2</_servicename>
        <_projectNumber>Prj-001</_projectNumber>
        <_sheetname>Schedule Sheet 001</_sheetname>
        <_zipfile>P0165-10TEST.zip</_zipfile>
        <_iszipfile>yes</_iszipfile>
        <scheduleOptions>
                <MapName>map name</MapName>
                <ActivityIdentifier>Id</ActivityIdentifier>
                <WBS_StartMergeFrom>start merge from number</WBS_StartMergeFrom>
                <WBS_NoOfLevelsToMerge> levels to merge
num</WBS_NoOfLevelsToMerge>
                <ActivityDeletion>Confirm</ActivityDeletion>
                <ActivityFileName>activity.xml</ActivityFileName>
                <SpreadFileName>spread.xml</SpreadFileName>
        </scheduleOptions>
<List_Wrapper>
</List_Wrapper>
</ulink>
```

Sample Spread Data

```
<APIBusinessObjects xmlns="http://xmlns.oracle.com/Primavera/P6/V7/API/BusinessObjects"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.oracle.com/Primavera/P6/V7/API/BusinessObjects
http://xmlns.oracle.com/Primavera/P6/V7/API/p6apibo.xsd">
<a>
        <id>A100</id>
        <sp>
        <sd>2011-04-04T00:00:00</sd>
        <ed>2012-02-23T17:00:00</ed>
        <pt>DAY</pt>
        <pd>
        <sd>2011-05-30T00:00:00</sd>
        <ed>2011-05-31T00:00:00</ed>
        <amt>5555.555555555556</amt>
        </pd>
<pd>
        <sd>2011-05-31T00:00:00</sd>
        <ed>2011-06-01T00:00:00</ed>
        <amt>11111.111111111111</amt>
        </pd>
        <pd>
        <sd>2011-06-01T00:00:00</sd>
        <ed>2011-06-02T00:00:00</ed>
        <amt>16666.666666666668</amt>
        </pd>
```

Additional Information

Format of spread file should be similar to Sample Spread Data example.

While processing spread information, Primavera Unifier will only look for <amt> tag.

If the Import from external source option is checked in Primavera Unifier and spread information is not available in XML file then system will error out. This error condition is also valid when this tag is not available in the XML file.

Activity File should start and end with <List_Wrapper> tag

Spread file should not contain any <List_Wrapper> tag

Update Schedule Sheet Activities from Oracle® Primavera™ P5 and P6 XML

Description

This method will insert/update activities into any project or shell schedule sheet. It will not overwrite existing activities in the schedule sheet.

Support

This method can only update activities in project or shell level schedule sheets.

Installation: ASP and Self host

Level	Yes or No
Company Level	No
Project or Shell Level	Yes

Prototype

public XMLObject updateScheduleActivities(String shortname, String authcode, String projectNumber, String sheetName, String sheetXML);

Parameters

Parameter	Description
shortname	Identifier of the company, company's short name
authcode	Authentication code for the company, in text string
projectNumber	Identifier of project or shell in Primavera Unifier.
sheetName	Name of Schedule Sheet to import activities into.
sheetXML	Content of activities to import. This is obtained from the Primavera P5/P6 xml content. The Primavera XML content should be generated by exporting the full Primavera project.

Return Value

XMLObject (XMLObject is described in Appendix D of this document)

Sample Method

updateScheduleActivities("acme","acme_authcode","Proj-0001","Schedule Sheet 001", "<XML Content>");

Sample XML Data

```
<list_wrapper>
-- complete xml of primavera P5 or P6 (starting with <BusinessObjects>)
</list_wrapper>
```

Additional Information

You must perform appropriate data mapping setups on the target schedule sheet. This includes mapping Primavera Unifier Schedule Sheet columns to Primavera XML elements, and setting appropriate XML Import options.

This method will insert/update activities into any project or shell schedule sheet. It will not overwrite existing activities in the schedule sheet. The method will utilize the data mapping setup to update only those elements that are mapped.

The method errors out if the data mapping option does not allow merge.

Activities are uniquely identified within Primavera by the Globally Unique Id (<GUID>). The GUID is assumed to be unique not just within Primavera but across all products including Primavera Unifier.

Resource assignments will be imported (inserted/updated) if the ‘Import Resource Assignments’ checkbox is selected. Resource assignments will be matched by Resource Name.

This service can be used for both Project (Standard) and Shells of cost code type WBS.

Update Schedule Sheet Activities V2 from Oracle® Primavera™ P6 XML

Description

This method will insert/update activities into any project or shell schedule sheet. It will not overwrite existing activities in the schedule sheet based on the data mapping parameter.

Support

This method can only support updation of activities in project or shell level schedule sheets.

Installation: ASP and Self host

Level	Yes or No
Company Level	No
Project or Shell Level	Yes

Prototype

public XMLObject updateScheduleActivitiesV2(String shortname, String authcode, String projectNumber, String sheetName, String sheetXML, String scheduleOptions);

Parameters

Parameter	Description
shortname	Identifier of the company, company's short name
authcode	Authentication code for the company, in text string
projectNumber	Identifier of project or shell in Primavera Unifier.
sheetname	Name of Schedule Sheet to import activities into.
sheetXML	Content of activities to import. This is obtained from the Primavera P5/P6 xml content. The Primavera XML content should be generated by exporting the full Primavera project.
scheduleOptions	This element will allow message to carry optional information. Following are the tags under this element <MapName/> <ActivityIdentifier/> <WBS_NoOfLevelsToMerge/> <WBS_StartMergeFrom/> <ActivityDeletion/>
MapName	Name of the data mapping that should be used while processing XML file
ActivityIdentifier	Possible values are GUID, ID. ID is the tag of Activity Id in Primavera.
WBS_NoOfLevelsToMerge	This parameter will take a numeric value. This parameter can be used by user to specify how many levels (WBS Levels) in P6 XML file should be merged together. - EXAMPLE: Assume that P6 has following structure (5.ON.DA.DA03.F350.U11.095.ABC20.CAct04) If you specify 3 then two levels will be merged together. But which 3 segments will be based on the next parameter (wbs_startmergefrom)
WBS_StartMergeFrom	- This parameter will take a numeric value. Value of this parameter will be used to determine the level from which this merge should start. - EXAMPLE: In the above example, if a user specifies 2, then system will start from level 2 and merge ON, DA and DA03. So after the merge the WBS Code build by import program will be 5.ONDADA03.F350.U11.095.ABC20.CAct04

Parameter	Description
ActivityDeletion	Indicate to delete activities from schedule sheet in Primavera Unifier if it is deleted in P6.

Return Value

XMLObject (XMLObject is described in Appendix D of this document)

Sample Method

updateScheduleActivitiesV2("acme","acme_authcode","Proj-0001","Schedule Sheet 001", "<XML Content>","Data Mapping 001","Id");

Sample XML Data

```
<ulink>
            <_shortname>acme</_shortname>
            <_authcode>acme_authcode</_authcode>
            <_servicename>updateScheduleActivitiesV2</_servicename>
            <_projectNumber>Prj-0001</_projectNumber>
            <_sheetname> Schedule Sheet 001</_sheetname>
            <scheduleOptions>
                        <MapName>Update</MapName>
                        <ActivityIdentifier>Id</ActivityIdentifier>
                        <WBS_StartMergeFrom>Merge start number</WBS_StartMergeFrom>
                        <WBS_NoOfLevelsToMerge>No of levels
number</WBS_NoOfLevelsToMerge>
            </scheduleOptions>
            <List_Wrapper>
                        <APIBusinessObjects
xmlns="http://xmlns.oracle.com/Primavera/P6/V7/API/BusinessObjects"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.oracle.com/Primavera/P6/V7/API/BusinessObjects
http://xmlns.oracle.com/Primavera/P6/V7/API/p6apibo.xsd">
                                             </APIBusinessObjects>
                      </List_Wrapper>
</ulink>
```

Additional Information

You must perform appropriate data mapping setups on the target schedule sheet. This includes mapping Primavera Unifier Schedule Sheet columns to Primavera XML elements, and setting appropriate XML Import options.

This method will insert/update activities into any project or shell schedule sheet. The method will utilize the data mapping setup to update only those elements that are mapped.

The method errors out if the data mapping option does not allow merge.

Activities are uniquely identified within Primavera by the Globally Unique Id (<GUID>) or the identified provided as part of XML file. GUID will be used if ActivityIdentifier is empty or blank.

Resource assignments will be imported (inserted/updated) if the ‘Import Resource Assignments’ checkbox is selected. Resource assignments will be matched by Resource Name.

This service can be used for both Project (Standard) and Shells of cost code type WBS.

- Activity identified with this creation method can be GUID or ID
- On the options tab of Data Mapping, merge existing schedule option should be checked.
- WBS Code will be derived automatically based on the XML file. For more information refer to Appendix A.
- ActivityDeletion tag logic is based on the "Delete Activity" checkbox available on data mapping window in Primavera Unifier. Following is the logic for this tag

Delete Checkbox	**Activity Deletion**	**Behavior**
Unchecked	Auto or Confirm	Activity Deletion tag is ignored. Activities that are removed under source schedule will not be removed from Primavera Unifier Schedule Sheet
Checked	Auto	Activity Deletion tag is considered Activities that are removed under source schedule will be deleted automatically under Primavera Unifier Schedule Sheet.
Checked	Confirm	Activity Deletion tag is considered Activities that are removed under source schedule will not be deleted. System will return an error and will not update Primavera Unifier schedule sheet. If no activities are removed under source schedule then system will update Primavera Unifier Schedule Sheet.

ActivityDeletion tags are only considered for updateScheduleSheetFromFileV2 service and not for create.

Update Schedule Activities From File V2

Description

This option will allow user to process a P6 XML file and add and update activities and resources to a Primavera Unifier schedule sheet. Along with activity information this process will also take activity spread information.

Support

Installation: ASP and Self host

Level	Yes or No
Company Level	No
Project and Shell Level	Yes

Prototype

public XMLObject updateScheduleActivitiesFromFileV2(String shortname, String authcode, String projectNumber, String sheetname, String scheduleoptions, String iszipfile, FileObject[] files);

Parameters

Parameter	Description
Shortname	identifier of the company, company's short name
Authcode	authentication key for the company, in text string
projectNumber	Identifier of the project or shell in Primavera Unifier.
sheetName	Identify the name of the schedule sheet in Primavera Unifier
iszipfile	This tag identifies if the file provided under files tag is a zip file or .xml file with just activity and resource information. If user wants to send activity spread information then the value of this tag should be yes and the file provided under files tag should be a zip file containing both activity and resource information and the spread information.
files	FileObject containing the Activity / Resource information and the activity spread information. FileObject is made up of two components String filename (this should be the name of the zip file) and the zip file with two files, one for activity / resource information and one with activity spread information. Name of the activity/ resource information should be same as the one given under <scheduleoptions>.<ActivityFileName>tag and the name of the name of the activity spread information should be same as the one given under <scheduleoptions>.<SpreadFileName>.

Parameter	Description
scheduleOptions	This element will allow message to carry optional information. Following are the tags under this element <MapName/> <ActivityIdentifier/> <WBS_NoOfLevelsToMerge/> <WBS_StartMergeFrom/> <ActivityDeletion> <ActivityFileName> <SpreadFileName>
MapName	Name of the data mapping that should be used while processing XML file
ActivityIdentifier	Possible values are GUID, ID. ID is the tag of Activity Id in Primavera.
WBS_NoOfLevelsToMerge	This parameter will take a numeric value. This parameter can be used by user to specify how many levels (WBS Levels) in P6 XML file should be merged together. ▸ EXAMPLE: Assume that P6 has following structure (5.ON.DA.DA03.F350.U11.095.ABC20.CAct04) If you specify 3 then two levels will be merged together. But which 3 segments will be based on the next parameter (wbs_startmergefrom)
WBS_StartMergeFrom	▸ This parameter will take a numeric value. Value of this parameter will be used to determine the level from which this merge should start. ▸ EXAMPLE: In the above example, if a user specifies 2, then system will start from level 2 and merge ON, DA and DA03. So after the merge the WBS Code build by import program will be 5.ONDADA03.F350.U11.095.ABC20.CAct04
ActivityDeletion	This tag will allow user to determine if the activities that does not exists in P6 should be deleted from Primavera Unifier schedule sheet. Possible values are Auto / Confirm.
ActivityFileName	Name of the file that contains the activity and resource information. Please refer to files tag
SpreadFileName	Name of the activity spread file name. Please refer to files tag

Return Value

XMLObject (XMLObject is described in Appendix D of this document)

Sample Method

updateScheduleActivitiesFromFileV2 ("acme","acme_authcode","Proj-0001","Schedule Sheet 001", "<XML Content>","yes","<scheduleoptions>",files);

Sample XML Data

```
<ulink>
        <_shortname>acme </_shortname>
        <_authcode>acme_authcode</_authcode>
        <_servicename>updateScheduleActivitiesFromFileV2</_servicename>
        <_projectNumber>Prj-001</_projectNumber>
        <_sheetname>Schedule Sheet 001</_sheetname>
        <_zipfile>zipfile.zip</_zipfile>
        <_iszipfile>yes</_iszipfile>
        <scheduleOptions>
                <MapName>Mapping</MapName>
                <ActivityIdentifier>Id</ActivityIdentifier>
                <WBS_StartMergeFrom>2</WBS_StartMergeFrom>
                <WBS_NoOfLevelsToMerge>3</WBS_NoOfLevelsToMerge>
                <ActivityDeletion>Confirm</ActivityDeletion>
                <ActivityFileName>Activity.xml</ActivityFileName>
                <SpreadFileName>Spread.xml</SpreadFileName>
        </scheduleOptions>
<List_Wrapper>
--- P6 XML file
</List_Wrapper>
</ulink>
```

Sample Spread Data

