



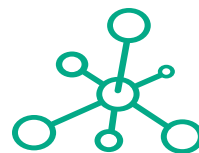
Project and Portfolio Management Center

Software Version: 9.40

Web Services Programmer's Guide

Document Release Date: September 2016

Software Release Date: September 2016



Hewlett Packard
Enterprise

Legal Notices

Warranty

The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from Hewlett Packard Enterprise required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notice

© 2016 Hewlett Packard Enterprise Development LP

Trademark Notices

Adobe® is a trademark of Adobe Systems Incorporated.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

UNIX® is a registered trademark of The Open Group.

Documentation Updates

To check for recent updates or to verify that you are using the most recent edition of a document, go to: <https://softwaresupport.hpe.com/>.

This site requires that you register for an HP Passport and to sign in. To register for an HP Passport ID, click **Register** on the HPE Software Support site or click **Create an Account** on the HP Passport login page.

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

The following table indicates changes made to this document since the last released edition.

Support

Visit the HPE Software Support site at: <https://softwaresupport.hpe.com/>.

Most of the support areas require that you register as an HP Passport user and to sign in. Many also require a support contract. To register for an HP Passport ID, click **Register** on the HPE Support site or click **Create an Account** on the HP Passport login page.

To find more information about access levels, go to: <https://softwaresupport.hpe.com/web/softwaresupport/access-levels>.

HPE Software Solutions Now accesses the HPSW Solution and Integration Portal website. This site enables you to explore HPE Product Solutions to meet your business needs, includes a full list of Integrations between HPE Products, as well as a listing of ITIL Processes. The URL for this website is <https://softwaresupport.hpe.com/km/KM01702731>.

Contents

Chapter 1: Demand Management Web Services	30
Overview	30
References	30
Terms and Concepts	31
Field Token	31
Remote Reference	32
URL Reference	32
Data Types	33
Request	34
simpleFields (type SimpleField)	36
tables (type Table)	37
notes (type Note)	38
fieldChangeNotes (type FieldChangeNote)	38
URLReferences (type URLReference)	39
remoteReference	40
Operations	42
getRequests	42
Purpose	42
Function	42
Limitations	43
Related Information	44
Input	44
Return	44
Java Interface	44
Java Examples	45
Errors and Exceptions	46
createRequest	46
Purpose	46
Function	47
Limitations	47
Related Information	48

Input	48
Return	48
Java Interface	48
Java Examples	49
Errors and Exceptions	51
setRequestFields	52
Purpose	52
Function	52
Limitations	53
Related Information	54
Input	54
Return	54
Java Interface	54
Java Examples	54
Errors and Exceptions	55
setRequestRemoteReferenceStatus	56
Purpose	56
Function	57
Limitations	57
Related Information	58
Input	58
Return	58
Java Interface	58
Java Examples	59
Errors and Exceptions	60
addRequestNotes	61
Purpose	61
Function	61
Related Information	62
Input	62
Return	62
Java Interface	62
Java Examples	63
Errors and Exceptions	63
executeWFTTransitions	64

Purpose	64
Function	64
Limitations	64
Related Information	65
Input	65
Return	65
Java Interface	65
Java Examples	66
Errors and Exceptions	66
deleteRequests	67
Purpose	67
Function	67
Limitations	68
Related Information	68
Input	68
Return	68
Java Interface	68
Java Examples	68
Errors and Exceptions	69
Chapter 2: Financial Management (Cost Rules) Web Services	70
Overview	70
References	70
Terms and Concepts	71
Cost Rules	71
Cost Factor	71
Cost Factor Value	71
Data Types	71
WSCostRuleBean	72
CostRateBean	73
WSCostFactorValueBean	74
WSCostFactorBean	75
CostRuleSearchFilter	76
Operations	77
createCostRules	78
Purpose	78

Function	78
Limitations	78
Related Information	78
Input	78
Return	79
Java Interface	79
Java Examples	79
Errors and Exceptions	80
Root Cause Descriptions	80
updateCostRules	83
Purpose	83
Function	83
Limitations	83
Related Information	83
Input	83
Return	84
Java Interface	84
Java Examples	84
Errors and Exceptions	85
Root Cause Descriptions	85
getCostRules	86
Purpose	86
Function	86
Limitations	86
Input	86
Return	86
Java Interface	87
Java Examples	87
Errors and Exceptions	87
Root Cause Descriptions	88
deleteCostRules	88
Purpose	88
Function	88
Related Information	89
Input	89

Return	89
Java Interface	89
Java Examples	89
Errors and Exceptions	90
Root Cause Descriptions	90
searchCostRules	90
Purpose	90
Function	91
Limitations	91
Related Information	91
Input	91
Return	91
Java Interface	91
Java Examples	92
Errors and Exceptions	92
Root Cause Descriptions	92
getCostFactors	93
Purpose	93
Function	93
Limitations	93
Related Information	93
Input	93
Return	93
Java Interface	94
Java Examples	94
Errors and Exceptions	94
Root Cause Descriptions	95
setCostFactors	95
Purpose	95
Function	95
Related Information	96
Input	96
Return	96
Java Interface	96
Java Examples	96

Errors and Exceptions	97
Root Cause Descriptions	97
Chapter 3: Financial Management (Finance Data) Web Services	99
Overview	99
References	99
Data Types	100
FinancialSummaryInfo	100
FinancialDataInfo	101
FinancialSummarySnapshotInfo	102
SnapshotInfo	103
UserDataInfo	104
ApprovedBudgetInfo	105
ForecastActualInfo	106
BenefitInfo	107
FinancialLineCellInfo	108
CostPeriodSumCellInfo	108
BenefitPeriodSumCellInfo	111
ParentInfo	111
FinancialDataParentInfo	112
PeriodInfo	112
AccessListInfo	112
Operations	113
readFinancialSummary	114
Purpose	114
Related Information	114
Input	114
Return	114
Java Examples	114
readFinancialSummarySnapshot	115
Purpose	115
Related Information	115
Input	116
Return	116
Java Examples	116
updateFinancialSummary	116