```
<APIBusinessObjects xmlns="http://xmlns.oracle.com/Primavera/P6/V7/API/BusinessObjects"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.oracle.com/Primavera/P6/V7/API/BusinessObjects
http://xmlns.oracle.com/Primavera/P6/V7/API/p6apibo.xsd">
<a>
        <id>A100</id>
        <sp>
        <sd>2011-04-04T00:00:00</sd>
        <ed>2012-02-23T17:00:00</ed>
        <pt>DAY</pt>
        <pd>
        <sd>2011-05-30T00:00:00</sd>
        <ed>2011-05-31T00:00:00</ed>
        <amt>5555.555555555556</amt>
        </pd>
<pd>
        <sd>2011-05-31T00:00:00</sd>
        <ed>2011-06-01T00:00:00</ed>
        <amt>11111.111111111111</amt>
        </pd>
        <pd>
        <sd>2011-06-01T00:00:00</sd>
        <ed>2011-06-02T00:00:00</ed>
        <amt>16666.666666666668</amt>
        </pd>
```

Additional Information

ActivityDeletion tag logic is based on the "Delete Activity" checkbox available on data mapping window in Primavera Unifier. Following is the logic for this tag

Delete Checkbox	Activity Deletion	Behavior
Unchecked	Auto or Confirm	Activity Deletion tag is ignored. Activities that are removed under source schedule will not be removed from Primavera Unifier Schedule Sheet
Checked	Auto	Activity Deletion tag is considered Activities that are removed under source schedule will be deleted automatically under Primavera Unifier Schedule Sheet.
Checked	Confirm	Activity Deletion tag is considered Activities that are removed under source schedule will not be deleted. System will return an error and will not update Primavera Unifier schedule sheet.

		If no activities are removed under source schedule then system will update Primavera Unifier Schedule Sheet.

ActivityDeletion tags are only considered for updateScheduleSheetFromFileV2 service and not for create.

While processing spread information, Primavera Unifier will only look for "" tag.

If the Import from external source option is checked in Primavera Unifier and spread information is not available in XML file then system will error out. This error condition is also valid when this tag is not available in the XML file.

Activity File should start and end with <List_Wrapper> tag

Spread file should not contain any <List_Wrapper> tag

Get Schedule Sheet Activities from Primavera Unifier

Description

This method will extract activities from any project or shell (cost code of type WBS) schedule sheet.

Support

This method can only extract activities from project or shell level schedule sheets.

Installation: ASP and Self host

Level	Yes or No
Company Level	No
Project and Shell Level	Yes
Program Level	No

Prototype

public XMLObject getScheduleActivities(String shortname, String authcode, String projectNumber, String scheduleSheet);

Parameters

Parameter	Description
Shortname	Identifier of the company, company's short name
Authcode	Authentication code for the company, in text string
projectNumber	Identifier of project or shell in Primavera Unifier.
scheduleSheet	Name of Schedule Sheet to import activities into.

Return Value

XMLObject (XMLObject is described in Appendix D of this document)

Sample Method

getScheduleActivities("acme", "acme_authcode", "Proj-0001","Schedule Sheet 001");

Sample Return XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="list_wrapper">
        <xs:complexType>
            <xs:sequence>
                <xs:element maxOccurs="1" name="ScheduleProperties">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element   maxOccurs="1" name="Name" type="xs:string" />
                            <xs:element maxOccurs="1"   name="Description" />
                            <xs:element maxOccurs="1"   name="Company" type="xs:string" />
                            <xs:element maxOccurs="1"   name="Author" type="xs:string" />
                            <xs:element   maxOccurs="1" name="CreationDate"
type="xs:string" />
                            <xs:element maxOccurs="1"   name="LastSaved" type="xs:string" />
                            <xs:element   maxOccurs="1" name="StartDate" type="xs:string" />
                            <xs:element   maxOccurs="1" name="FinishDate" type="xs:string" />
                            <xs:element   maxOccurs="1" name="MainSheet"
type="xs:boolean" />
                            <xs:element   maxOccurs="1" name="Status" type="xs:string" />
                            <xs:element   maxOccurs="1" name="SheetLock" type="xs:boolean"
/>
                            <xs:element   maxOccurs="1" name="ColumnTop" type="xs:string"
/>
                            <xs:element   maxOccurs="1" name="ColumnLeft" type="xs:string"
/>
                            <xs:element   maxOccurs="1" name="ColumnRight" type="xs:string"
/>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
                <xs:element maxOccurs="1"   name="Activities">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element maxOccurs="unbounded" name="Activity">
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:element name="description" />
                                        <xs:element name="ActivityName" type="xs:string" />
                                        <xs:element name="ActivityId" type="xs:positiveInteger"
/>
                                        <xs:element name="Start" type="xs:string" />
                                        <xs:element name="Finish" type="xs:string" />
                                        <xs:element name="Duration" type="xs:decimal" />
                                        <xs:element name="PercentComplete"
type="xs:decimal" />
                                        <xs:element name="FlagMilestone"
type="xs:positiveInteger" />
                                        <xs:element name="FlagComplete"
type="xs:positiveInteger" />
                                        <xs:element name="Float" type="xs:decimal" />
                                        <xs:element name="FixedCost" type="xs:decimal" />
                                        <xs:element name="LaborCost" type="xs:decimal" />
                                        <xs:element name="NonLaborCost" type="xs:decimal"
/>
                                        <xs:element name="TotalCost" type="xs:decimal" />
                                        <xs:element name="ActivityWorkHours"
type="xs:decimal" />
                                        <xs:element name="IgnoreResourceCalculation"
type="xs:positiveInteger" />
                                        <xs:element name="OutlineLevel"
type="xs:unsignedByte" />
                                        <xs:element name="SummaryRow"
type="xs:unsignedByte" />
                                        <xs:element name="Lastmodified" type="xs:string" />
                                        <xs:element name="NotifyRescource"
type="xs:boolean" />
                                        <xs:element name="EarlyStart" type="xs:string" />
                                        <xs:element name="EarlyFinish" type="xs:string" />
                                        <xs:element name="LateStart" type="xs:string" />
                                        <xs:element name="LateFinish" type="xs:string" />
                                        <xs:element name="FlagUpdateActivityPercent"
type="xs:positiveInteger" />
                                        <xs:element name="Guid" type="xs:string" />
                                        <xs:element name="ParentGuid" type="xs:string" />
                                    </xs:sequence>
                                </xs:complexType>
                            </xs:element>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
                <xs:element maxOccurs="1"   name="WBSItems">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element maxOccurs="unbounded" name="WBSItem">
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:element name="ActivityGUID" />
                                        <xs:element name="ActivityID" type="xs:positiveInteger"
/>
                                        <xs:element name="SplitPercent" type="xs:decimal" />
                                        <xs:element name="WBSCode" type="xs:string" />
                                        <xs:element name="WBSItem" type="xs:string" />
                                    </xs:sequence>
                                </xs:complexType>
                            </xs:element>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
                <xs:element   maxOccurs="1" name="Predecessors">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="Predecessor">
                                <xs:complexType>
                                    <xs:sequence>
```

```
                                        <xs:element name="ActivityGUID" />
                                        <xs:element name="ActivityID" type="xs:positiveInteger"
/>
                                        <xs:element name="PredecessorGUID" />
                                        <xs:element name="PredecessorActivityID"
type="xs:positiveInteger" />
                                        <xs:element name="Lag" type="xs:decimal" />
                                        <xs:element name="Type" type="xs:string" />
                                    </xs:sequence>
                                </xs:complexType>
                            </xs:element>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
                <xs:element   maxOccurs="1" name="ResourceAssignments">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element maxOccurs="unbounded"
name="ResourceAssignment">
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:element name="ActivityGUID" />
                                        <xs:element name="ActivityID" type="xs:positiveInteger"
/>
                                        <xs:element name="RoleName" type="xs:string" />
                                        <xs:element name="ResourceName" type="xs:string" />
                                        <xs:element name="ResourceCode" type="xs:string" />
                                    <xs:element name="ResourceCapacity"
type="xs:positiveInteger" />
                                        <xs:element name="WorkHours" type="xs:decimal" />
                                        <xs:element name="WorkHoursPercent"
type="xs:decimal" />
                                        <xs:element name="PercentComplete"
type="xs:decimal" />
                                    </xs:sequence>
                                </xs:complexType>
                            </xs:element>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

Additional Information

The XML return object contains a Primavera Unifier specific format. You will need to write additional integration logic to convert it to other formats.

This service can be used for both Project (Standard) and Shells of cost code type WBS.

Get a List of Project Schedule Sheets

Description

This method will return a list of Schedule Sheets within a Project or Shell.

Support

This method can only list Schedule Sheets from projects or shells.

Installation: ASP and Self host

Level	Yes or No
Company Level	No
Project and Shell Level	Yes
Program Level	No

Prototype

public XMLObject getScheduleSheetList(String shortname, String authcode, String projectnumber, String options);

Parameters

Parameter	Description
shortname	Identifier of the company, company's short name
authcode	Authentication code for the company, in text string
projectnumber	Identifier of project or shell in Unifier.
options	This parameter allows the user to filter the list based on the following: <status>: select a status of Active or Inactive. If no status is provided then all sheets are returned. <main_sheet>: Set to true or false in order to specifiy if the master sheet is to only be returned. Setting true will return only the master schedule. <filter_condition>: two filter conditions are available; name and sheet_lock. The name is the name of the sheet and sheet_lock can have a value of true or false in order to only return locked schedule sheets.

Return Value

XMLObject (XMLObject is described in Appendix D of this document). Message will return list of Schedule Sheets based on filter condition

Sample Method

getScheduleSheetList("acme", "acme_authcode", "P-000001", status=Active);

Sample XML Input

```
<ulink>
        <_shortname>acme</_shortname>
        <_authcode>acme_authcode</_authcode>
        <_servicename> getScheduleSheetList </_servicename>
                <_projectnumber>proj0001</_projectnumber>
        <options>
                         <status>Active</status>
                         <main_sheet>false</main_sheet>
                 <filtercondition> </filtercondition>
                </options>
</ulink>
```

Sample XML Output

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<List_Wrapper>
        <schedulesheet>
           <name>Schedule Sh1</name>
           <creationdate>02-12-2012 17:00:00</creationdate>
           <status>Active</status>
           <sheet_lock>Unlocked</sheet_lock>
           <main_sheet>Yes</main_sheet>
        </schedulesheet>
        <schedulesheet>
           <name>Schedule Sh2</name>
           <creationdate>02-12-2012 17:00:00</creationdate>
           <status>Active</status>
           <sheet_lock>Locked</sheet_lock>
           <main_sheet>No</main_sheet>
        </schedulesheet>
</List_Wrapper>
```

Get a List of Schedule Sheet Data Mappings

Description

This method will return a list of Data Mappings configured for a Schedule Sheet within a Project or Shell.

Support

This method can only list Data Mappings from Schedule Sheets within projects or shells.

Installation: ASP and Self host

Level	Yes or No
Company Level	No
Project and Shell Level	Yes
Program Level	No

Prototype

public XMLObject getScheduleSheetDataMaps(String shortname, String authcode, String projectnumber, String sheetname, String options);

Parameters

Parameter	Description
shortname	Identifier of the company, company's short name
authcode	Authentication code for the company, in text string
projectnumber	Identifier of project or shell in Unifier.
sheetname	Identifier of the schedule sheet within the project or shell in Unifier.
options	This parameter allows the user to filter the list based on the following: <datamap_name>: The name of a specific data mapping to be retrieved. <filter_condition>: one filter conditions is available; isdefault. If set to true only the data mapping flagged as the default will be returned.

Return Value

XMLObject (XMLObject is described in Appendix D of this document). Message will return list of Schedule Sheets based on filter condition

Sample Method

getScheduleSheetDataMaps ("acme", "acme_authcode", "P-00001", "Master Schedule", isdefault=true);

Sample XML Input

```
<ulink>
          <_shortname>acme</_shortname>
          <_authcode>acme_authcode</_authcode>
          <_servicename> getScheduleSheetList </_servicename>
                    <_projectnumber>P-00001</_projectnumber>
                    <_sheetname>Master Schedule</_sheetname>
          <options>
                               <datamap_name> </ datamap_name >
                     <filtercondition> </filtercondition>
                    </options>
</ulink>
```

Sample XML Output

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<List_Wrapper>
          <datamap>
              <name>p6 Create Map</name>
          </datamap>
          <datamap>
              <name>p6 Update Map</name>
          </datamap>
</List_Wrapper>
```

Shell Methods

Create Shell

Description

This method will allow users to create shell instances under a shell type. User can create more than one shell at a time.

Support

This method only supports creation of shells under a company and is not associated with any project.

Installation: ASP and Self host

Level	Yes or No
Company Level	Yes

Level	Yes or No
Project and Shell Level	No
Program Level	No

Prototype

public XMLObject createShell(String shortname, String authcode, String copyFromShellTemplate, String shellXML);

Parameters

Parameter	Description
shortname	Identifier of the company, company's short name
authcode	Authentication code for the company, in text string
copyFromShellTemplate	This will be number of another shell instance or shell template. Based on shell template, shell type will be determined to create new shell
shellXML	This will be based on Shell integration interface definition. Without this definition defined user will not be allowed to create a shell instance

shellXML Elements

Tag Name	Required	Description	Valid values
<_shell>	Yes	Base tag for new shell data. This tag can be repeated to create more than one shell instance	
<_moduleoptions>	No	Base tag for options.	This tag is optional. If this tag is not provided then Primavera Unifier will create shell by copying user / groups from template or shell provided under copyFromShellTemplate.

Tag Name	Required	Description	Valid values
<copyModules>	No	Option to either copy modules or not.	This tag can only go as a child element under <_moduleOptions> tag. This tag is identify either to create a new shell by copying all modules including user / group or just creates a new shell by copying user / group from template provided under copyFromShellTemplate. Only valid values are Yes / No. Yes means copy all modules including user / groups. No means copy only user / groups. If tag is not provided or value is empty then default behavior will be to No.
<defaultCurrency >	No	Option to specify the currency code for the shell	A new child element of <defaultCurrency> will be available within the <_moduleoptions> tag. Valid values for the <defaultCurrency> tag are the code for the currency rather than the names. For example, USD is valid, United States Dollars is not. If the code is not a valid code then the web service will fail. If the tag is present and the value is empty then template currency will be selected.

Return Value

XMLObject (XMLObject is described in Appendix D of this document). Message will contain new Shell Number and Shell Name that is created within Primavera Unifier.

Sample Method

createShell("acme", "acme_authcode", "Proj-0001","<XML Content>");

Sample XML

```
<List_Wrapper>
<_shell>
     <!– all data elements added by user in shell integration interface definition. This includes
shell name, shell number etc..-- >
      <_moduleoptions>
         <copyModules>no</copyModules>
             <defaultCurrency>USD</defaultCurrency>
    </moduleoptions>
</_shell>
</List_Wrapper>
```

Note: If you do not provide values for elements that are not required then the values from template will be used to create new shell.

Additional Information

User will be allowed to create shells of type WBS and Generic. For more information out types of shells refer to uDesigner Guide.

User will be able to pass in a value for Location Picker element while creating shell instance.

All other validation while creating shell instance manually will be honored through Integration viz.., required fields, duplicate shell id etc..,

Users will be allowed to create multiple shells using this service call. If creation of one shell fails then entire transaction will be rolled back.

All shells created will be based on same template

Shell currency will be based on the template.

Update Shell

Description

This method will allow users to update shell instances. User can update more than one shell at a time.

Support

This method only supports update of shells under a company and is not associated with any project.

Installation: ASP and Self host

Level	Yes or No
Company Level	Yes
Project and Shell Level	No
Program Level	No

Prototype

public XMLObject updateShell (String shortname, String authcode, String shellType, String shellXML);

Parameters

Parameter	Description
shortname	Identifier of the company, company’s short name
authcode	Authentication code for the company, in text string
shellType	To identify type of the shell that is getting updated. Ex: Building, Projects, Capital Projects, Properties etc..,
shellXML	This will be based on Shell integration interface definition. Without this definition defined user will not be allowed to update a shell instance

shellXML Elements

Tag Name	Required	Description	Valid values
<_shell>	Yes	Base tag for new shell data. This tag can be repeated to create more than one shell instance	

Return Value

XMLObject (XMLObject is described in Appendix D of this document). Message will contain Shell Number and Shell Name that is updated within Primavera Unifier.

Sample Method

updateShell("acme", "acme_authcode", "Buildings","<XML Content>");

Sample XML

```
<List_Wrapper>
<_shell>
    <!– all data elements added by user in shell integration interface definition. This includes shell
name, shell number etc..-- >
</_shell>
</List_Wrapper>
```

Additional Information

Shell that will be updated will be based on shell type and shell number provided as part of XML content.

User will be allowed to update both WBS and Generic shells. For more information about shells refer to uDesigner Guide.

User will be able to pass in a value for Location Picker element while updating shell instance. Location picker value will be validated.

All other validation while creating shell instance manually will be honored through Integration viz.., required fields, duplicate shell id etc.

This method will allow user to update one more shell of same type at a time.

Get Shell List

Description

This method will allow users to get a list of shell instances. User can get information of more than one shell at a time.

Support

Installation: ASP and Self host

Level	Yes or No
Company Level	Yes
Project or Shell Level	No
Program Level	No

Prototype

public XMLObject getShellList(String shortname, String authcode, String shellType, String filterCondition);

Parameters

Parameter	Description
shortname	Identifier of the company, company's short name
authcode	Authentication code for the company, in text string
shellType	To identify type of the shell that is getting updated. Ex: Building, Projects, Capital Projects, Properties etc..,
filtercondition	This will allow user to filter list of shells returned. Following is list of elements that use can filter shell list on <shell number>: data element used to capture shell number <shell name>: data element used to capture shell name Status (uuu_shell_status): Status of the shell

Return Value

XMLObject (XMLObject is described in Appendix D of this document). Message will return list of shells based on filer condition

Sample Method

getShellList("acme", "acme_authcode", "Buildings", uuu_shell_status=Active);

Sample XML Input

```
<ulink>
        <_shortname>acme</_shortname>
        <_authcode>acme_authcode</_authcode>
        <_servicename>getShellList</_servicename>
        <_shelltype>Shell Type</_shelltype>
        <_filtercondition>DE for Shell name=shell name</_filtercondition>
</ulink>
```

Additional Information

Only one filter condition can be provided at a time.

Only operator available for filter condition will be "=".

Example:

If user wants to get a list of shells with status Active then filterCondition should be <filterCondition>uuu_shell_status = Active</filterCondition>

If user wants to get a shell with name Building 101 then filterCondition should be <filterCondition>my_building_name = Building 101</filterCondition). Assuming that my_building_name is the data element used to capture name of the shell using SYS Shell Name.

Get Project/Shell List

Description

This method will allow users to get a list of project and shell instances. User can get information of more than one project and shell at a time.

Support

Installation: ASP and Self host

Level	Yes or No
Company Level	Yes
Project or Shell Level	No
Program Level	No

Prototype

public XMLObject getProjectShellList(String shortname, String authcode, String options);

Parameters

Parameter	Description
Shortname	Identifier of the company, company's short name
Authcode	Authentication code for the company, in text string
Options	This parameter allows the user to filter the list based on the following: <status>: select a status of Active, Inactive, On-Hold or View-Only. If no status is provided then all projects and shells are returned. <type>: select the type to be returned. Value can be project, wbs_shell or generic_shell. If no type if provided then all projects and shells will be returned. <filter_condition>: filter the results based on either the projectnumber or projectname.

Return Value

XMLObject (XMLObject is described in Appendix D of this document). Message will return list of project and shells based on filter conditions

Sample Method

getShellList("acme", "acme_authcode", type=wbs_shell);

Sample XML Input

```
<ulink>
        <_shortname>acme</_shortname>
        <_authcode>acme_authcode</_authcode>
        <_servicename>getProjectShellList</_servicename>
        <options>
                        <status>Active</status>
                        <type>wbs_shell</type>
                 <filtercondition> </filtercondition>
                </options>
</ulink>
```

Sample XML Output

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<List_Wrapper>
        <projectshell>
           <projectname>California</projectname>
           <projectnumber>State0001</projectnumber>
           <status>Active</status>
           <type>Project</type>
        </projectshell>
        <projectshell>
            <projectname>Virginia</projectname>
            <projectnumber>State0002</projectnumber>
            <status>Active</status>
            <type>Project</type>
        </projectshell>
</List_Wrapper>
```

Configurable Manager Methods

Create Records in Code & Records-based Generic Manager

Description

This method creates records under a generic manager class. This applies only to code-& records based generic managers.

Support

This applies to code-& records based generic managers at Company and Project or Shell levels.

Installation: ASP and Self host

Level	Yes or No
Company Level	Yes
Project and Shell Level	Yes
Program Level	No

Prototype

public XMLObject createConfigurableModuleRecord(String shortname, String authcode, String projectNumber, String CMCode, String ClassName, String copyFromRecord, String recordXML);

Parameters

Parameter	Description
Shortname	identifier of the company, company's short name
Authcode	authentication key for the company, in text string

Parameter	Description
projectNumber	Identifier of project or shell in Primavera Unifier. Value can be left empty/ignored if specified CM is a company level manager.
CMCode	Identify CM (‘CM1’, ‘CM2’, …)
ClassName	Name of the class (Buildings, Electrical, IT Equipments etc..,)
copyFromRecord	Can be left empty if not copying from another record
recordXML	content of the record that needs to be created. This is based on Integration interface design for class in uDesigner.

Return Value

XMLObject (XMLObject is described in Appendix D of this document)

Sample Method

createConfigurableModuleRecord ("acme", "acme_authcode", "Buildings","CM5", "Electrical","", "XML")

Sample XML Return Data

```
<List_Wrapper>
    <_bp>
        < uuu_cm5_name />
        < status />
  </_bp>
  </List_wrapper>
```

Additional Information

Same logic while creating new records from UI will be used while creating a record from integration. This includes validation logic, setting status, etc.

All validations on detail form including required fields and user defined form validations will be honored while creating new records through web services.

Once record is created, detail form will be in Finish Editing mode. Data will be rolled up to the corresponding manager sheet.

Data Element which is based on data definition viz.., SYS Numeric Logical Datasource, SYS Date Logical Datasource, SYS Business Process Datasource and SYS Project Cost Datasource will not be available as part of Integration.

This service can be used for both Project (Standard) and Shells of cost code type WBS and Generic.

Update Records in Code & Records-based Generic Manager

Description

This method updates records under a generic manager class. This applies only to code-& records based generic managers.

Support

This applies to code-& records based generic managers at Company and Project or Shell levels.

Installation: ASP and Self host

Level	Yes or No
Company Level	Yes
Project and Shell Level	Yes
Program Level	No

Prototype

public XMLObject updateConfigurableModuleRecord(String shortname, String authcode, String projectNumber, String CMCode, String ClassName, String copyFromRecord, String recordXML);

Parameters

Parameter	Description
Shortname	identifier of the company, company's short name
Authcode	authentication key for the company, in text string
projectNumber	Identifier of project or shell in Primavera Unifier. Value can be left empty/ignored if specified CM is a company level manager.
CMCode	Identify CM ('CM1', 'CM2', …)
ClassName	Name of the class (Buildings, IT Equipments etc..,)
copyFromRecord	Can be left empty if not copying from another record
recordXML	Content of the record that needs to be created. This is based on Integration interface design for class in uDesigner.

Return Value

XMLObject (XMLObject is described in Appendix D of this document)

Sample Method

updateConfigurableModuleRecord ("acme", "acme_authcode", "Buildings","CM5", "Electrical","", "XML")

Sample XML Return Data

```
<List_Wrapper>
    <_bp>
        < uuu_cm5_name />
        < status />
  </_bp>
  </List_wrapper>
```

Additional Information

Same logic while creating new records from UI will be used while creating a record from integration. This includes validation logic, setting status, etc.

All validations on detail form including required fields and user defined form validations will be honored while creating new records through web services.

Once record is created, detail form will be in Finish Editing mode. Data will be rolled up to the corresponding manager sheet.

Data Element which is based on data definition viz.., SYS Numeric Logical Datasource, SYS Date Logical Datasource, SYS Business Process Datasource and SYS Project Cost Datasource will not be available as part of Integration.

This service can be used for both Project (Standard) and Shells of cost code type WBS and Generic

Exchange Rates Methods

Update Exchange Rates

Description

This method adds and updates exchange rates defined at company level.

Support

This applies to exchange rates at company level only.

Installation: ASP and Self host

Level	Yes or No
Company Level	Yes
Project and Shell Level	No
Program Level	No

Prototype

public XMLObject updateExchangeRates(String shortname, String authcode, String rates;

Parameters

Parameter	Description
Shortname	identifier of the company, company’s short name
Authcode	authentication key for the company, in text string
Rates	Exchange rates of one or more currency codes with respect to company base currency.

Return Value

XMLObject (XMLObject is described in Appendix D of this document)

Sample Method

updateExchangeRates ("acme", "acme_authcode", "XML")

Sample XML Input Data

```
<list_wrapper>
        <exchange_rate>
                <currency_name>Australia Dollar</currency_name>
                <currency_code>AUD</currency_code>
                <rate>2.086</rate>
        </exchange_rate>
        <exchange_rate>
                <currency_name>United States Dollar</currency_name>
                <currency_code>USD</currency_code>
                <rate>1.45</rate>
        </exchange_rate>
</list_wrapper>
```

Additional Information

User can add rates using Web Services call.

updateExchangeRates will always create new exchange rate record and will mark it "Active."

New currency codes can be added as part of this call.

If an invalid currency code is provided as part of currency code then that code will be ignored and other codes will be processed.

If the call is successful, new record with rates will be added.

Currency name is not mandatory.

For valid currency codes refer to Appendix C.

Get Exchange Rates

Description

This method will get exchange rates from the latest exchange rates record.

Support

Installation: ASP and Self host

Level	Yes or No
Company Level	Yes
Project and Shell Level	No
Program Level	No

Prototype

public XMLObject getExchangeRates(String shortname, String authcode, String rates);

Parameters

Parameter	Description
Shortname	identifier of the company, company's short name
Authcode	authentication key for the company, in text string
Rates	Exchange rates of one or more currency codes with respect to company base currency.

Return Value

XMLObject (XMLObject is described in Appendix D of this document)

Sample Method

getExchangeRates("acme", "acme_authcode", rates)

Sample XML Return Data

```
<list_wrapper>
        <exchange_rate>
                <currency_name>Australia Dollar</currency_name>
                <currency_code>AUD</currency_code>
                <rate>2.086</rate>
        </exchange_rate>
        <exchange_rate>
                <currency_name>United States Dollar</currency_name>
                <currency_code>USD</currency_code>
                <rate>1.45</rate>
        </exchange_rate>
</list_wrapper>
```

Schedule of Values Methods

Get SOV

Description

This method will get schedule of values information of a commitment.

Support

Installation: ASP and Self host

Level	Yes or No
Company Level	No
Project and Shell Level	Yes
Program Level	No

Prototype

public XMLObject getSOV(String shortname, String authcode, String projectNumber, String BPName, String recordNumber);

Parameters

Parameter	Description
Shortname	Identifier of the company, company's short name
Authcode	Authentication key for the company, in text string
projectNumber	Identifier of project or shell in Primavera Unifier
BPName	Identifier of the base commits business process in Primavera Unifier, for example: Contracts. This is case sensitive, and should be the name as given in uDesigner
RecordNumber	The record number is base commit record number that the SOV is associated to.

Return Value

XMLObject (XMLObject is described in Appendix D of this document)

Sample Method

getSOV("acme", "acme_authcode", "PRJ-0001", "Contracts", "CON-0001")

Sample XML Request

```
<ulink>
        <_shortname>acme</_shortname>
        <_authcode>acme_authcode</_authcode>
        <_servicename>getSOV</_servicename>
        <_projectNumber>PRJ-0001</_projectNumber>
        <_BPName>base commit business process name</_BPName>
        <_recordNumber>record number</_recordNumber>
</ulink>
```

Sample XML Return Data

```
<List_Wrapper>
<SOV>
                <SOV_header>
                        <c1>Ref</c1>
                        <c2>Description</c2>
                        <c3>WBS Code</c3>
                        <c4>Breakdown</c4>
                        <c5>Scheduled Value</c5>
<this section will also include all other columns that are defined on the SOV sheet>
                </SOV_header>
                <SOV_row>
                        <c1>1</c1>
                        <c2>Accountingcode</c2>
                        <c3>2</c3>
                        <c4 />
                        <c5>20.0</c5>
                </SOV_row>
                <SOV_row>
                        <c1 />
                        <c2 />
                        <c3>1</c3>
                        <c4>breakdown 1</c4>
                        <c5>0.0</c5>
                </SOV_row>
</SOV>
</List_Wrapper>
```

Additional Information

getSOV can be used to get the SOV for General Spends and Payment Applications.

Tags under <SOV_header> parent tag will change if the SOV is group by commit codes or individual line items.

User Administration Methods

Create User

Description

This public method will allow users to create one or more company users.

> **Note**: If you access Unifier through OIM, you will not be able to create user.

Support

Installation: ASP and Self host

Level	Yes or No
Company Level	Yes
Project and Shell Level	No
Program Level	No

Prototype

public XMLObject createUser(String shortname, String authcode, String copyFromUserPreferenceTemplate, String userXML);

Parameters

Parameter	Description
Shortname	Identifier of the company, company's short name
Authcode	Authentication key for the company, in text string
copyFromUserPreferenceTemplate	Name of the template from which user preference should be copied over when new user is created
userXML	Information of the user(s) that must be created based on the integration interface definition of the User Attribute form in uDesigner.

Return Value

XMLObject (XMLObject is described in Appendix D of this document)

Sample Method

createUser("acme", "acme_authcode", "User Pref Temp 1", "<user information>")

Sample XML Input Data

```
</_user>
                <elements added to user integration interface of user attribute in uDesigner>
</_user>
```

Additional Information

If the user preference template is provided then the user preference will be set based on the user preference template on the newly created company user.

'_user' will be the base tag which can be repeated with the data in order to create multiple company users.

The Web Services call for user administration includes a XML tag to support the Proxy login setup. The Data Element (DE) for the XML tag is: "`uuu_user_proxy_config`."

This is not a required field, so the default value remains unchecked.

Users can be created using the CreateOject WS method, and the XML tag is: "`allowproxy`."

To map the values for checkbox:

- If `uuu_user_proxy_config` or `allowproxy` is set to Yes, then the Check box “Do not allow Proxies” in Proxy tab should be unchecked.
- If `uuu_user_proxy_config` or `allowproxy` is set to No, then the Check box “Do not allow Proxies” in Proxy tab should be checked.

Sample XML

```
  <?xml version="1.0" encoding="ISO-8859-1" ?>
- <ulink>
  <_shortname>U90All</_shortname>
  <_authcode>123123</_authcode>
  <_copyfromuserpreferencetemplate />
  <_servicename>createUser</_servicename>
- <List_wrapper>
- <_user>
  <uuu_user_fax />
  <uuu_user_pager />
  <uuu_user_mobilephone />
  <uuu_user_status>1</uuu_user_status>
  <uuu_user_title />
  <uuu_user_homephone />
  <bp_data_picker_ps />
  <uuu_user_workphone />
  <uuu_user_loginname>webservicesuser</uuu_user_loginname>
  <uuu_user_proxy_config>Yes</uuu_user_proxy_config>
  <title>QA22</title>
  <uuu_user_dateformat>0</uuu_user_dateformat>
  <uuu_user_email>abc@skire.com</uuu_user_email>
  <uuu_user_firstname>ws1</uuu_user_firstname>
```

```
<uuu_user_lastname>user1</uuu_user_lastname>
<uuu_user_timezone_pref>51</uuu_user_timezone_pref>
<uuu_user_timezone>51</uuu_user_timezone>
<uuu_user_password>123123</uuu_user_password>
</_user>
</List_wrapper>
</ulink>
```

Update User

Description

This public method will allow users to update one or more company users.

Support

Installation: ASP and Self host

Level	Yes or No
Company Level	Yes
Project and Shell Level	No
Program Level	No

Prototype

public XMLObject updateUser(String shortname, String authcode, String userXML);

Parameters

Parameter	Description
Shortname	identifier of the company, company's short name
Authcode	authentication key for the company, in text string
userXML	Information of the user(s) that must be updated based on the integration interface definition of the User Attribute form in uDesigner.

Return Value

XMLObject (XMLObject is described in Appendix D of this document)

Sample Method

updateUser("acme", "acme_authcode", "<user information>")

Sample uLink XML Input Data

```
<List_Wrapper>
                <_user>
                    <uuu_user_loginname>user1login</uuu_user_loginname>
                    <uuu_user_firstname>user1firstname</uuu_user_firstname>
                    <uuu_user_lastname>user1lastname</uuu_user_lastname>
                    <uuu_user_email>user1email@acme.com</uuu_user_email>
                    <uuu_user_status>1</uuu_user_status>
                    <uuu_user_timezone_pref>110</uuu_user_timezone_pref>
                    <uuu_user_dateformat>MM/DD/YYYY HH:MM AM</uuu_user_dateformat>
                </_user>
  </List_Wrapper>
```

Additional Information

'_user' will be the base tag which can be repeated with the data in order to update multiple company users.

The unique identifier will be the user name and based on the matching user name, the passed on fields should be updated.

The Web Services call for user administration includes a XML tag to support the Proxy login setup. The Data Element (DE) for the XML tag is: "`uuu_user_proxy_config`."