Purpose	116
Related Information	116
Input	117
Return	117
Java Examples	117
createFinancialSummarySnapshot	117
Purpose	117
Input	117
Return	118
Java Examples	118
readFinancialSummaryACL	118
Purpose	118
Related Information	118
Input	118
Return	118
Java Examples	119
updateFinancialSummaryACL	119
Purpose	119
Input	119
Return	120
Java Examples	120
readFinancialData	120
Purpose	120
Input	120
Return	120
Java Examples	120
updateFinancialData	121
Purpose	121
Related Information	121
Input	121
Return	121
Java Examples	121
createFinancialData	122
Purpose	122
Related Information	122

Input	122
Return	123
Java Examples	123
readFinancialDataACL	123
Purpose	123
Related Information	123
Input	123
Return	124
Java Examples	124
updateFinancialDataACL	124
Purpose	124
Related Information	124
Input	124
Return	125
Java Examples	125
Chapter 4: Portfolio Management Web Services	126
Overview	126
References	126
Operations available in PPM Center	127
Data Types	127
PortfolioWSHeaderBean	127
PortfolioContentItemBean	129
PortfolioProgramBean	129
PortfolioWSBean	130
Operations	131
createPortfolio	132
Purpose	132
Function	132
Related Information	132
Input	132
Return	133
Java Interface	133
Java Examples	133
Errors and Exceptions	134
updatePortfolio	135

Purpose	135
Function	135
Limitations	135
Related Information	136
Input	136
Return	136
Java Interface	136
Java Examples	137
Errors and Exceptions	137
readPortfolio	138
Purpose	138
Function	139
Related Information	139
Input	139
Return	139
Java Interface	139
Java Examples	140
Errors and Exceptions	140
deletePortfolio	141
Purpose	141
Function	141
Related Information	142
Input	142
Return	142
Java Interface	142
Java Examples	143
Errors and Exceptions	143
addPortfolioChildren	144
Purpose	144
Function	144
Limitations	145
Related Information	145
Input	145
Return	145
Java Interface	145

Java Examples	146
Errors and Exceptions	146
removePortfolioChildren	147
Purpose	147
Function	147
Related Information	147
Input	147
Return	148
Java Interface	148
Java Examples	148
Errors and Exceptions	149
addPortfolioPrograms	149
Purpose	149
Function	150
Limitations	150
Related Information	150
Input	150
Return	150
Java Interface	150
Java Examples	151
Errors and Exceptions	151
removePortfolioPrograms	152
Purpose	152
Function	152
Related Information	152
Input	153
Return	153
Java Interface	153
Java Examples	153
Errors and Exceptions	154
addPortfolioContent	155
Purpose	155
Function	155
Limitations	155
Related Information	155

Input	155
Return	155
Java Interface	156
Java Examples	156
Errors and Exceptions	156
removePortfolioContent	158
Purpose	158
Function	158
Related Information	159
Input	159
Return	159
Java Interface	159
Java Examples	160
Errors and Exceptions	160
Chapter 5: Project Management Web Services	162
Overview	162
Data Types	162
ProjectType	163
CreateProjectResultType	165
ProjectInputType	166
WorkPlanInputType	166
AnchorType	166
taskAnchors	167
AddTaskResultType	167
SearchProjectPreferenceType	167
SearchTaskPreferenceType	169
AssignmentType	170
UpdateActualsInput	170
ResourceType	171
TaskType	171
ActivityType	174
ScheduleInfo	174
TaskActualType	175
MoneyInfo	175
CostBean	175

RoleInfo	176
DependencyInfo	176
References	177
Operations	177
createProject	178
Purpose	178
Function	178
Related Information	178
Input	179
Return	179
Java Examples	179
bulkImportProjects	180
Purpose	180
Function	180
Related Information	180
Return	180
Java Examples	181
updateProject	181
Purpose	181
Related Information	181
Input	181
Return	182
Java Examples	182
getProjectDetails	182
Purpose	182
Function	183
Input	183
Return	183
Java Examples	183
executeWorkflowTransition	184
Purpose	184
Input	184
Return	184
Java Examples	184
createWorkPlanFromTemplate	185

Purpose	185
Input	185
Return	185
Java Examples	186
createBlankWorkPlan	186
Purpose	186
Input	186
Return	186
Java Examples	187
importWorkPlanTasks	187
Purpose	187
Function	187
Input	188
Return	188
Java Examples	188
addTasksToExistingWorkPlan	189
Purpose	189
Function	190
Related Information	190
Input	190
Return	190
Java Examples	191
exportWorkPlanFromProject	191
Purpose	191
Function	191
Input	191
Return	191
Java Examples	192
readTasks	192
Purpose	192
Input	192
Return	192
Java Examples	193
updateTaskActuals	193
Purpose	193

Function	193
Input	193
Return	194
Java Examples	194
updateWorkPlanStatus	194
Purpose	194
Function	194
Input	194
Return	195
Java Examples	195
searchProjects	195
Purpose	195
Input	195
Return	196
Java Examples	196
searchTasks	196
Purpose	196
Input	196
Return	197
Java Examples	197
Chapter 6: Program Management Web Services	198
Overview	198
References	198
Operations available in PPM Center	199
Data Types	199
ProgramInfo	200
StatusCodeInfo	201
HealthInfo	202
ContentItemIdentifierInfo	202
ContentItemInfo	203
ContentInfo	203
CostInfo	204
ProgramCostInfo	205
IssueInfo	205
RiskInfo	206

ScopeChangeInfo	207
BusinessObjectiveIdentifierInfo	207
BusinessObjectiveInfo	207
WeightInfo	208
SettingsInfo	208
ProgramSearchFiltersInfo	210
Note	211
Operations	212
createProgram	212
Purpose	212
Function	212
Related Information	213
Input	213
Return	213
Java Examples	213
Errors and Exceptions	214
readProgram	214
Purpose	214
Function	214
Related Information	215
Input	215
Return	215
Java Examples	215
Errors and Exceptions	215
updateProgram	216
Purpose	216
Function	216
Related Information	216
Input	216
Return	217
Java Examples	217
Errors and Exceptions	217
deleteProgram	218
Purpose	218
Function	218