This is not a required field, so the default value remains unchecked.

Users can be created using the CreateOject WS method, and the XML tag is: "`allowproxy`."

To map the values for checkbox:

- If `uuu_user_proxy_config` or `allowproxy` is set to Yes, then the Check box “Do not allow Proxies” in Proxy tab should be unchecked.
- If `uuu_user_proxy_config` or `allowproxy` is set to No, then the Check box “Do not allow Proxies” in Proxy tab should be checked.

Update User Group Membership

Description

This public method will allow users to update one or more company user’s membership to company level groups.

Support

Installation: ASP and Self host

Level	Yes or No
Company Level	Yes

Level	Yes or No
Project and Shell Level	No
Program Level	No

Prototype

public XMLObject updateUserGroupMembership(String shortname, String authcode, String membershipXML);

Parameters

Parameter	Description
Shortname	identifier of the company, company's short name
Authcode	authentication key for the company, in text string
membershipXML	Information of the user(s) that must be added and/or removed from company level groups

Return Value

XMLObject (XMLObject is described in Appendix D of this document)

Sample Method

updateUserGroupMembership("acme", "acme_authcode", "<user information>")

Sample uLink XML Input Data

```
<ulink>
        <_shortname>acme</_shortname>
        <_authcode>acme_authcode</_authcode>
        <_servicename>updateUserGroupMembership</_servicename>
        <List_Wrapper>
                <_user>
                        <username>user1</username>
                        <group_add>Group1</group_add>
                        <group_remove>Group2</group_remove>
                </_user>
        </List_Wrapper>
</ulink>
```

Additional Information

'_user' will be the base tag which can be repeated with the data in order to update multiple company users' group membership.

The unique identifier will be the user name and based on the matching user name, the passed on fields should be updated.

‘Group1’ and ‘Group2" in this sample XML data refer to group names for the user to be added to and removed from, respectively.

Get User List

Description

This public method will allow users to get a list of company users.

Support

Installation: ASP and Self host

Level	Yes or No
Company Level	Yes
Project and Shell Level	No
Program Level	No

Prototype

public XMLObject getUserList(String shortname, String authcode, String filterCondition);

Parameters

Parameter	Description
Shortname	identifier of the company, company’s short name
Authcode	authentication key for the company, in text string
filterCondition	Information or condition to filter users list

Return Value

XMLObject (XMLObject is described in Appendix D of this document)

Sample Method

getUserList("acme", "acme_authcode", "Status = Active")

Sample XML Return Data

```
<_user>
-               <elements added to user integration interface of user attribute in uDesigner>
</_user>
```

Additional Information

This method will allow users to get a list of company users based on the matching filter condition on either UserName or First Name or Last Name or Email or Status.

Only one filter condition can be provided at a time using the "=" operator.

Method will return the list of Users with matching conditions.

Update User Shell Membership

Description

This public method adds users to shell membership, updates user status in shell membership and adds/removes users to/from groups at shell level.

Support

Installation: ASP and Self host

Level	Yes or No
Company Level	No
Project and Shell Level	Yes

Prototype

public XMLObject updateUserShellMembership(String shortname, String authcode, String projectNumber, String membershipXML);

Parameters

Parameter	Description
Shortname	Identifier of the company, company's short name
Authcode	Authentication key for the company, in text string
projectNumber	Identifier of project or shell in Primavera Unifier
membershipXML	Information of the user(s) that must be added and/or removed from shell groups including activation / inactivation of user under a project or shell

Return Value

XMLObject (XMLObject is described in Appendix D of this document)

Sample Method

updateUserShellMembership("acme", "acme_authcode", "Proj-0001", "<user information>")

Sample uLink XML Input Data

```
<ulink>
        <_shortname>acme</_shortname>
        <_authcode>acme_authcode</_authcode>
                <_projectnumber>PROJ-00001</_projectnumber>
        <_servicename>updateUserShellMembership</_servicename>
        <List_Wrapper>
                        <_shelluser>
                                <username>user1</username>
                                <group_add>Group1</group_add>
                                <group_remove>Group2</group_remove>
                        </_ shelluser >
                        <_ shelluser >
                                <username>user2</username>
                                <status>Inactive</status>
                        </_ shelluser >
                        <_ shelluser >
                                <username>user3</username>
                                <status>Inactive</status>
                                <group_add>Group1</group_add>
                                <group_remove>Group2</group_remove>
                        </_ shelluser >
                        <_ shelluser >
                                <username>user4</username>
                                <status>Active</status>
                        </_ shelluser >
                </List_Wrapper>
</ulink>
```

Additional Information

The project/shell number is mandatory

'_ shelluser' will be the base tag which can be repeated with the data in order to update multiple company/shell users and their group membership.

The unique identifier will be the user name and based on the matching user name, the passed on fields should be updated.

‘group_add’, ‘group_remove’ and ‘status’ are optional tags.

Status can be either ‘Active’, ‘Inactive’ or empty.

Empty status or no status should be interpreted as ‘Active’ status.

‘Group1’ and ‘Group2’ in this sample XML data refer to group names for the user to be added to and removed from, respectively.

Pre-requisite for this web service to successfully add users to the shell membership:

Add the company user to the shell membership only if the user exists in company users log at the company level and is with active or on-hold status.

Add the partner user to the shell membership if the user exists in partner user log at the company level and is with active or on-hold status. If the partner user doesn’t exists at the company level than the web service will user to both company level partner user log and to the shell membership after checking for license restrictions

Planning Manager Methods

Get Planning Item Record

Description

This method gets a specific Planning Item record in Primavera Unifier. This works for both company level and project or shell level Planning Items.

Support

This process supports all type of Planning Items.

Installation: ASP and Self host

Level	Yes or No
Company Level	Yes
Project and Shell Level	Yes

Prototype

public XMLObject getPlanningItem (String shortname, String authcode, string projectNumber, String BPName, String planning Item);

Parameters

Parameter	Description
shortname	identifier of the company, company's short name
authcode	authentication key for the company, in text string
projectNumber	Identifier of the project or shell in Primavera Unifier. Pass null for Company Level Planning Items
BPName	identifier of the planning item type in Primavera Unifier, for example: Capital Projects
planningItem	identifier of the planning item example: Tower

Return Value

XMLObject (XMLObject is described in Appendix D of this document)

Sample Method

getPlanningItem("acme", "acme_authcode", "proj-01", "Capital Projects","Tower") will return the XML Record of the Planning Item.

Additional Information

If the planningItem is left blank or invalid planningItem then the method will returns the Planning Item’s skeletal XML.

To get company level Planning Item record information, send null for projectNumber parameter. If you send a project number for a Planning Item type that is at Company level then you will get an error that Planning Item type is not correct.

This service can be used for both Project (Standard) and Shells of cost code type WBS and Generic.

Space Manager Methods

Create Level

Description

This method creates level record in Unifier. This works only for project or shell level.

Support

Installation: ASP and Self host

Level	Yes or No
Company Level	No
Project and Shell Level	Yes

Prototype

public XMLObject createLevel (String shortname, String authcode, string projectNumber, String levelXML);

Parameters

Parameter	Description
shortname	identifier of the company, company’s short name
authcode	authentication key for the company, in text string
projectNumber	Identifier of the project or shell in Unifier.
levelXML	XML that contains information related to level that is getting created

Return Value

XMLObject (XMLObject is described at the beginning of this document)

Sample Method

createLevel ("acme", "acme_authcode", "proj-01", levelXML) will return the XML Record of the Level created.

Sample INPUT Data

```
<ulink>
        <_shortname>acme</_shortname>
        <_authcode>acme_authcode</_authcode>
        <_servicename>createLevel</_servicename>
        <_projectNumber>proj-01</_projectNumber>
<List_wrapper>
  <_level>
           <uuu_sp_level_name>Level 1</uuu_sp_level_name>
           <status>status1</status>
           <uuu_sp_uom></uuu_sp_uom>
  </_level>
  <_level>
           <uuu_sp_level_name>Level 2 </uuu_sp_level_name>
           <status>status1</status>
           <uuu_sp_uom></uuu_sp_uom>
  </_level>
</List_wrapper>
</ulink>
```

Additional Information

Multiple levels can be created by repeating <_level> for each level record.

This service can be used for both Shells of cost code type WBS and Generic.

XML tags under _level element are based on the integration interface definition of level form.

Update Level

Description

This method updates level record in Unifier. This works only for project or shell level.

Support

Installation: ASP and Self host

Level	Yes or No
Company Level	No
Project and Shell Level	Yes

Prototype

public XMLObject updateLevel (String shortname, String authcode, string projectNumber, String levelXML);

Parameters

Parameter	Description
shortname	identifier of the company, company’s short name
authcode	authentication key for the company, in text string
projectNumber	Identifier of the project or shell in Unifier.
levelXML	XML that contains information related to level that is getting created

Return Value

XMLObject (XMLObject is described at the beginning of this document)

Sample Method

updateLevel ("acme", "acme_authcode", "proj-01", levelXML) will return the XML Record of the Level updated.

Sample INPUT Data

```
<ulink>
            <_shortname>acme</_shortname>
            <_authcode>acme_authcode</_authcode>
            <_servicename>updateLevel</_servicename>
            <_projectNumber>proj-01</_projectNumber>
<List_wrapper>
   <_level>
                <uuu_sp_level_name>Level 1</uuu_sp_level_name>
                <status>status1</status>
                <uuu_sp_uom>sq.ft</uuu_sp_uom>
   </_level>
   <_level>
                <uuu_sp_level_name>Level 2 </uuu_sp_level_name>
                <status>status1</status>
                <uuu_sp_uom> sq.ft </uuu_sp_uom>
   </_level>
</List_wrapper>
</ulink>
```

Additional Information

Multiple levels can be updated by repeating <_level> for each level record.

This service can be used for both Shells of cost code type WBS and Generic.

XML tags under _level element are based on the integration interface definition of level form.

To upate a level record in Unifier, value under <uuu_sp_level_name> will be used as identifier.

Get Level List

Description

This method will get a list of level record from Unifier. This works only for project or shell level.

Support

Installation: ASP and Self host

Level	Yes or No
Company Level	No
Project and Shell Level	Yes

Prototype

public XMLObject getLevelList(String shortname, String authcode, string projectNumber);

Parameters

Parameter	Description
shortname	identifier of the company, company's short name
authcode	authentication key for the company, in text string
projectNumber	Identifier of the project or shell in Unifier.

Return Value

XMLObject (XMLObject is described at the beginning of this document)

Sample Method

getLevelList ("acme", "acme_authcode", "proj-01") will return the XML Record of the Level records.

Sample INPUT Data

```
<ulink>
        <_shortname>acme</_shortname>
        <_authcode>acme_authcode</_authcode>
        <_servicename>getLevelList</_servicename>
        <_projectNumber>proj-01</_projectNumber>
</ulink>
```

Sample Output Data

```
<List_Wrapper>
<_level>
      <uuu_sp_level_name>Floor 1</uuu_sp_level_name>
      <status>In_Service</status>
      <uuu_sp_uom>sq.ft</uuu_sp_uom>
</_level>
<_level>
      <uuu_sp_level_name>Floor 2</uuu_sp_level_name>
      <status>In_Service</status>
      <uuu_sp_uom>sq.ft</uuu_sp_uom>
</_level>
</List_Wrapper>
```

Additional Information

XML tags under _level will be based on integration interface definition of level form.

This service can be used for both Shells of cost code type WBS and Generic.

Create Space

Description

This method creates space record of a space type in Unifier. This works only for project or shell level.

Support

Installation: ASP and Self host

Level	Yes or No
Company Level	No
Project and Shell Level	Yes

Prototype

public XMLObject createSpace (String shortname, String authcode, string projectNumber, String spacetype, String spaceXML);

Parameters

Parameter	Description
shortname	identifier of the company, company’s short name
authcode	authentication key for the company, in text string
projectNumber	Identifier of the project or shell in Unifier.
spacetype	Identifier of the space type
spaceXML	XML that contains information related to space that is getting created

Return Value

XMLObject (XMLObject is described at the beginning of this document)

Sample Method

createSpace ("acme", "acme_authcode", "proj-01", "Useable Spaces", spaceXML) will return the XML Record of the Level created.

Sample INPUT Data

```
<ulink>
        <_shortname>acme</_shortname>
        <_authcode>acme_authcode</_authcode>
        <_servicename>createSpace</_servicename>
        <_projectNumber>proj-01</_projectNumber>
                <_spacetype>Useable Spaces</_spacetype>
<List_wrapper>
  <_space>
           <uuu_sp_level_picker>Floor 1</uuu_sp_level_picker>
           <space_sp_space_name>Room 1</space_sp_space_name>
           <space_usable_space_type></space_usable_space_type>
           <uuu_sp_area>200</uuu_sp_area>
           <uuu_sp_uom>Sq.ft</uuu_sp_uom>
  </_space>
  <_space>
           <uuu_sp_level_picker>Floor 1</uuu_sp_level_picker>
           <space_sp_space_name>Room 2</space_sp_space_name>
           <space_usable_space_type></space_usable_space_type>
           <uuu_sp_area>200</uuu_sp_area>
           <uuu_sp_uom>Sq.ft</uuu_sp_uom>
  </_space>
</List_wrapper>
</ulink>
```

Additional Information

Multiple spaces can be created by repeating <_space> for each space record.

This service can be used for both Shells of cost code type WBS and Generic.

XML tags under _space element are based on the integration interface definition of space form.

uuu_sp_level_picker element value will be used to identify the level under which the space record should be created.

Update SPACE

Description

This method will update space record in Unifier. This works only for project or shell level.

Support

Installation: ASP and Self host

Level	Yes or No
Company Level	No
Project and Shell Level	Yes

Prototype

public XMLObject updateSpace(String shortname, String authcode, string projectNumber, String spacetype, String spaceXML);

Parameters

Parameter	Description
shortname	identifier of the company, company's short name
authcode	authentication key for the company, in text string
projectNumber	Identifier of the project or shell in Unifier.
spacetype	Identifier of the space type
spaceXML	XML that contains information related to space that is getting updated

Return Value

XMLObject (XMLObject is described at the beginning of this document)

Sample Method

updateSpace ("acme", "acme_authcode", "proj-01","Useable Spaces", spaceXML) will return the XML Record of the Level updated.

Sample INPUT Data

```
<ulink>
        <_shortname>acme</_shortname>
        <_authcode>acme_authcode</_authcode>
        <_servicename>updateSpace</_servicename>
        <_projectNumber>proj-01</_projectNumber>
                <_spacetype>Useable Spaces</_spacetype>
<List_wrapper>
  <_space>
           <uuu_sp_level_picker>Floor 1</uuu_sp_level_picker>
           <space_sp_space_name>Room 1</space_sp_space_name>
           <space_usable_space_type></space_usable_space_type>
           <uuu_sp_area>200</uuu_sp_area>
           <uuu_sp_uom>Sq.ft</uuu_sp_uom>
  </_space>
  <_space>
           <uuu_sp_level_picker>Floor 1</uuu_sp_level_picker>
           <space_sp_space_name>Room 2</space_sp_space_name>
           <space_usable_space_type></space_usable_space_type>
           <uuu_sp_area>200</uuu_sp_area>
           <uuu_sp_uom>Sq.ft</uuu_sp_uom>
  </_space>
</List_wrapper>
</ulink>
```

Additional Information

Multiple spaces can be updated by repeating <_level> for each space record.

This service can be used for both Shells of cost code type WBS and Generic.

XML tags under _space element are based on the integration interface definition of space form.

To upate a space record in Unifier, value under <uuu_sp_level_picker> and data element based on data definition Sys Space Name will be used as identifier.

Get Space List

Description

This method will get a list of space record from Unifier. This works only for project or shell level.

Support

Installation: ASP and Self host

Level	Yes or No
Company Level	No
Project and Shell Level	Yes

Prototype

public XMLObject getSpaceList(String shortname, String authcode, string projectNumber);

Parameters

Parameter	Description
shortname	identifier of the company, company’s short name
authcode	authentication key for the company, in text string
projectNumber	Identifier of the project or shell in Unifier.

Return Value

XMLObject (XMLObject is described at the beginning of this document)

Sample Method

getSpaceList ("acme", "acme_authcode", "proj-01") will return the XML Record of the Level records.

Sample INPUT Data

```
<ulink>
        <_shortname>acme</_shortname>
        <_authcode>acme_authcode</_authcode>
        <_servicename>getSpaceList</_servicename>
        <_projectNumber>proj-01</_projectNumber>
</ulink>
```

Sample Output Data

```
  <_space>
           <uuu_sp_level_picker>Floor 1</uuu_sp_level_picker>
           <space_sp_space_name>Room 1</space_sp_space_name>
           <space_usable_space_type></space_usable_space_type>
           <uuu_sp_area>200</uuu_sp_area>
           <uuu_sp_uom>Sq.ft</uuu_sp_uom>
  </_space>
  <_space>
           <uuu_sp_level_picker>Floor 1</uuu_sp_level_picker>
           <space_sp_space_name>Room 2</space_sp_space_name>
           <space_usable_space_type></space_usable_space_type>
           <uuu_sp_area>200</uuu_sp_area>
           <uuu_sp_uom>Sq.ft</uuu_sp_uom>
  </_space>
```

Additional Information

XML tags under _space will be based on integration interface definition of level form.

This service can be used for both Shells of cost code type WBS and Generic.

Resource Manager Methods

Create and/or update Role

Description

This method creates (and updates) Resource Manager Roles.

Support

This method can only support creating and updating Roles at the company level.

Installation: ASP and Self host

Level	Yes or No
Company Level	Yes
Project and Shell Level	No

Prototype

public XMLObject createUpdateRole (String shortname, String authcode, String rolesXML);

Parameters

Parameter	Description
shortname	identifier of the company, company’s short name
authcode	authentication key for the company, in text string
rolesXML	Information of the role(s) that must be created and/or updated

Return Value

XMLObject (XMLObject is described in Appendix D of this document)

Sample Method

createUpdateRole("acme", "acme_authcode", "XML")

Sample XML input Data

```
<ulink>
        <_shortname>acme</_shortname>
        <_authcode>acme_authcode</_authcode>
        <_servicename>createUpdateRole</_servicename>
<list_wrappe>
 <_role>
              <role_name>Architect</role_name>
              <role_status>Active</role_status>
              <role_currency>United States Dollar (USD)</role_currency>
<<Any other data elements that are defined as part of role integration interface definition.>>
</_role>
<_role>
              <role_name>Architect</role_name>
              <role_status>Active</role_status>
              <role_currency>United States Dollar (USD)</role_currency>
<<Any other data elements that are defined as part of role integration interface definition.>>
</_role>
</list_wrapper>
</ulink>
```

Additional Information

_role tag can be repeated to create multiple roles.

Role name will be used as a unique identifier to update existing role information.

Value of <role_currency> element should be valid, refer to list of valid currencies in this document.

Get Role List

Description

This method gets a list of roles created at company level.

Support

This method is only supported at company level.

Installation: ASP and Self host

Level	Yes or No
Company Level	Yes
Project and Shell Level	No

Prototype

public XMLObject getRoleList (String shortname, String authcode, String filterCondition);

Parameters

Parameter	Description
shortname	identifier of the company, company's short name
authcode	authentication key for the company, in text string
filtercondition	Condition based on which list of roles can be filtered

Return Value

XMLObject (XMLObject is described in Appendix D of this document)

Sample Method

getRoleList("acme", "acme_authcode",uuu_role_status=Active)

Sample XML input Data

```
<ulink>
        <_shortname>acme</_shortname>
        <_authcode>acme_authcode</_authcode>
        <_servicename>getRoleList</_servicename>
<_filtercondition>uuu_role_status=Active</_filtercondition>
</ulink>
```

Additional Information

Only one filter condition can be provided at a time.

Only operator available for filter condition will be "=".

Example:

If user wants to get a list of roles with status Active then filterCondition should be <filterCondition>uuu_role_status = Active</filterCondition>

If user wants to get a role with name "Architect" then filterCondition should be <filterCondition>uuu_role_name = Architect</filterCondition>.

Create and/or update Resource

Description

This method creates (and updates) Resource Manager Resources.

Support

This method can only support creating and updating Resources at the company level.

Installation: ASP and Self host

Level	Yes or No
Company Level	Yes
Project and Shell Level	No

Prototype

public XMLObject createUpdateResource (String shortname, String authcode, String resourceXML);

Parameters

Parameter	Description
shortname	identifier of the company, company's short name
authcode	authentication key for the company, in text string
resourceXML	Information of the resource(s) that will be created and/or updated

Return Value

XMLObject (XMLObject is described in Appendix D of this document)

Sample Method

createUpdateResource("acme", "acme_authcode", XML)

Sample XML input Data

```
<ulink>
            <_shortname>acme</_shortname>
            <_authcode>acme_authcode</_authcode>
            <_servicename>createUpdateResource</_servicename>
<list_wrapper>
<_resource>
<uuu_resc_name>Resource Name</uuu_resc_name>
<uuu_resc_code></uuu_resc_code>
<uuu_resc_status></uuu_resc_status>
<uuu_resc_proficiency></uuu_resc_proficiency>
<uuu_description></uuu_description>
<uuu_resc_capacity></uuu_resc_capacity>
<uuu_resc_skill></uuu_resc_skill>
<uuu_resc_sun_capacity></uuu_resc_sun_capacity>
<uuu_resc_mon_capacity></uuu_resc_mon_capacity>
<uuu_resc_fri_capacity></uuu_resc_fri_capacity>
<uuu_resc_sat_capacity></uuu_resc_sat_capacity>
<uuu_resc_thu_capacity></uuu_resc_thu_capacity>
<uuu_resc_tue_capacity></uuu_resc_tue_capacity>
<uuu_resc_wed_capacity></uuu_resc_wed_capacity>
<uuu_user_lastname>last name</uuu_user_lastname>
<uuu_user_firstname>first name</uuu_user_firstname>
<uuu_user_email></uuu_user_email>
</_resource>
</List_Wrapper>
</ulink>
```

Additional Information

_resource tag can repeat to create multiple resources.

Get Resource List

Description

This method gets a list of resources created at company level.

Support

This method is only supported at company level.

Installation: ASP and Self host

Level	Yes or No
Company Level	Yes
Project and Shell Level	No

Prototype

public XMLObject getResourceList(String shortname, String authcode, String filterCondition);

Parameters

Parameter	Description
shortname	identifier of the company, company’s short name
authcode	authentication key for the company, in text string
filtercondition	Condition based on which list of roles can be filtered

Return Value

XMLObject (XMLObject is described in Appendix D of this document)

Sample Method

getRoleList("acme", "acme_authcode",uuu_resc_status=Active)

Sample XML input Data

```
<ulink>
        <_shortname>acme</_shortname>
        <_authcode>acme_authcode</_authcode>
        <_servicename>getResourceList</_servicename>
<_filtercondition>uuu_resc_status=Active</_filtercondition>
</ulink>
```

Additional Information

Only one filter condition can be provided at a time.

Only operator available for filter condition will be "=".

Example:

If user wants to get a list of resources with status Active then filterCondition should be <filterCondition>uuu_resc_status = Active</filterCondition>

General instructions

BP record Data Processing Rules

Data Definition tags which have invalid value names will produce an error. If the tag is not sent, default value 0 is set if the type of the field is an integer.

Character fields: check for length.

Amount tag, where specified, checks for valid amount

For Cost type BP's, WBS code must be there.

For cost BP's, if the status tag is not given, Primavera Unifier scans to check if the BP has only one status value. If it has only one, Primavera Unifier will assume it as the default value for this record. If Primavera Unifier finds more than one status value, an error is reported.

Error checking is currently supported for Cost BP's only.

Data element of type Pull down will always take label value when a message is sent to Primavera Unifier.

For Business process that involved fund information, FBS code will be validated.

Any transaction against inactive WBS code will be rejected.

If a business process form has dynamic data set definition then data combination will be validated while creating or updating records through Integration.

Business Process Picker Support

Following Web Services calls will support Business Process Picker: createBPRecord, updateBPRecord, addLineItem, getBPRecord, getCompleteBPRecord, getBPList

Following validation will be performed when a value is send for Business Process Picker data element

System will assume that the value send through Integration message is the value for Source Element selected as part of reference process definition.

One and only one combination exists

If the combination is not valid or results into two records from reference business process, then integration will error out.

Similar process will be followed for line item picker on detail form.

General Validation Rules across all services

General commenting is only supported by getCompleteBPRecord service.

Special characters must be escaped in a valid XML document. The following characters **"**, ‘, < , >, & shall be escaped to: **&quot;** , **&apos;** , **&lt;**, **&gt;**, and **&amp;**

For Web Services createBPRecord, addBPLineItems, updateBPRecord, createProject the XML is scanned for validity and any errors are reported when found.

A non-required tag like <hello>hello</hello> is completely ignored

Date fields (which are set as Date type) must be in the following format:

MM/dd/yyyy

MM-dd-yyyy

MM/dd/yyyy HH:mm:ss

MM-dd-yyyy HH:mm:ss

MM-dd-yyyy HH:mm:ss z (this format accepts timezone)

MM/dd/yyyy HH:mm:ss z (this format accepts timezone)

yyyy/MM/dd HH:mm:ss

yyyy-MM-dd'T'HH:mm:ss

yyyy-MM-dd

MM-dd-yyyy hh:mm is a 24-hour clock. Users should specify time in hours instead of AM/PM notation.

In MM/dd/yyyy HH:mm:ss 'z' signifies timezone.

Please note that Primavera Unifier converts user specified time in PDT timezone and stores in the database in a PDT format.

For example, if a user sends 01/01/2011 15:01 EDT as the time format to Primavera Unifier, it stores it as 01/01/2011 12:01 PDT in the database. This setting is based on user preferences and Primavera Unifier will display the date according to the time zone specified by the user.

Tag names are case sensitive ex: <_BP> is not a valid tag.

When designing BP's which are related to Integration, All BP's should have the record_no tag in the upper forms and for cost related BP's amount field should be present in the line items.

If the XML has a leading tag <?xml encoding="iso-8859-1" ?> we support all characters within this ISO standard.

There are no form level validations (required field, form level validation rules etc..,), auto-populate feature through Integration if integration interface is not designed for a business process.

Tag names are case sensitive ex: <_BP> is not a valid tag.

When designing BP's which are related to Integration, All BP's should have the record_no **tag in the upper forms and for cost related BP's** amount **field should be present in the line items.**

If the XML has a leading tag <?xml encoding="iso-8859-1" ?> we support all characters within this ISO standard.

There are no form level validations (required field, form level validation rules etc..,), auto-populate feature through Integration if integration interface is not designed for a business process.

General Limitations across all services

getBPList service will always return label data for a pull down field. It will not return the values.

You cannot send fund related information for Cost Type BPs such as General Spends and Payment Applications.

Only following ISO tags are supported as part of Web Services message

–	NBSP 00A0 160	¡ 00A1 161	¢ 00A2 162	£ 00A3 163	¤ 00A4 164	¥ 00A5 165	¦ 00A6 166	§ 00A7 167	¨ 00A8 168	© 00A9 169	ª 00AA 170	« 00AB 171	¬ 00AC 172	SHY 00AD 173	® 00AE 174	¯ 00AF 175
B –	° 00B0 176	± 00B1 177	² 00B2 178	³ 00B3 179	´ 00B4 180	µ 00B5 181	¶ 00B6 182	· 00B7 183	¸ 00B8 184	¹ 00B9 185	º 00BA 186	» 00BB 187	¼ 00BC 188	½ 00BD 189	¾ 00BE 190	¿ 00BF 191
C –	À 00C0 192	Á 00C1 193	Â 00C2 194	Ã 00C3 195	Ä 00C4 196	Å 00C5 197	Æ 00C6 198	Ç 00C7 199	È 00C8 200	É 00C9 201	Ê 00CA 202	Ë 00CB 203	Ì 00CC 204	Í 00CD 205	Î 00CE 206	Ï 00CF 207
D –	Ð 00D0 208	Ñ 00D1 209	Ò 00D2 210	Ó 00D3 211	Ô 00D4 212	Õ 00D5 213	Ö 00D6 214	× 00D7 215	Ø 00D8 216	Ù 00D9 217	Ú 00DA 218	Û 00DB 219	Ü 00DC 220	Ý 00DD 221	Þ 00DE 222	ß 00DF 223
E –	à 00E0 224	á 00E1 225	â 00E2 226	ã 00E3 227	ä 00E4 228	å 00E5 229	æ 00E6 230	ç 00E7 231	è 00E8 232	é 00E9 233	ê 00EA 234	ë 00EB 235	ì 00EC 236	í 00ED 237	î 00EE 238	ï 00EF 239
F –	ð 00F0 240	ñ 00F1 241	ò 00F2 242	ó 00F3 243	ô 00F4 244	õ 00F5 245	ö 00F6 246	÷ 00F7 247	ø 00F8 248	ù 00F9 249	ú 00FA 250	û 00FB 251	ü 00FC 252	ý 00FD 253	þ 00FE 254	ÿ 00FF 255
	—0	—1	—2	—3	—4	—5	—6	—7	—8	—9	—A	—B	—C	—D	—E	—F

Following pickers are not supported through Web Services call:

- Commit Short Description
- Asset Picker
- Account Code Picker
- Planning Item Picker
- Resource Picker
- Project Picker
- Activity Picker
- Role Picker
- User Picker

Auto-populate from following attribute forms is not supported:

- Cost Attribute Form
- Fund Attribute Form
- DMS Attribute Form

For all web services that allow the addition of attachments, only zip format is supported only if the iszipfile tag is set to "yes". The gzip files (including the similar file formats) are not supported.

Known issues across all services

createWBS service is case insensitive. Service will create a WBS code with A-B-C-D and a-b-c-d.

Any transaction against inactive FBS code will not be rejected.

Auto populate of amount field on header form, from line items, is only supported through the design that has been done through uDesigner. Refer to uDesigner and Primavera Unifier User and Admin guide for more information.

Shell Name can be same under different parents. In this scenario, if users try to create a business process record by passing in a value for a shell picker data element, then system will throw an error that it found more than one shell with same name.

APPENDIX A

Create and Update Schedule Sheet

A Schedule Sheet under Primavera Unifier Schedule Manager supports the following objects. For more information, refer to the Primavera Unifier User Guide, Primavera Unifier Administration Guide, and uDesigner User Guide.

- Activities
- Resource Assignments
- Activity Dependencies (Successors and Predecessors) along with relationship (Start to Finish, Finish to Start, etc.) and lag information.
- Association of an activity with a cost code (WBS Code) from Primavera Unifier Cost Manager.

Integration with P6 involves creation, modification and association of these objects automatically through Web Services calls.

As of the current Primavera Unifier release, integration with P6 Server supports the following information for each object.

- Activities: All elements including User Defined Fields.
- Resource Assignments: All elements including User Defined Fields and Unit of Measure
- Activity Dependencies: Based on Activity Dependencies option
- Association with Cost Codes: Primavera Unifier uses special logic as part of integration to build WBS Code to match with Primavera Unifier WBS Code or Cost Codes under Cost Manager. Refer to Associate WBS Codes with Activities section in this document.

To create and update these objects as part of Integration, the user must first create a data mapping relationship between data elements of Primavera Unifier and Data Elements of P6. The following section explains Data Mapping concepts and how to create one.

Apart from data mapping, Primavera Unifier should also know how to identify Activities and Resources that are coming from Primavera Project. This is to allow Primavera Unifier to support update web service method. Primavera Unifier currently supports two ways to identify an Activity and one way to identify a Resource.