Input	218
Return	219
Java Examples	219
Errors and Exceptions	219
searchProgram	219
Purpose	219
Function	220
Input	220
Return	220
Java Examples	220
Errors and Exceptions	220
Chapter 7: Resource Management Web Services	222
Overview	222
References	222
Data Types	223
ResourceReference	223
RegionReference	224
OrgUnitReference	225
RoleReference	225
SkillReference	226
ResourcePoolReference	226
ResourcePoolAccessControlBean	227
ResourcePool	228
ResourcePoolSearchFilter	230
Role	232
Skill	233
ResourceParticipation	234
ResourceDistributionGroup	235
ResourcePoolDistribution	235
Operations	236
createResourcePools	237
Purpose	237
Function and Parameters	237
Limitations	238
Related Information	238

Input	238
Return	238
Java Interface	239
Java Examples	239
Errors and Exceptions	241
searchResourcePools	243
Purpose	243
Function and Parameters	243
Limitations	244
Input	244
Return	244
Java Interface	244
Java Examples	245
Errors and Exceptions	246
getResourcePools	247
Purpose	247
Function and Parameters	247
Limitations	247
Input	247
Return	248
Java Interface	248
Java Examples	248
Errors and Exceptions	249
updateResourcePools	249
Purpose	249
Function and Parameters	250
Limitations	250
Related Information	251
Input	251
Return	251
Java Interface	251
Java Examples	252
Errors and Exceptions	253
getResourceParticipation	255
Purpose	255

Function and Parameters	255
Related Information	255
Input	255
Return	256
Java Interface	256
Java Examples	256
Errors and Exceptions	257
setResourceParticipation	258
Purpose	258
Function and Parameters	258
Limitations	259
Related Information	259
Input	259
Return	259
Java Interface	259
Java Examples	259
Errors and Exceptions	261
createRoles	263
Purpose	263
Function and Parameters	263
Limitations	263
Input	264
Return	264
Java Interface	264
Java Examples	264
Errors and Exceptions	265
createSkills	266
Purpose	266
Function and Parameters	266
Limitations	267
Input	267
Return	267
Java Interface	267
Java Examples	268
Errors and Exceptions	269

Chapter 8: Resource Management (Staffing Profile) Web Services ..	270
Overview	270
References	270
Data Types	271
StaffingProfileHeaderType	272
ParentEntityType	273
ParentType	273
ParentIdentifier	274
StaffingProfileACLType	274
Entry	274
positionIdentifierType	275
positionLineType	275
effortContourType	277
effortCellType	277
effortSimpleType	278
assignmentType	278
promisedAssignmentType	278
skillProficiencyType	279
userDataType	279
PositionUserDataInfo	281
resultObjectType	288
Operations	289
createStaffingProfileHeader	290
Purpose	290
Function	290
Related Information	290
Input	291
Return	291
Java Interface	291
Java Examples	291
Errors and Exceptions	292
readStaffingProfileHeader	297
Purpose	297
Function	297
Related Information	298

Input	298
Return	298
Java Interface	298
Java Examples	299
Errors and Exceptions	299
updateStaffingProfileHeader	300
Purpose	300
Function	300
Related Information	300
Input	301
Return	301
Java Interface	301
Java Examples	301
Errors and Exceptions	302
readStaffingProfileACL	307
Purpose	307
Function	308
Related information	308
Input	308
Return	308
Java Interface	308
Java Examples	309
Errors and Exceptions	309
updateStaffingProfileACL	310
Purpose	310
Function	310
Related information	310
Input	310
Return	311
Java Interface	311
Java Examples	311
Errors and Exceptions	312
readPositionLines	313
Purpose	313
Function	313

Related information	313
Input	314
Return	314
Java Interface	314
Java Examples	314
Errors and Exceptions	315
addPositionLines	315
Purpose	315
Function	316
Related information	316
Input	316
Return	316
Java Interface	317
Java Examples	317
Errors and Exceptions	317
updatePositionLines	319
Purpose	319
Function	319
Related information	319
Input	320
Return	320
Java Interface	320
Java Examples	320
Errors and Exceptions	321
deletePositionLines	322
Purpose	322
Function	322
Related information	322
Input	323
Return	323
Java Interface	323
Java Examples	323
Errors and Exceptions	324
getPositionIdsforStaffingProfile	325
Purpose	325

Function	325
Input	325
Return	325
Java Interface	325
Java Examples	326
Errors and Exceptions	326
clearStaffingProfileForecastAndAssignmet	327
Purpose	327
Function	327
Limitations	327
Input	328
Return	328
Java Interface	328
Java Examples	328
Errors and Exceptions	329
Chapter 9: Time Management Web Services	330
Overview	330
References	330
Terms and Concepts	331
Time Sheet Policy	331
Period Type	331
Time Period	331
Charge Code	332
Activity	332
Data Types	332
TimeSheetBean	332
Related Information	346
TimeSheetLineBean	348
TimeActualsBean	355
ChargeCodeBean	356
UserDataBean	357
TimeSheetStatus	359
TimeSheetLineStatus	360
TimeSheetPolicyBean	361
TimeSheetSearchCriteriaBean	368

TimeFilter	370
WorkItemActualTime	371
Operations	373
createTimeSheet	374
Purpose	374
Function	374
Input	375
Return	379
Limitations	380
Related Information	380
Java Interface	380
Java Examples	382
Errors and Exceptions	384
updateTimeSheet	386
Purpose	386
Function	387
Input	388
Return	389
Limitations	389
Java Interface	389
Java Examples	391
Errors and Exceptions	393
getTimeSheet	395
Purpose	395
Function	395
Input	395
Return	396
Java Interface	396
Java Examples	396
Errors and Exceptions	397
searchTimeSheets	397
Purpose	397
Function	398
Input	398
Return	398