An Activity can be identified by a GUID. Each activity in Primavera Project has a GUID. This GUID will be stored when an activity is created in Primavera Unifier and can be used later as part of update process.

The second way to identify an activity is through Activity ID. Each activity in Primavera Project has an Activity ID. This ID will be stored when an activity is created and will be used later as part of update process.

Resources are identified by their name. Each resource in Primavera Project has a name and this will be stored when a resource assignment is created in Primavera Unifier.

Data Mapping

To create and update these objects as part of Integration, the user must first create a data mapping relationship between Primavera Unifier and P6 data elements. The user can create one or more data mapping for each Schedule Sheet.

To define a data mapping

1) Navigate to your project or shell and open a schedule sheet.
2) From the Menu bar, choose **File** > **Data Mapping**. The Data Mappings window opens.

   The Data Mappings window lists any mappings you created, and their default statuses. At run time, status tells Primavera Unifier to use this mapping and do not prompt for instructions. That means the Select Default Mapping window will not open.
3) Click the **Add** button. The Data Mapping window opens.
   - On the **General** tab, enter the name and the description for the data mapping and click **Apply** to save changes and keep the window open, or click **OK** to save changes and close the window.
   - On the **Activity** tab, you can map Primavera Unifier schedule sheet columns to external CSV headers and XML tags for the import or export of activity information. Choose data elements from Activity and Resource Attribute forms.

Data format	Data mapping requirements
Column	Select the data elements you want to import. Minimum required fields for XML import are Activity ID, Activity Name, Start Date, Finish Date, and Duration.
CSV Header	Enter the CSV header from the external source.
XML Element	Enter the corresponding XML from the external source

   - On the **Resources** tab, you can map resource information through XML integration. Enter the header in the XML element that corresponds to the column. Mapping resources is mandatory only if you select Import Resource Assignments on the Options tab.
   - On the **Options** tab you can configure XML options that the system uses when importing data.

In this field:	Do this:
XML Import	Define options for importing XML activity schedules into Unifier. Mapping options are the same for Primavera XML and Microsoft Project XML.
Options	You can either retain existing schedule information in Unifier, or overwrite it complete upon importing an external XML file. 1) Merge into existing schedule. With this selection, you have a sub-option to Delete Activities removed from the source schedule.

In this field:	Do this:
	2) Overwrite existing schedule replaces the existing schedule.
Data Elements	Select the appropriate checkboxes if you want to import activities (rows). 1) Activity Dependencies: Select this checkbox to retain activities from the XML source file. 2) Activity Calendar: Select this checkbox to retain the activity calendar from the XML source file. If imported, the activity calendar will trigger the recalculation of activity dates as needed, and will override any existing activity calendar association. If a calendar is not imported, the activity will use the existing calendar defined in the Schedule Sheet properties. There must be a calendar in Unifier with the same name as the calendar in the import file. (See "Creating Multiple Calendars".) 1) Resource Assignments: Select this checkbox to retain the resource assignments from the XML source file. If you want to import resources, define the resource types for the data definition SYS Resource Type. For Microsoft Project files, use standard resource types: Work, Material, and Cost. Upon import, these resource types will soft book. 2) WBS Codes: Select this checkbox to import WBS codes 3) Number of Levels: If you choose to import Primavera WBS codes, you can specify the segments that should be considered in the Primavera XML file for the codes (from 1 to 9) and the WBS code suffix mask. See "WBS code options for Primavera XML and Microsoft Project XML" for details. 4) Suffix Mask: Does not apply for Microsoft Project XML files, and this value is ignored when files of this type are imported. You can use a constant or a data element value in the Suffix Mask.
Data Elements	Select the appropriate checkboxes if you want to export activities (rows), and whether to include dependencies, resource assignments, and/or WBS codes.

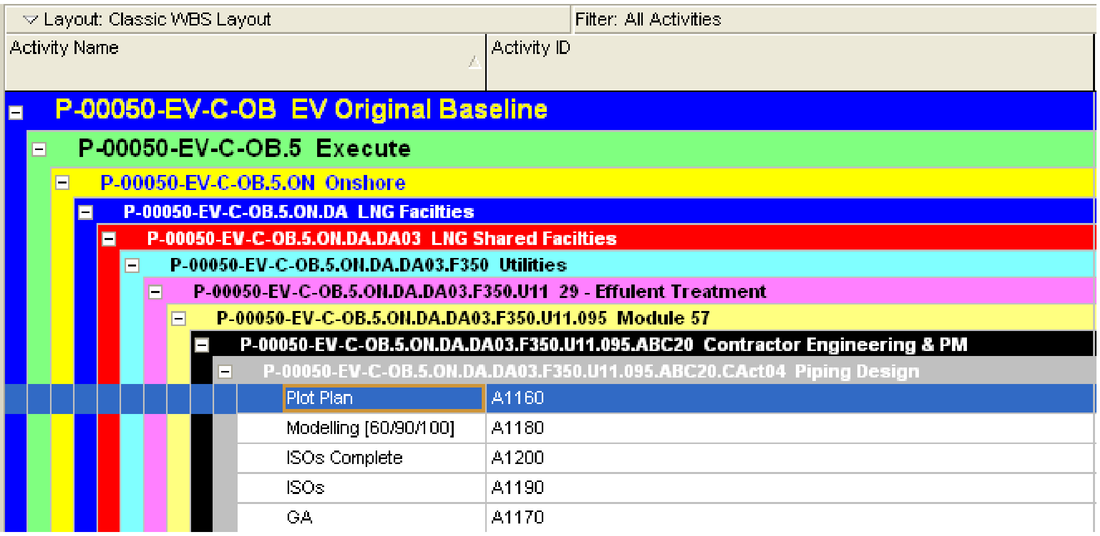
Click **OK**.

One of the data mappings can be selected as default. This defaulting concept will be used while processing a web services call. How to use a data mapping and where default data mapping is used is explained in Web Service Methods section.

As shown in Fig 7 a data mapping can be marked to override existing data or merge existing data. Based on the web services call and business need appropriate option must be selected.

Associate WBS Codes with Activities

In an Oracle® Primavera™ Project, Activities are grouped together by WBB Codes. Following is a sample project data.



Layout: Classic WBS Layout

Filter: All Activities

Activity Name	Activity ID
P-00050-EV-C-OB EV Original Baseline	
P-00050-EV-C-OB.5 Execute	
P-00050-EV-C-OB.5.ON Onshore	
P-00050-EV-C-OB.5.ON.DA LNG Facilties	
P-00050-EV-C-OB.5.ON.DA.DA03 LNG Shared Facilties	
P-00050-EV-C-OB.5.ON.DA.DA03.F350 Utilities	
P-00050-EV-C-OB.5.ON.DA.DA03.F350.U11 29 - Effulent Treatment	
P-00050-EV-C-OB.5.ON.DA.DA03.F350.U11.095 Module 57	
P-00050-EV-C-OB.5.ON.DA.DA03.F350.U11.095.ABC20 Contractor Engineering & PM	
P-00050-EV-C-OB.5.ON.DA.DA03.F350.U11.095.ABC20.CAct04 Piping Design	
Plot Plan	A1160
Modelling [60/90/100]	A1180
ISOs Complete	A1200
ISOs	A1190
GA	A1170

In this sample, observe that Activities A1160, A1180 are under a WBS Code built by a sequence of codes, 5 (Execute), ON (Onshore), DA (LNG Facilities) and so on. In this sample, activity A1160 (Plot Plan) is associated with WBS Code 5.ON.DA.DA03.F350.U11.095.ABC20.CAct04. XML file that gets generated by Primavera does not include the entire string of WBS Code (5.ON.DA.DA03.F350.U11.095.ABC20.CAct04) with all segments. XML structure is built based on a parent / child WBS segments. So Activity A1160 only has reference to its immediate parent segment, which in this example is CAct04. CAct04 segment will have a reference to ABC20 segment; ABC20 has reference to 095 and so on.

To associate an activity with a WBS Code, Primavera Unifier will go through this hierarchical structure of WBS segments to build the final WBS Code. Primavera Unifier will perform this function when WBS Code import option is checked on the Options tab of Data Mapping. The WBS Code string generated will then be validated with leaf level WBS Codes created under Cost Manager. If the WBS Code is available, then the activity will be associated with the WBS Code. If the WBS Code is not available, then the import process will fail.

Additional options are available to create entire string of WBS Code. These options can be used to address any structural differences between the way WBS Codes are built in Primavera Unifier Cost Manager and WBS Codes in Primavera Project.

- Levels: User can choose number of WBS segments that should be considered while processing a Primavera XML file. For example, if user enters 8 then last segment CAct04 will not be considered while building WBS Code.
- Suffix Mask: User can optionally enter a suffix mask, which will be added to the end of WBS Codes built based on Primavera XML file. Users can either enter a constant value of select a data element from activity attribute form.
- Note: With the help of this data element being selected as suffix, it is possible to build WBS Code and assign it to an activity based on attributes of activities coming from P6 instead of the WBS Code hierarchy of the activities.

APPENDIX B

XMLObject is described at the beginning of this document.

Currency

When importing Business Processes (BPs), you often must specify a project currency that Unifier can use for cost BPs and other calculations. You may need to enter the code corresponding to the currency used where the project is located. The currency can also be set in a user preference template. The following is a list of supported currencies and their corresponding codes in parenthesis.

- Afghani (AFN)
- Algeria Dinar (DZD)
- Ariary (MGA)
- Australia Dollar (AUD)
- Azerbaijani Manat (AZN)
- BCEAO Franc (XOF)
- Bahamas Dollar (BSD)
- Bahrain Dinar (BHD)
- Barbados Dollar (BBD)
- Belarus Ruble (BYR)
- Belize Dollar (BZD)
- Bermuda Dollar (BMD)
- Bhutan Ngultrum (BTN)
- Bolivia Boliviano (BOB)
- Botswana Pula (BWP)
- Brazil Real (BRL)
- Brunei Dollar (BND)
- Burundi Franc (BIF)
- CFA Franc BEAC (XAF)
- Canada Dollar (CAD)
- Cape Verde Escudo (CVE)
- Cayman Islands Dollar (KYD)
- Chile Peso (CLP)
- Chile UF (CLF)
- China Yuan Renminbi (CNY)
- Colombia Peso (COP)
- Comoros Franc (KMF)
- Comptoirs Francais du Pacifique Franc (XPF)

- Congo-Kinshasa Franc (CDF)
- Convertible Marka (BAM)
- Costa Rica Colon (CRC)
- Croatia Kuna (HRK)
- Cuba Peso (CUP)
- Cyprus Pound (CYP)
- Czech Republic Koruna (CZK)
- Denmark Krone (DKK)
- Dirham (AED)
- Djibouti Franc (DJF)
- Dobra (STD)
- Dominican Republic Peso (DOP)
- Dram (AMD)
- East Caribbean Dollar (XCD)
- Ecuador Dollar (ECD)
- Egypt Pound (EGP)
- El Salvador Colon (SVC)
- El Salvador Dollar (SVD)
- Eritrea Nakfa (ERN)
- Estonia Kroon (EEK)
- Ethiopia Birr (ETB)
- Euro (EUR)
- Falkland Islands Pound (FKP)
- Fiji Dollar (FJD)
- Gambia Dalasi (GMD)
- Georgia Lari (GEL)
- Ghana New Cedi (GHS)
- Gibraltar Pound (GIP)
- Gold Ounce (XAU)
- Guatemala Quetzal (GTQ)
- Guernsey Pound (GGP)
- Guilder (AWG)
- Guinea Franc (GNF)
- Guyana Dollar (GYD)
- Haiti Gourde (HTG)
- Honduras Lempira (HNL)
- Hong Kong Dollar (HKD)
- Hungary Forint (HUF)
- Iceland Krona (ISK)
- Indian Rupee (INR)
- Indonesia Rupiah (IDR)
- International Monetary Fund Special Drawing Right (XDR)

- Iran Rial (IRR)
- Iraq Dinar (IQD)
- Isle of Man Pound (IMP)
- Israel New Shekel (ILS)
- Jamaica Dollar (JMD)
- Jersey Pound (JEP)
- Jordan Dinar (JOD)
- Kazakhstan Tenge (KZT)
- Kenya Shilling (KES)
- Kuwait Dinar (KWD)
- Kwanza (AOA)
- Laos Kip (LAK)
- Latvia Lat (LVL)
- Lebanon Pound (LBP)
- Lek (ALL)
- Lesotho Loti (LSL)
- Lev (BGN)
- Liberia Dollar (LRD)
- Libya Dinar (LYD)
- Lithuania Litas (LTL)
- Macau Pataca (MOP)
- Macedonia Denar (MKD)
- Malawi Kwacha (MWK)
- Malaysia Ringgit (MYR)
- Maldives Rufiyaa (MVR)
- Malta Lira (MTL)
- Mauritania Ouguiya (MRO)
- Mauritius Rupee (MUR)
- Mexico Peso (MXN)
- Mongolia Tughrik (MNT)
- Morocco Dirham (MAD)
- Mozambique New Metical (MZN)
- Myanmar Kyat (MMK)
- Namibia Dollar (NAD)
- Nepal Rupee (NPR)
- Netherlands Antilles Guilder (ANG)
- New Romanian Leu (RON)
- New Turkish Lira (TRY)
- New Zealand Dollar (NZD)
- Nicaragua Gold Cordoba (NIO)
- Nigeria Naira (NGN)
- North Korea Won (KPW)

- Norway Krone (NOK)
- OmanRial (OMR)
- Paanga (TOP)
- Pakistan Rupee (PKR)
- Palladium Ounce (XPD)
- PanamaBalboa (PAB)
- Papua New Guinea Kina (PGK)
- Paraguay Guarani (PYG)
- Peru Nuevo Sol (PEN)
- Peso (ARS)
- Philippines Peso (PHP)
- Platinum Ounce (XPT)
- Poland Zloty (PLN)
- Portugal Escudo [obsolete] (PTE)
- Pound Sterling (GBP)
- Qatar Riyal (QAR)
- Riel (KHR)
- Russia Ruble (RUB)
- Rwanda Franc (RWF)
- Saint Helena Pound (SHP)
- Saudi Arabia Riyal (SAR)
- Seborga Luigino (SPL)
- Serbian dinar (RSD)
- Seychelles Rupee (SCR)
- Sierra Leone Leone (SLL)
- Silver Ounce (XAG)
- Singapore Dollar (SGD)
- Slovakia Koruna (SKK)
- Slovenia Tolar (SIT)
- Solomon Islands Dollar (SBD)
- Som (KGS)
- Somalia Shilling (SOS)
- South Africa Rand (ZAR)
- South Korea Won (KRW)
- South Sudanese pound (SSP)
- Sri Lanka Rupee (LKR)
- Sudanese Dinar (SDG)
- Suriname Dollar (SRD)
- Swaziland Lilangeni (SZL)
- Sweden Krona (SEK)
- Switzerland Franc (CHF)
- Syria Pound (SYP)

- Taiwan New Dollar (TWD)
- Tajikistan Somoni (TJS)
- Taka (BDT)
- Tala (WST)
- Tanzania Shilling (TZS)
- Thailand Baht (THB)
- Transnistria Moldova Leu (MDL)
- Trinidad and Tobago Dollar (TTD)
- Tunisia Dinar (TND)
- Turkmenistan Manat (TMM)
- Tuvalu Dollar (TVD)
- Uganda Shilling (UGX)
- Ukraine Hryvna (UAH)
- United States Dollar (USD)
- Uruguay Peso (UYU)
- Uzbekistan Som (UZS)
- Vatican City, Lira [obsolete] (VAL)
- Vatu (VUV)
- Venezuela Bolivar Fuerte (VEF)
- Vietnam Dong (VND)
- Western African Franc (CFA)
- Yemen Rial (YER)
- Yen (JPY)
- Yugoslavia New Dinar (YUM)
- Zambia Kwacha (ZMK)
- Zambia New Kwacha (ZMW)
- Zimbabwe Dollar (ZWD)

Time Zone

When importing user records, you must enter a starting time zone that Unifier can use for time stamps. You may wish to enter the code corresponding to the company office where the user is located. Time zone is a user preference setting that the user can change later if necessary. The time zone can also be set in a user preference template.