Limitation	398
Java Interface	398
Java Examples	399
Errors and Exceptions	400
getActualTime	400
Purpose	400
Function	400
Input	400
Return	401
Java Interface	401
Java Examples	402
Errors and Exceptions	403
getTimeSheetPolicy	403
Purpose	403
Function	404
Input	404
Return	404
Java Interface	404
Java Examples	405
Errors and Exceptions	406
submitTimeSheet	407
Purpose	407
Function	408
Input	408
Return	408
Java Interface	408
Java Examples	409
Errors and Exceptions	410
approveTimeSheet	411
Purpose	411
Function	412
Input	412
Return	412
Java Interface	412
Java Examples	413

Errors and Exceptions	414
approveTimeSheetLine	415
Purpose	415
Function	416
Input	416
Return	416
Java Interface	416
Java Examples	417
Errors and Exceptions	419
rejectTimeSheet	420
Purpose	420
Function	421
Input	421
Return	421
Java Interface	421
Java Examples	422
Errors and Exceptions	423
rejectTimeSheetLine	424
Purpose	424
Function	425
Input	425
Return	425
Java Interface	425
Java Examples	426
Errors and Exceptions	427
reworkTimeSheetLine	429
Purpose	429
Function	429
Input	430
Return	430
Java Interface	430
Java Examples	430
Errors and Exceptions	432
freezeTimeSheet	434
Purpose	434

Function	434
Input	434
Return	435
Java Interface	435
Java Examples	435
Errors and Exceptions	436
closeTimeSheet	438
Purpose	438
Function	438
Input	438
Return	439
Java Interface	439
Java Examples	439
Errors and Exceptions	440
cancelTimeSheet	441
Purpose	441
Function	442
Input	442
Return	442
Java Interface	442
Java Examples	443
Errors and Exceptions	444
Chapter 10: Multilingual User Interface Support in Web Services	446
Overview	446
Preferred Language Setting	446
Specifying the Session Language	446
SOAPHeaderCreator	447
Purpose	447
Function	447
Limitations	448
Java Interface	448
Java Example	449
Errors and Exceptions	452
Chapter 11: Web Service Security	453

Overview	453
Authentication	453
Authorization	454
Web Service Security on PPM Server	454
WS-Security Authentication	454
Enable/Disable WS-Security Authentication	454
WS-Security Timestamp	456
WS-Security Encryption	456
HTTP Basic Authentication	457
Web Service Authentication for Web Service Toolkit	458
Specify a User Through Configuration File	458
Set Headers	458
Set Password	459
Specify User	460
HTTP Basic Authentication	460
NTLM Authentication	461
Configure Web Service Client for HTTPS	461
Custom Key Store	462
JDK Default Key Store	462
Web Service Single Sign-On	463
PPM Center Server Configuration	463
Integration with a Client-Side Log-In Module	463
Working with Proxy Servers and Proxy Authentication	465
All Web Service Calls Go Through Proxy: Configure Proxy By Using client-axis2.xml	465
Some Web Service Calls Go Through Proxy: Configure Proxy By Using Java Code	466
Send documentation feedback	468

Chapter 1: Demand Management Web Services

- " Overview" below
- " References " below
- "Terms and Concepts" on the next page
- " Data Types" on page 33
- "Operations" on page 42

Overview

Demand Management Web services provides interfaces for reading, creating, updating, and deleting request entities in Project and Portfolio Management Center (PPM Center). In addition, you can execute workflow transitions on a specified request by using these Web service operations.

You can access all request fields, including user defined fields by using field token, except note, and remote reference.

References

Data types:

`webservice_toolkit\java\conf\xsd\Demand.xsd`

Operations:

`webservice_toolkit\java\conf\wsdl\DemandService.wsdl`

Java sample code:

`webservice_toolkit\java\client\src\examples\dm\DemandServiceClient.java`

.NET sample code:

`webservice_toolkit\MicrosoftDotNet\DemandServiceTest`

Terms and Concepts

This section describes the following three Demand Management terms.

- ["Field Token" below](#)
- ["Remote Reference" on the next page](#)
- ["URL Reference" on the next page](#)

For additional terms and concepts, refer to *Demand Management User's Guide*.

Field Token

The Demand Management Web services identifies request fields with field tokens. A field token is a string with the following components and attributes:

- Alphanumeric characters
- Underscores (_)
- Always upper case
- Unique among all fields of the request type and request header type

When field tokens are used in a Web service operation, a prefix is required to indicate the source or type of the field. Valid prefixes include the following:

Prefix	Description	Example
REQ	Indicates the field is from a request header type.	REQ.DESCRPTION
REQD	Indicates the field is from a request type.	REQD.TABLE_TOKEN
UD	Indicates the field is a user data field.	UD.CUST_FIELD
T	Indicates the field is a column of a table component field.	T. TABLE_TOKEN. COLUMN_TOKEN

Note: In some cases, you will see the prefixes "VP" or "P." For example, REQD.VP.PLATFORM. These are legacy prefixes that were designed to differentiate the display value (with the prefix VP) and the internal code value (with the prefix P). The current Web service implementation will always use the display value instead. As a result, legacy prefixes are ignored.

Remote Reference

A remote reference refers to an item that exists on a remote system, which can be a PPM Server or another type of server.

Remote reference items appear in the **Reference** section of a request. A remote reference is only available for the Web service interface and cannot be created through the PPM Center standard interface. The purpose of the remote reference is to link a request or an item on an external system to a request in the current PPM Center system.

The following example illustrates a remote reference. The hyperlink attached to the name column is pointing to the remote system URL.

- Remote References		
Name	Status	Description
31448	Assigned	This is remote reference through web service.

URL Reference

A URL reference refers to another type of item in the **References** section of a request. You can also create a URL reference by using the PPM Center standard interface.

The following example illustrates a URL reference. The hyperlink attached to the name column is pointing to the referred URL.

- URLs		
Name	Date	Description
www.shopping.hp.com	August 31, 2012 1:16:01 PM CST	HP Shopping Web Site

When you use the `getRequests` operation to obtain the content of a request, references of any type are converted to URL references. For more information, see the operation "[getRequests](#)" on page 42.