About UTC

The Coordinated Universal Time (UTC) is a 24-hour time standard. The westernmost time zone uses UTC−12, being twelve hours behind UTC; the easternmost time zone, theoretically, uses UTC+12, being twelve hours ahead of UTC. Calculation of the UTC time standard is the same as with Greenwich Mean Time (GMT).

Note: The local time in London is the same as UTC time.

The following table lists the UTC time standards used in Unifier.

Code	Time Standard/Zone	Information	Type
10	UTC+13:00	Samoa	Pacific/Apia
20	UTC-12:00	International Date Line West	Etc/GMT+12
21	UTC-10:00	Hawaii	Pacific/Honolulu
30	UTC-11:00	Coordinated Universal Time-11	Etc/GMT+11
40	UTC-09:00	Alaska	America/Anchorage
41	UTC-07:00	Chihuahua, La Paz, Mazatlan	America/Chihuahua
50	UTC-08:00	Pacific Time (US and Canada)	America/Los_Angeles
51	UTC-08:00	Baja California	America/Santa_Isabel
70	UTC-07:00	Arizona	America/Phoenix
71	UTC-07:00	Mountain Time (US and Canada)	America/Denver
80	UTC-06:00	Central America	America/Guatemala
90	UTC-06:00	Central Time (US and Canada)	America/Chicago
91	UTC-06:00	Saskatchewan	America/Regina
92	UTC-06:00	Guadalajara, Mexico City, Monterrey	America/Mexico_City
110	UTC-05:00	Eastern Time (US and Canada)	America/New_York
111	UTC-05:00	Indiana (East)	America/Indianapolis
112	UTC-05:00	Bogota, Lima, Quito	America/Bogota
130	UTC-04:00	Atlantic Time (Canada)	America/Halifax
131	UTC-04:30	Caracas	America/Caracas
132	UTC-04:00	Asuncion	America/Asuncion
133	UTC-04:00	Cuiaba	America/Cuiaba
134	UTC-04:00	Georgetown, La Paz, Manaus, San Juan	America/La_Paz
135	UTC-04:00	Santiago	America/Santiago
140	UTC-03:30	Newfoundland	America/St_Johns
150	UTC-03:00	Brasilia	America/Sao_Paulo
151	UTC-03:00	Buenos Aires	America/Buenos_Aires
152	UTC-03:00	Cayenne, Fortaleza	America/Cayenne

153	UTC-03:00	Greenland	America/Godthab
154	UTC-03:00	Montevideo	America/Montevideo
155	UTC-03:00	Salvador	America/Bahia
160	UTC-02:00	Coordinated Universal Time-02	Etc/GMT+2
170	UTC-01:00	Azores	Atlantic/Azores
171	UTC-01:00	Cape Verde Is.	Atlantic/Cape_Verde
180	UTC	Dublin, Edinburgh, Lisbon, London	Europe/London
181	UTC	Dublin	Europe/Dublin
182	UTC	Casablanca	Africa/Casablanca
183	UTC	Coordinated Universal Time	Etc/GMT
184	UTC	Monrovia, Reykjavik	Atlantic/Reykjavik
200	UTC+01:00	Amsterdam	Europe/Amsterdam
201	UTC+01:00	Brussels, Copenhagen, Madrid, Paris	Europe/Paris
202	UTC+01:00	Belgrade, Bratislava, Budapest, Ljubljana, Prague	Europe/Budapest
203	UTC+01:00	Amsterdam, Berlin, Bern, Rome, Stockholm, Vienna	Europe/Berlin
204	UTC+01:00	Sarajevo, Skopje, Warsaw, Zagreb	Europe/Warsaw
205	UTC+01:00	West Central Africa	Africa/Lagos
206	UTC+01:00	Windhoek	Africa/Windhoek
210	UTC+02:00	Athens, Bucharest	Europe/Bucharest
211	UTC+02:00	Cairo	Africa/Cairo
212	UTC+02:00	Beirut	Asia/Beirut
213	UTC+02:00	Jerusalem	Asia/Jerusalem
214	UTC+02:00	Helsinki, Kyiv, Riga, Sofia, Tallinn, Vilnius	Europe/Helsinki
215	UTC+02:00	Harare, Pretoria	Africa/Johannesburg
220	UTC+02:00	Damascus	Asia/Damascus
221	UTC+02:00	Istanbul	Europe/Istanbul
222	UTC+02:00	Tripoli	Africa/Tripoli
223	UTC+03:00	Amman	Asia/Amman
224	UTC+03:00	Kaliningrad, Minsk	Europe/Kaliningrad
225	UTC+03:00	Kuwait, Riyadh	Asia/Riyadh
230	UTC+03:00	Nairobi	Africa/Nairobi

231	UTC+04:00	Moscow, St. Petersburg, Volgograd	Europe/Moscow
232	UTC+03:00	Baghdad	Asia/Baghdad
240	UTC+03:30	Tehran	Asia/Tehran
250	UTC+04:00	Abu Dhabi, Muscat	Asia/Dubai
251	UTC+04:00	Baku	Asia/Baku
252	UTC+04:00	Port Louis	Indian/Mauritius
253	UTC+04:00	Tbilisi	Asia/Tbilisi
254	UTC+04:00	Yerevan	Asia/Yerevan
260	UTC+04:30	Kabul	Asia/Kabul
261	UTC+05:00	Ashgabat, Tashkent	Asia/Tashkent
270	UTC+06:00	Yekaterinburg	Asia/Yekaterinburg
280	UTC+05:00	Islamabad, Karachi	Asia/Karachi
290	UTC+05:30	Chennai, Kolkata, Mumbai, New Delhi	Asia/Calcutta
300	UTC+06:00	Astana	Asia/Almaty
301	UTC+06:00	Dhaka	Asia/Dhaka
302	UTC+06:30	Yangon (Rangoon)	Asia/Rangoon
310	UTC+07:00	Bangkok, Hanoi, Jakarta	Asia/Bangkok
311	UTC+08:00	Krasnoyarsk	Asia/Krasnoyarsk
320	UTC+08:00	Perth	Australia/Perth
321	UTC+08:00	Beijing, Chongqing, Hong Kong, Urumqi	Asia/Shanghai
322	UTC+08:00	Kuala Lumpur, Singapore	Asia/Singapore
323	UTC+08:00	Taipei	Asia/Taipei
324	UTC+09:00	Irkutsk	Asia/Irkutsk
340	UTC+09:00	Seoul	Asia/Seoul
341	UTC+09:00	Osaka, Sapporo, Tokyo	Asia/Tokyo
342	UTC+10:00	Yakutsk	Asia/Yakutsk
350	UTC+09:30	Darwin	Australia/Darwin
351	UTC+09:30	Adelaide	Australia/Adelaide
360	UTC+10:00	Brisbane	Australia/Brisbane
361	UTC+10:00	Canberra, Melbourne, Sydney	Australia/Sydney
362	UTC+10:00	Hobart	Australia/Hobart
363	UTC+11:00	Vladivostok	Asia/Vladivostok

364	UTC+10:00	Guam, Port Moresby	Pacific/Port_Moresby
380	UTC+07:00	Novosibirsk	Asia/Novosibirsk
390	UTC+08:00	Ulaanbaatar	Asia/Ulaanbaatar
391	UTC+11:00	Solomon Is., New Caledonia	Pacific/Guadalcanal
392	UTC+12:00	Magadan	Asia/Magadan
400	UTC+12:00	Coordinated Universal Time+12	Etc/GMT-12
410	UTC+12:00	Fiji	Pacific/Fiji
411	UTC+12:00	Auckland, Wellington	Antarctica/McMurdo
420	UTC+13:00	Nuku'alofa	Pacific/Tongatapu
430	UTC+05:30	Sri Jayawardenepura	Asia/Colombo
440	UTC+05:45	Kathmandu	Asia/Katmandu

Date Format

The date format is a user preference. You must enter a value for the user. The date format is a user preference setting that the user can change later if necessary. The date format can also be set in a user preference template.

The following lists the supported date formats:

Note: The time format for all dates is: HH:MM AM.

- MM/DD/YYYY
- DD/MM/YYYY
- MM/DD/YY
- DD/MM/YY
- MM-DD-YYYY
- DD-MM-YYYY
- MM-DD-YY
- DD-MM-YY
- DD.MM.YYYY
- YYYY-MM-DD
- MMM/DD/YYYY
- DD/MMM/YYYY
- YYYY/MMM/DD
- M/D/YYYY
- M/D/YY
- D/M/YYYY
- D/M/YY
- YY/M/D

- YYYY/M/D
- YY/MM/DD
- YYYY/MM/DD

Company Address

Use the codes included in the following table for the address fields, when entering the user records.

Address Type	Code
Headquarters	1
Main	2
Branch Office	3
Billing	4
Shipping	5
Billing and Shipping	6
Satellite Office	7

Status

Use the values included in the following table for any record in which Active, Inactive, or On Hold are the status choices. The status you choose for the import value will be effective as soon as the record is imported. For example, if you enter a "1" for the status, the user will be an active Unifier user upon successful import of the record; if you enter "2", the user will be On Hold until an administrator activates the user record from within Unifier.

Status	Value
Inactive	0
Active	1
On Hold	2

Check Box

For check boxes on BP forms, you can import the following values to select ("check") a check box or leave it unchecked.

Check Box	Value
Check box not selected	0
Check box selected	1

APPENDIX C

Currency Codes

Currency Name	Currency Code
Afghani	AFN
Algeria Dinar	DZD
Ariary	MGA
Australia Dollar	AUD
Azerbaijanian Manat	AZN
Bahamas Dollar	BSD
Bahrain Dinar	BHD
Barbados Dollar	BBD
BCEAO Franc	XOF
Belarus Ruble	BYR
Belize Dollar	BZD
Bermuda Dollar	BMD
Bhutan Ngultrum	BTN
Bolivia Boliviano	BOB
Botswana Pula	BWP
Brazil Real	BRL
Burundi Franc	BIF
Canada Dollar	CAD
Cape Verde Escudo	CVE
Cayman Isl ds Dollar	KYD
CFA Franc BEAC	XAF
Chile Peso	CLP
Chile UF	CLF
China Yuan Renminbi	CNY
Colombia Peso	COP

Currency Name	Currency Code
Comoros Franc	KMF
Comptoirs Francais du Pacifique Franc	XPF
Congo-Kinshasa Franc	CDF
Convertible Marka	BAM
Costa Rica Colon	CRC
Croatia Kuna	HRK
Cuba Peso	CUP
Cyprus Pound	CYP
Czech Republic Koruna	CZK
Denmark Krone	DKK
Dirham	AED
Djibouti Franc	DJF
Dobra	STD
Dominican Republic Peso	DOP
Dram	AMD
East Caribbean Dollar	XCD
Egypt Pound	EGP
El Salvador Colon	SVC
Eritrea Nakfa	ERN
Estonia Kroon	EEK
Ethiopia Birr	ETB
Euro	EUR
Falkland Islands Pound	FKP
Fiji Dollar	FJD
Gambia Dalasi	GMD
Georgia Lari	GEL
Ghana New Cedi	GHs
Gibraltar Pound	GIP
Guatemala Quetzal	GTQ
Guilder	AWG
Guinea Franc	GNF

Currency Name	Currency Code
Guyana Dollar	GYD
Haiti Gourde	HTG
Honduras Lempira	HNL
Hong Kong Dollar	HKD
Hungary Forint	HUF
Iceland Krona	ISK
Indian Rupee	INR
Indonesia Rupiah	IDR
Iran Rial	IRR
Iraq Dinar	IQD
Israel New Shekel	ILS
Jamaica Dollar	JMD
Jordan Dinar	JOD
Kazakhstan Tenge	KZT
Kenya Shilling	KES
Kuwait Dinar	KWD
Kwanza	AOA
Laos Kip	LAK
Latvia Lat	LVL
Lebanon Pound	LBP
Lek	ALL
Lesotho Loti	LSL
Lev	BGN
Liberia Dollar	LRD
Libya Dinar	LYD
Lithuania Litas	LTL
Macau Pataca	MOP
Macedonia Denar	MKD
Malawi Kwacha	MWK
Malaysia Ringgit	MYR
Maldives Rufiyaa	MVR

Currency Name	Currency Code
Malta Lira	MTL
Mauritania Ouguiya	MRO
Mauritius Rupee	MUR
Mexico Peso	MXN
Mongolia Tughrik	MNT
Morocco Dirham	MAD
Myanmar Kyat	MMK
Namibia Dollar	NAD
Nepal Rupee	NPR
Netherlands Antilles Guilder	ANG
New Mozambique Metical	MZN
New Romania Leu	RON
New Turkish Lira	TRY
New Zealand Dollar	NZD
Nicaragua Gold Cordoba	NIO
Nigeria Naira	NGN
North Korea Won	KPW
Norway Krone	NOK
OmanRial	OMR
Paanga	TOP
Pakistan Rupee	PKR
PanamaBalboa	PAB
Papua New Guinea Kina	PGK
Paraguay Guarani	PYG
Peru Nuevo Sol	PEN
Peso	ARS
Philippines Peso	PHP
Poland Zloty	PLN
Pound Sterling	GBP
Qatar Riyal	QAR
Riel	KHR

Currency Name	Currency Code
Russia Ruble	RUB
Rwanda Franc	RWF
Saint Helena Pound	SHP
Saudi Arabia Riyal	SAR
Serbian Dinar	RSD
Seychelles Rupee	SCR
Sierra Leone Leone	SLL
Singapore Dollar	SGD
Slovakia Koruna	SKK
Slovenia Tolar	SIT
Solomon Islands Dollar	SBD
Som	KGS
Somalia Shilling	SOS
South Africa Rand	ZAR
South Korea Won	KRW
South Sudanese Pound	SSP
Sri Lanka Rupee	LKR
Sudanese Dinar	SDG
Suriname Dollar	SDG
Sweden Krona	SEK
Switzerland Franc	CHF
Syria Pound	SYP
Taiwan New Dollar	TWD
Taka	BDT
Tala	WST
Tanzania Shilling	TZS
Thailand Baht	THB
Transnistria Moldova Leu	MDL
Trinidad and Tobago Dollar	TTD
Tunisia Dinar	TND
Turkmenistan Manat	TMM

Currency Name	Currency Code
Tuvalu Dollar	TVD
Uganda Shilling	UGX
Ukraine Hryvna	UAH
United States Dollar	USD
Uruguay Peso	UYU
Uzbekistan Som	UZS
Vatu	VUV
Venezuela Bolivar Fuerte	VEF
Vietnam Dong	VND
Yemen Rial	YER
Yen	JPY
Zambia Kwacha	ZMK
Zimbabwe Dollar	ZWD

APPENDIX D

Return Values

For all of the following methods, the return object will be XMLObject.