Note: For more information about general Demand Management terms and concepts, see the

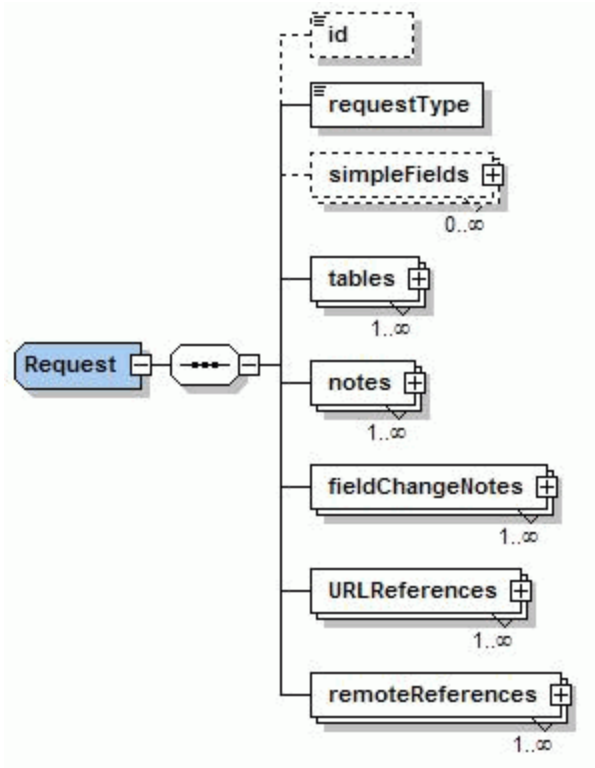
Demand Management User's Guide.

Data Types

Demand Management Web services includes the following data types:

- "Request " on the next page
- "simpleFields (type SimpleField)" on page 36
- "tables (type Table)" on page 37
- "notes (type Note)" on page 38
- "fieldChangeNotes (type FieldChangeNote)" on page 38
- "URLReferences (type URLReference)" on page 39
- "remoteReference" on page 40

Request



Note:

For more information about how to create Request/id, see ["ExampleGetRequests01" on page 45.java](#).

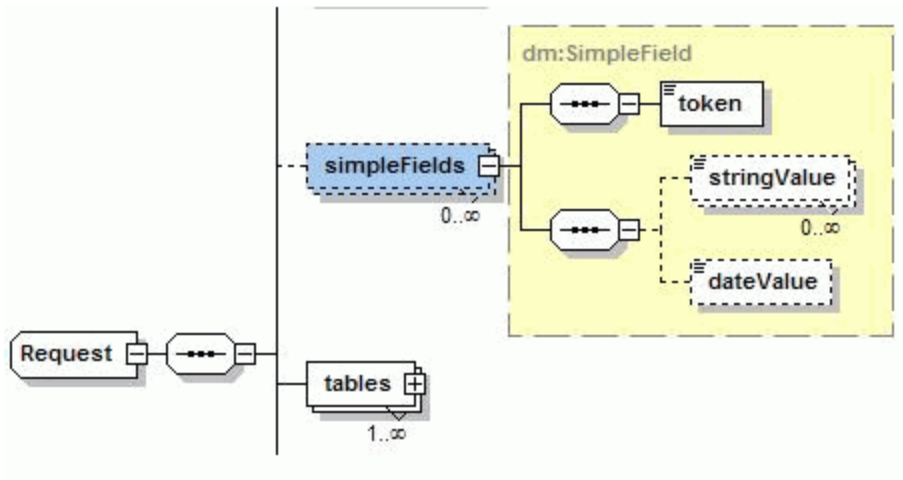
For more information about how to create Request/requestType, see ["ExampleCreateRequest" on page 49.java](#).

Property	Type	Description	Required
Id	string	Request ID. A numeric value, which is typically greater than 30,000.	No ^a
requestType	string	Request type name. This field is case-sensitive.	Yes
simpleFields	SimpleField	An array list of the following elements:	No

Property	Type	Description	Required
		<ul style="list-style-type: none"> token stringValue dateValue 	
tables	Table	An array list of token and columns elements.	No
notes	Note	An array list of the following optional elements: <ul style="list-style-type: none"> author creationDate content 	No
fieldChangeNotes	FieldChangeNote	An array list of the following optional elements: <ul style="list-style-type: none"> author creationDate content The following are required elements: <ul style="list-style-type: none"> fieldPrompt oldValue newValue 	No
URLReferences	URLReference	An array list of the following optional elements: <ul style="list-style-type: none"> addedBy creationDate description name The following is the required element: <ul style="list-style-type: none"> refUR 	No
remoteReferences	RemoteReference	An array list of the following optional elements: <ul style="list-style-type: none"> addedBy creationDate 	No

Property	Type	Description	Required
		<ul style="list-style-type: none"> description name <p>The following are required elements:</p> <ul style="list-style-type: none"> Identifier displayURL status 	
a. Property ID is required for certain operations such as setRequestFields.			

simpleFields (type SimpleField)

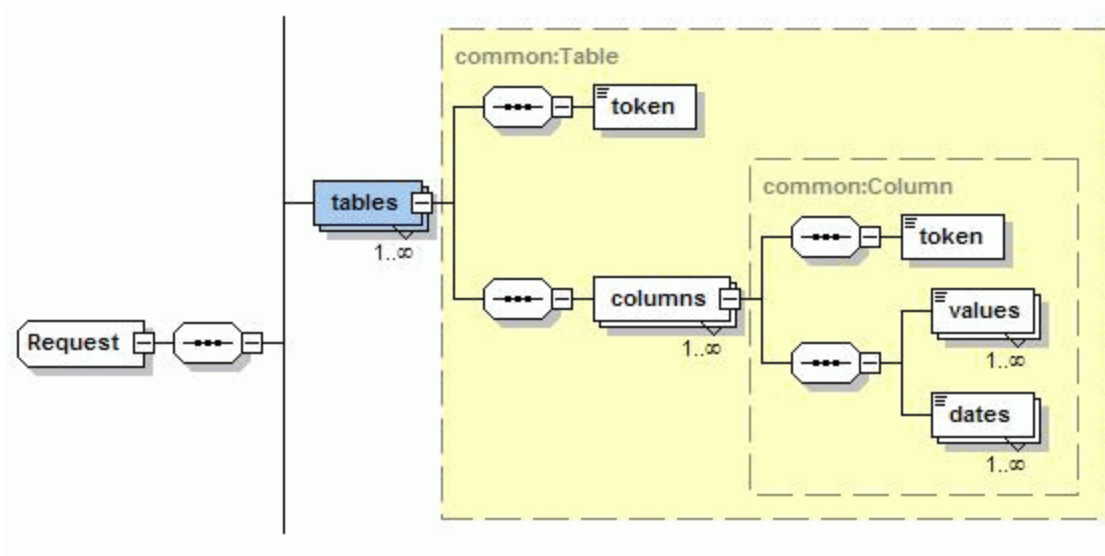


Note: For more information about how to create Request/simpleFields, see ["ExampleCreateRequest" on page 49](#). java.

Property	Type	Description	Required
token	String	A token value to identify the field. A proper prefix is required. See "Field Token" on page 31 for more details.	Yes
stringValue	String	The value that represents the token.	No
dateValue	String	String value of the date field. The date field format is as defined in the XML Schema data type dateTime. For more information, refer to the <i>XML Schema Part 2: Datatypes Second Edition (section 3.2.7)</i> at the W3C Web site:	No

Property	Type	Description	Required
		http://www.w3.org	

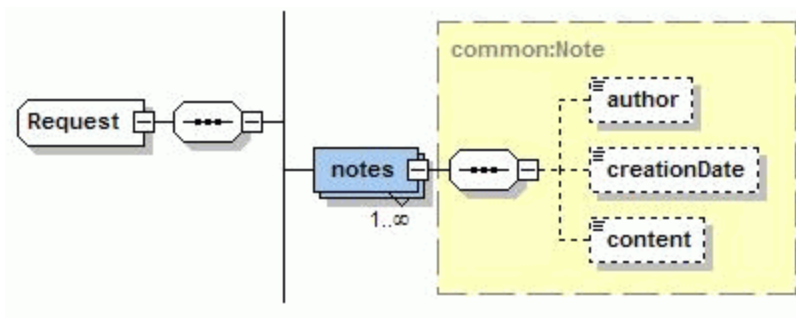
tables (type Table)



Note: For more information about how to create Request/tables, see ["ExampleCreateRequest" on page 49](#).java.

Property	Type	Description	Required
token	String	Token value to identify the field. A proper prefix is required. See "Field Token" on page 31 for more details.	Yes
columns	Column	Array list of token, values, and dates. See the following three properties for details.	Yes
columns /token	String	Token value to identify the field. Proper prefix is required. See "Field Token" on page 31 for more details.	Yes
columns /values	String	Array list of string values of the field. It allows single or multiple values for multi-select fields.	Yes
columns /dates	dateTime	Array list of string values of the field. It allows single or multiple values for multiple selected fields. This XML Schema type dateTime is equivalent to java.util.Calendar.	Yes

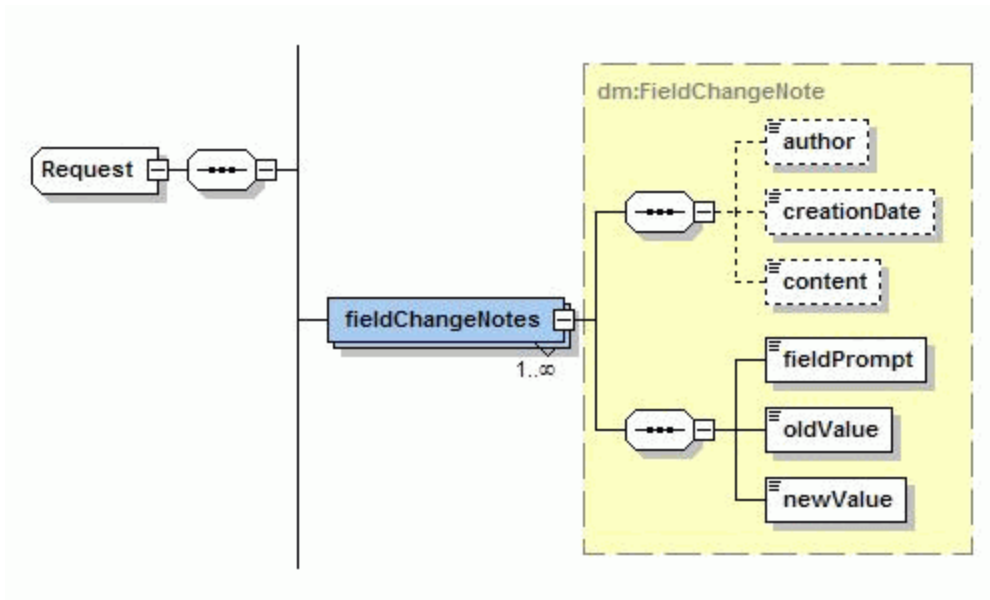
notes (type Note)



Note: For more information about how to create Request/notes, see ["ExampleCreateRequest" on page 49](#).java.

Property	Type	Description	Required
author	string	Author of the note. This field can be any string value and it is not necessary to be a valid user name in PPM Center. This field is shown as "originally added by" in the Note section.	Yes
creationDate	dateTime	Date of the note. Notes are grouped by date when displayed. The XML Schema type dateTime is equivalent to java.util.Calendar.	Yes
content	string	Content of the note.	Yes