XMLObject contains 3 string elements:

- **String Status Code** (see table below)
- **String Array** – error log indicating reason of failure
- **String XMLContents** – a string contains XML values, if a return value is expected
- **XMLFileObject** – a file object that contains files as return value

String Status Code	Description
200	OK
300	Bad Data
302	Invalid Short name/AuthCode
303	Invalid Project Number
304	Invalid Business Process Name
405	Method Not Allowed
406	Invalid Server URL
500	Server Error, contact the system administrator
501	Not Implemented
503	Out of Resources
505	Invalid data
506	Invalid login
507	Error in creating directory for business process
601	Invalid Shortname/Authentication code
602	Project Number is not Correct
603	Business Process Name is not correct

String Status Code	Description
605	No Budget setup for the project
606	Budget is not correct.
608	XML is not valid
609	Project Template/Project Number is not correct
610	Only Cost and Line Item BPs are supported
611	Only Cost and Line Item BPs are supported
612	Project Template/Project name already exists
613	Project number already exists
614	ProjectXML should contain project number and project name tags.
615	Error processing XML.
616	XML is null or empty.
617	Found duplicate project numbers.
650	Create BP failed because …..
651	WBS Code is not Present
654	WBS Code is not in the correct format.
655	Could not create WBS Item.
656	Object Name is null or empty.
657	Invalid record_no.
658	field record_no is not present.
659	field record_no is empty.
660	field record_no is duplicate.
661	Integration settings for default workflow link id not set.
662	Integration setting for default workflow or creator or BP not set.
663	Status field value is not correct.
664	Create Project failed because
665	Data values for field tag
690	Only Lineitem non workflow BPs supported
691	Rule engine.
692	Cannot create multiple records for single record BP

String Status Code	Description
693	Cannot create workflow process
694	Project Name can only contain the following characters: alphanumeric, spaces, and ()[],.;:-_&<>'
695	Project Number can only contain the following characters : alphanumeric and [.-_] characters are allowed
701	Invalid Column Name.
702	Column is not defined as manual entry.
703	Administrator is inactive. Cannot create column data.
704	Duplicate Column Name.
705	Update Column Data failed because:
706	Get Column Data failed because:
707	Cost sheet does not exist.
708	Unable to run report. Contact System Administrator.
709	Report name is not valid. Check if report exists or is enabled for Integration.
710	Cannot create Asset type BP record using createBPRecord interface.
711	Invalid sheet name.
712	Import of Schedule activities failed reason :
713	Schedule sheet name is not valid.
714	Schedule sheet cannot be created; reason XML Import property is set to 'merge into existing schedule'.
715	Schedule sheet cannot be updated, reason XML Import property is set to 'overwrite existing schedule'.
716	The SOV of the selected contract is locked by another Business Process.
717	Selected contract does not have SOV. Check if the contract has reached terminal status.
718	Field name "+bp_picker_source_name +" is required.
719	pickervalue +" is invalid "+"value for "+source_name +"."
720	Reference BP not supported
721	Multiple records found for field "+source_name + " with value " + pickervalue+ "."
722	Invalid field name(s)

String Status Code	Description
723	Spend Business Process not supported.
724	SOV is not created yet for the base commit associated with this Business Process.
725	You cannot change a Business Process picker name + <source_name> + , that has line item(s) associated with it.
726	Reference Commit picker value cannot be changed.
800	No Active multiple instance
803	Copy from shell template is not active
804	(can be a combination of the following given below) can only contain the following characters: alphanumeric and [.-_] characters cannot contain any of the following characters: / \ : * ? " < > | Current shell cannot be a sub-shell of the selected shell Administrator is not a valid user
805	shell_name <shell name> already exists under <location>
806	shell_number <shell num> already exists
807	shell_name <shell name> already exists under <location>
808	shell_number <shell num> already exists
810	Integration Interface not found
811	Invalid Filter Condition
812	Copy from shell/template number is not correct
854	Amount exceeds remaining Reference Commit balance for one or more WBS codes
855	Reference commit value cannot be modified
867	Correct the syntax and resubmit the statement.
869	Number of line items in the input file exceeds number of lines in the SOV

For More Information

Where to Get Documentation

If you need help with an operation in Primavera Unifier, there are a number of resources to help you.

> **Note**: The Company Administrators for Unifier Cloud must use the *Primavera Unifier Cloud Service Administrators Setup Guide* instead of all the other installation and configuration guides in the Primavera Unifier Online Documentation Library.

Online Help

If you have a question or need further assistance regarding a specific feature, review the online help.

Click the **Help** menu from any Primavera Unifier window, and choose:

- **Unifier Help**: This link goes to the Oracle Technical Network (OTN) Primavera Unifier Online Document Library. From here you can access:
  - *Primavera Unifier Online Help*
  - *Primavera Unifier Administration Guide*
  - *Primavera Unifier User's Guide*
  - *Primavera uDesigner User's Guide*
  - *Primavera Unifier Reference Guide*

  You must have Adobe Acrobat Reader installed (available free at www.adobe.com).
- **BP-specific Help**: If your company has provided customized Help files for individual business processes, you can access them through the business process log or the Help menu.

> **Note:** For the full list of system requirements, supported software, and versions, see the *Tested Configurations* document under “Installation and Configuration Documentation” in the Primavera Unifier Online Documentation Library.

Documentation

Complete documentation libraries for Primavera Unifier releases are available on the Oracle Technology Network (OTN) at:

http://www.oracle.com/technetwork/documentation/default-1870233.html

From this location you can either view libraries online or download them to have local copies. We recommend viewing them from OTN to ensure you always access the latest versions, including critical corrections and enhancements.

Primavera Unifier is configured to access its help system directly on OTN. However, a downloadable version of the help system is also available on OTN if you need to download, deploy, and access a local copy.

The documentation assumes a standard setup of the product, with full access rights to all features and functions.

The following table describes the core documents available for Primavera Unifier.

Supplementary Documentation

Title	Description
What's New	Contains key enhancements in the current release of Primavera Unifier. All users should read this guide.
Cumulative Features Overview Tool	Provides an overview of features and enhancements included in Oracle Primavera products. It is intended solely to help you assess the business benefits of upgrading and to plan your I.T. projects. All users should read this guide.
Readme for Media Pack	Contains a table that can help you determine what downloads are required to run your licenses. All users should read this guide.
Licensing Information User Manual	Contains information about the third-party proprietary and open source software used in Oracle Primavera Unifier. All users should read this guide.
Where to Get Documentation, Training, and Support	Contains information about resources to help you with the operation of Oracle Primavera Unifier. All users should read this guide.

Installation Documentation

Title	Description
Primavera Unifier Installation and Setup Guide For WebLogic Application Server	Describes how to: - Set up Primavera Unifier servers and third party services - Install and configure Primavera Unifier components IT professionals who are installing and configuring the server environment, and the Primavera Unifier network administrator/database administrator and Primavera Unifier administrator should read this guide.
Primavera Unifier Installation and Setup Guide for WebSphere Application Server	Describes how to: - Set up the Primavera Unifier servers and third party services on Linux (64-bit) - Install and configure Primavera Unifier components IT professionals who are installing and configuring the server environment, and the Primavera Unifier network administrator/database administrator and Primavera Unifier administrator should read this guide.
Unifier Performance and Sizing Guide	Provides guidance for planning product deployment with: - Estimates of hardware and software requirements for Primavera Unifier - Recommendations for deploying small-scale to large-scale configurations The Unifier network administrator/database administrator and Unifier administrator should read this guide.
Unifier Upgrade Guide	Provides information on how to upgrade to the latest version of with Oracle Primavera Unifier. The Unifier network administrator/database administrator and Unifier administrator should read this guide.

Title	Description
Tested Configurations Previously: *System Configuration Information*	Provides the tested configurations for Oracle Primavera Unifier and includes list of applications and supported version levels. All users should read this guide. The Unifier network administrator/database administrator and Unifier administrator should read this guide.
Primavera Unifier Security Guide	Provides guidelines on how to plan your security strategy for Oracle Primavera Unifier. All users should read this guide. The Unifier network administrator/database administrator and Unifier administrator should read this guide.

Configuration Documentation

Title	Description
Primavera uClient Configurator Setup Guide	Describes how to set up the Primavera uClient Configurator. All users should read this guide.
Primavera Unifier File Transfer Utility Instructions	Describes how to download and install the Primavera Unifier File Transfer Utility. All users should read this guide.
uCAD Installation and User's Guide	Provides instructions on how to download and install the Primavera Unifier uCAD Application on your desktop and configure it to work with your AutoCAD® application. It also includes details on how to use uCAD with AutoCAD and Primavera Unifier. All users should read this guide.

Title	Description
Configuring BI Publisher for Unifier	Provides instructions on how to: - Download BI Publisher. - Configure BI Publisher settings. - Configure BI Publisher in Primavera Unifier. - Create BI Publisher reports. - Run BI Publisher reports. The Primavera Unifier network administrator/database administrator and Primavera Unifier administrator should read this guide.
Configuring Unifier for Single Sign On	Provides procedures to configure Primavera Unifier that is running on Windows (64-bit) or on Linux operating systems for Single Sign On (SSO). The Primavera Unifier network administrator/database administrator and Primavera Unifier administrator should read this guide.
Connecting Oracle Enterprise Manager to Primavera Unifier Metrics	Provides instructions on how to enable Enterprise Manager to display certain Primavera Unifier metrics. The Primavera Unifier network administrator/database administrator and Primavera Unifier administrator should read this guide.
Connecting Oracle Identity Manager 11g to Primavera Unifier	Guides administrators in configuring Oracle Identity Manager (OIM) to be used with Primavera Unifier. The Primavera Unifier network administrator/database administrator and Primavera Unifier administrator should read this guide.

Title	Description
Connecting the Content Repository to Primavera Unifier	Provides instructions on how to: - Configure Oracle WebCenter Content Core Capabilities after you install it. - Configure Microsoft SharePoint after you install it. - Configure CMIS-compliant repositories after you install one. - Configure the Primavera Unifier File Repository. - Set your content repository settings in Primavera Unifier to connect to your content repository. The Primavera Unifier network administrator/database administrator and Primavera Unifier administrator should read this guide.

User Documentation

Title	Description
Primavera Unifier Help	Provides topic-oriented information about Primavera Unifier to assist users with features and procedures. All users should read this guide.
Primavera Unifier User's Guide	Guides users on how to use Primavera Unifier application. All users should read this guide.
Primavera uDesigner User's Guide	Guides users on how to use the uDesigner feature of Primavera Unifier application. All users should read this guide.
Primavera Unifier Administration Guide	Provides details about the Administration Mode of Primavera Unifier, which covers access to Primavera Unifier functionality that is granted through permissions. All users should read this guide.

Title	Description
Primavera Unifier Project Controls User's Guide	Explains Oracle Primavera: - Capital Planning which is used for managing budget forecasts (both planned and in execution) for the user company. - Cost Controls which supports project tracking and project cost management. - Project Delivery Management which is designed to optimize project results. Anyone who wants to use the Unifier Project Controls base product should read this guide.

Database Documentation

Title	Description
Primavera Unifier Reference Guide	Provides details about data definitions, data elements, permission settings, import codes, and pickers with Primavera Unifier. All users should read this guide.

Integration Documentation

Title	Description
Primavera Unifier Integration Interface Guide	Describes the methods required for the integration, and the data being passed, between Primavera Unifier and an external system. Anyone who wants to develop applications which interact with Primavera Unifier should read this guide.

Distributing Information to the Team

You can copy the online documentation to a network drive for access by project participants. Each team member can then view or print those portions that specifically relate to his or her role in the organization.

Where to Get Training

To access comprehensive training for all Primavera products, go to: http://education.oracle.com

Oracle Learning Library

The Oracle Learning Library (OLL) provides online learning content covering Primavera products. Content includes whitepapers, videos, tutorials, articles, demos, step-by-step instructions to accomplish specific tasks, and self-paced interactive learning modules.

To access the learning library's Primavera content, go to: http://www.oracle.com/oll/primavera

Where to Get Support

If you have a question about using Oracle products that you or your network administrator cannot resolve with information in the documentation or help, click http://support.oracle.com/. This page provides the latest information on contacting Oracle Global Customer Support, knowledge articles, and the support renewals process. For more information about working with Support, visit https://support.oracle.com/epmos/faces/DocumentDisplay?id=888813.2 to view **Support Tools & Tips**.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/us/support/contact-068555.html or visit http://www.oracle.com/us/corporate/accessibility/support/index.html if you are hearing impaired.

Using Primavera's Support Resource Centers

Primavera's Support Resource Center provides links to important support and product information. Primavera's Product Information Centers (PICs) organize documents found on My Oracle Support (MOS), providing quick access to product and version specific information such as important knowledge documents, Release Value Propositions, and Oracle University training. PICs also offer documentation on Lifetime Management, from planning to installs, upgrades, and maintenance.

Visit https://support.oracle.com/epmos/faces/DocumentDisplay?id=1486951.1 to access links to all of the current PICs.

PICs also provide access to:

- **Communities**, which are moderated by Oracle providing a place for collaboration among industry peers to share best practices.
- **News** from our development and strategy groups.
- **Education** via a list of available Primavera product trainings through Oracle University. The Oracle Advisor Webcast program brings interactive expertise straight to the desktop using Oracle Web Conferencing technology. This capability brings you and Oracle experts together to access information about support services, products, technologies, best practices, and more.

Additional Support

The following is a list of additional support that are available in Unifier:

- Unifier Help
- Unifier Library
- Contact Support
- Download Plugins
- About Unifier

> **Note:** The Contact Support and Download Plugins options both open a single window, Support. The only difference is as follows:
> - Contact Support opens the *Contact* tab of the Support window by default
> - Download Plugins opens the *Download* tab of the Support window by default

Unifier Help

Click **Help** from the upper right-hand section of the Unifier window to open a contextual menu and click **Unifier Help** to open the Unifier and uDesigner Help.

Unifier Library

Click **Help** from the upper right-hand section of the Unifier window to open a contextual menu and click **Unifier Library** to open the Unifier documentation library.

Contact Support

Click **Help** from the upper right-hand section of the Unifier window to open a contextual menu and click **Contact Support** to open the Support window.

The Support window has two tabs:

- Contact, which opens by default
- Download

In the Contact tab, Contact Information section, you can see the name, phone, and contact instructions of someone at your company that you can contact when you need help with Primavera Unifier. For assistance, try this person, first.

> **Note:** This information is available only if your company administrator has provided internal support contact information.

In the Contact tab, eLearning section, you see the location (Access), name of the contact (Contact), and any instructions that you might need in order to access the eLearning materials.

Note: Your company may elect to provide access to the eLearning suite, where you can access interactive Primavera Unifier learning materials. Alternatively, your company may have its own customized support or training materials that can be accessed on an internal site/location. If your company administrator provides this contact information, it will be listed at the bottom of the window.

In the Download tab, Download section, you can see a list of the following application and plug-ins available to download, with installation instructions.

- File Transfer Application
- Unifier uCAD Plug-in

Each application and plug-in has two options:

- Download
- Installation instructions

Follow the links within each option to proceed.

Download Plugins

Click **Help** from the upper right-hand section of the Unifier window to open a contextual menu and click **Download Plugins** to open the Support window.

The Support window has two tabs:

- Contact
- Download, which opens by default

In the Download tab, Download section, you can see a list of the following application and plug-ins available to download, with installation instructions.

- File Transfer Application
- Unifier uCAD Plug-in

Each application and plug-in has two options:

- Download
- Installation instructions

Follow the links within each option to proceed.

In the Contact tab, Contact Information section, you can see the name, phone, and contact instructions of someone at your company that you can contact when you need help with Primavera Unifier. For assistance, try this person, first.

Note: This information is available only if your company administrator has provided internal support contact information.

In the Contact tab, eLearning section, you see the location (Access), name of the contact (Contact), and any instructions that you might need in order to access the eLearning materials.

Note: Your company may elect to provide access to the eLearning suite, where you can access interactive Primavera Unifier learning materials. Alternatively, your company may have its own customized support or training materials that can be accessed on an internal site/location. If your company administrator provides this contact information, it will be listed at the bottom of the window.

About Unifier

Click **Help** from the upper right-hand section of the Unifier window to open a contextual menu and click **About Unifier** to access version number, copyright information, and software agreement details for the Oracle Primavera Unifier.

Legal Notices

Oracle Primavera Prime Integration Interface Guide

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are “commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate failsafe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

This software or hardware and documentation may provide access to or information on content, products and services from third-parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.