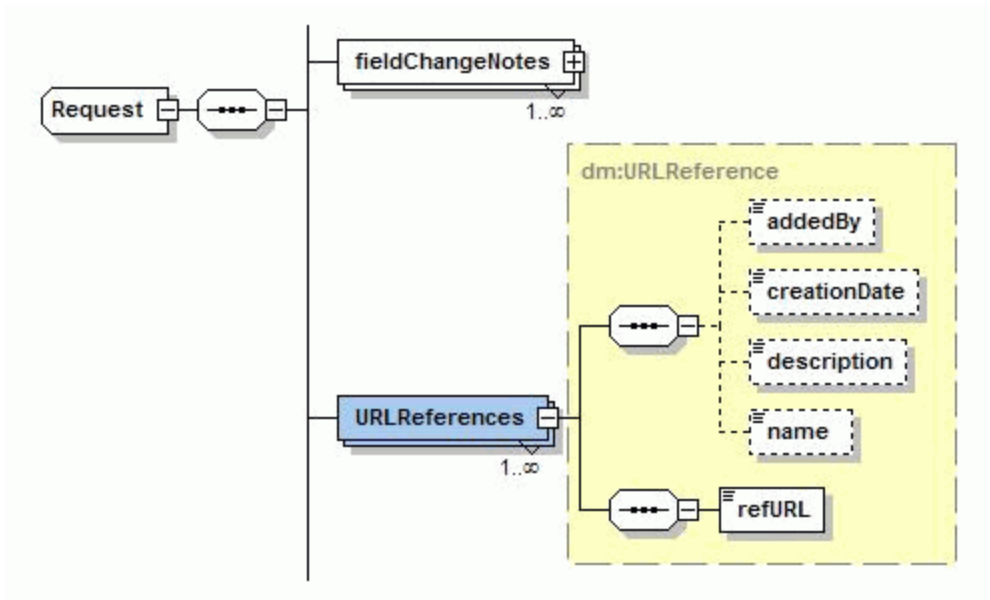
fieldChangeNotes (type FieldChangeNote)



Note: For more information about how to create Request/fieldChangeNotes, see ["ExampleGetRequests01" on page 45 .java](#).

Property	Type	Description	Required
author	string	Author of the note. This field can be any string value and it is not necessary to be a valid user name in PPM Center. This field is shown as "originally added by" in the Note section.	No
creationDate	dateTime	Date of the note. Notes are grouped by date when displayed. The XML Schema type dateTime is equivalent to java.util.Calendar.	No
content	string	Content of the note.	No
fieldPrompt	string	The filed prompt value to be changed.	Yes
oldValue	string	The old value of the field prompt.	Yes
newValue	string	The new value of the field prompt.	Yes

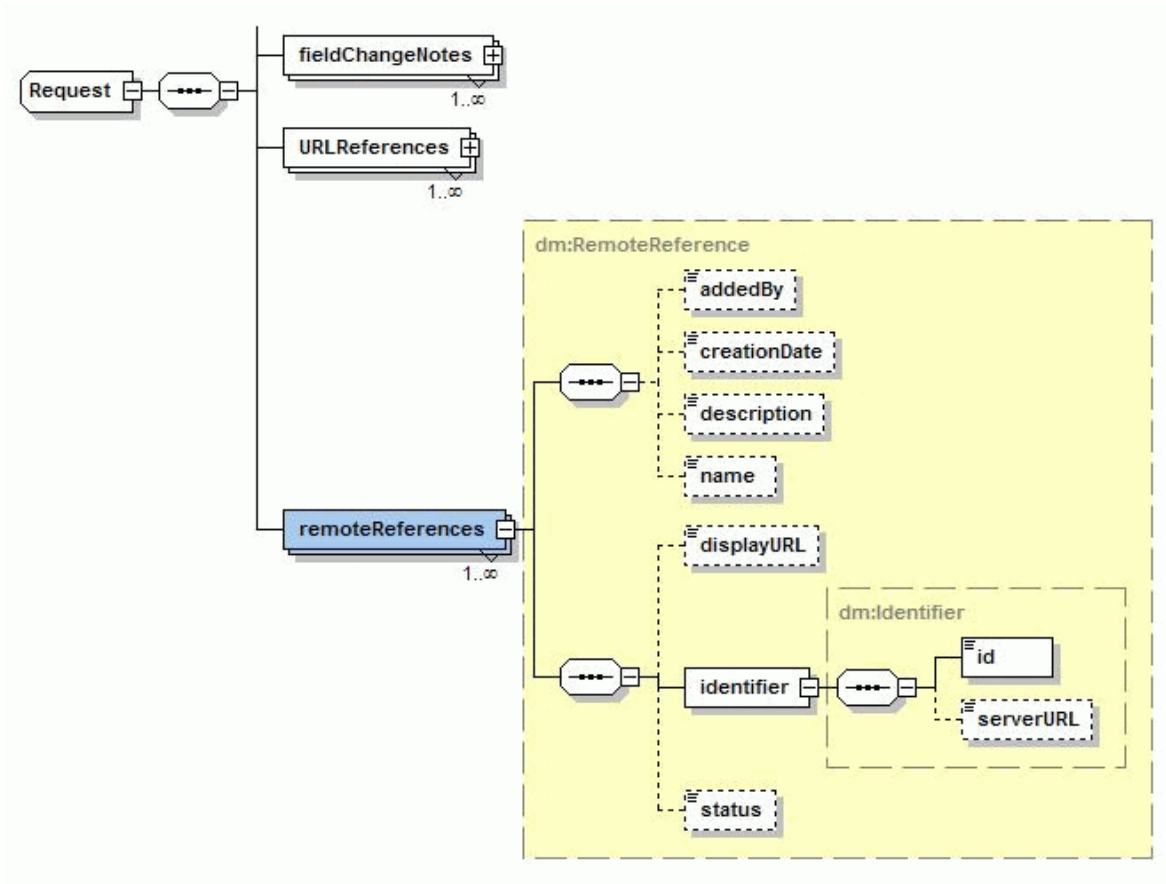
URLReferences (type URLReference)



Note: For more information about how to create Request/URLReferences, see ["ExampleCreateRequest" on page 49](#).java.

Property	Type	Description	Required
addedBy	string	The user that added the URL reference. It can be any string value and it is not necessary to be a valid user name in PPM Center.	No
creationDate	dateTime	Date on which the note is created. The XML Schema type dateTime is equivalent to java.util.Calendar.	No
description	string	The description of the URL reference.	No
name	string	The name of the URL reference.	No
refURL	string	The old value of the field prompt.	Yes

remoteReference



Note: For more information about how to create Request/remoteReferences, see ["ExampleCreateRequest" on page 49](#). java.

Property	Type	Description	Required
addedBy	string	The name of the user who added the remote reference	No
creationDate	dateTime	The date on which the remote reference is added. The XML Schema type dateTime is equivalent to java.util.Calendar.	No
description	string	The description of the remote reference.	No
name	string	The name of the remote reference	No
displayURL	string	The URL of the remote reference	No
identifier	Identifier	The ID and optional serverURL elements	Yes
identifier / id	string	Request ID. A numeric value, which is normally greater than 30,000.	Yes

Property	Type	Description	Required
identifier / serverURL	string	URL of the PPM Server. It is mainly used in request remote reference. This value defaults to an empty string.	Yes
status	string	Status of the reference	Yes

Operations

The following operations are included in Demand Management Web services:

- ["getRequests" below](#)
- ["createRequest" on page 46](#)
- ["setRequestFields" on page 52](#)
- ["setRequestRemoteReferenceStatus" on page 56](#)
- ["addRequestNotes" on page 61](#)
- ["executeWFTransitions" on page 64](#)
- ["deleteRequests" on page 67](#)

getRequests

Purpose

This operation reads the information of a request in PPM Center.

Function

This operation reads the content of a request or a list of requests, identified by request IDs in PPM Center.

The following content of a request can be retrieved through this operation:

- Simple Fields, defined in request header type, request type, or field group if applicable
- Field of Table Component
- Note
- FieldChangeNotes
- References, except remote references

All reference types are converted to URL references. See the following list for details:

- Attachment

URL name is the attachment name, and the URL hyperlink points to the URL to access the attachment download.
- Package Reference

URL name is in the format of "Package <package ID> on localhost," and the URL hyperlink points to the URL to access the package.
- Program Reference

URL name is in the format of "Program <program name> on localhost," and the URL hyperlink points to the URL to access the program.
- Project Reference

URL name is in the format of "Project <project name> on localhost," and the URL hyperlink points to the URL to access the project.
- Task Reference

URL name is in the format of "Task <task name> on localhost," and the URL hyperlink points to the URL to access the task.
- Request Reference

URL name is in the format of "Request <request ID> on localhost," and the URL hyperlink points to the URL to access the request.
- URL Reference

Limitations

This operation has the following limitations:

- Does not read remote references, which can be added to a request when creating the request by using the ["createRequest" on page 46](#) operation.
- For attachments, this operation does not return the attached document, but rather the URL pointing to the document on the PPM server.
- No additional security check. After the user passes authentication, the user can read any request even if the user does not have access to that request.

Related Information

The following data types:

- ["Request " on page 34](#)
- ["fieldChangeNotes \(type FieldChangeNote\)" on page 38](#)
- ["URLReferences \(type URLReference\)" on page 39](#)

Input

A collection of request IDs

Return

A collection of the Request objects. If the specified request ID does not exist in PPM Center, the operation is ignored and no error is generated.

Java Interface

GetRequestsResponseDocument getRequests(GetRequestsDocument in)

Parameters	Description
GetRequestsDocument	Wrapper of the collection of request identifiers, or Identifier[].
GetRequestsResponseDocument	Wrapper of the collection of request objects, or Request[].

Java Examples

Example: get data for a single request.

ExampleGetRequests01

```
public class ExampleGetRequests01{
    . . .
    private void getRequests(String serviceURL, String
requestId) throws Exception {

        // construct the Identifier array
        Identifier[] ids = new Identifier[1];
        ids[0] = Identifier.Factory.newInstance();
        ids[0].setId(requestId);
        // get Webservice handler
        DemandServiceStub stub = new DemandServiceStub(ctx,
serviceURL);
        // Construct message to send
        GetRequestsDocument inDoc =
GetRequestsDocument.Factory.newInstance();
        GetRequestsDocument.GetRequests getRequests =
inDoc.addNewGetRequests();
        getRequests.setRequestIdsArray(ids);
    . . .
}
}
```

Example: get data for a list of request IDs

```
. . .
    private void getRequests(String serviceURL, String[]
requestIds) throws Exception {
        // construct the Identifier array
        Identifier[] ids = new Identifier[requestIds.length];
        for (int i = 0; i < requestIds.length; i++) {
            ids[i] = Identifier.Factory.newInstance();
            ids[i].setId(requestIds[i]);
        }
        // get Webservice handle
        DemandServiceStub stub = new DemandServiceStub(ctx,
serviceURL);
        // Construct message to send
        GetRequestsDocument inDoc =
GetRequestsDocument.Factory.newInstance();
        GetRequestsDocument.GetRequests getRequests =
```

```
inDoc.addNewGetRequests();
    getRequests.setRequestIdsArray(ids);
    . . .
  }
}
```

Errors and Exceptions

Message Code	Message	Cause(s)	Possible Corrective Action
KNTA-11186	Internal error has occurred while calling PPM Web Service. Contact PPM support with the detail information if the problem persists. (KNTA-11186) Details: Missing required element.	You have not set the request ID.	Provide a request ID for the request.
N/A	An error occurred when reading requests Open API Exception 9: Request with ID 30392 was not found in the system. Import was not performed.	The request ID does not exist on the PPM server.	Provide an existing request ID for the request.
KNTA-11186	Internal error has occurred while calling PPM Web Service. Contact PPM support with the detail information if the problem persists. (KNTA-11186) Details: Non digit data was set for number field. Original exception message: For input string: "30362A".	The request ID value you provide is invalid.	The request ID value has to be a numeric value. Correct the value and try again.

createRequest

Purpose

This operation creates a new request in PPM Center.

Function

This operation creates a new request in PPM Center. You can specify the following request contents by using this operation:

- Simple Fields, defined in request header type, request type, or field group, if applicable
- Field of Table Component
- Note
- URL References
- Remote References

During this operation, the system applies the rules defined on the request. The field level rules are applied according to the sequence of the fields in the Web service request. Any field value that you enter will override a rule-derived value.

For simple fields defined as budget, benefit, or staffing profile, the field value is the exact name of the budget, benefit, or staffing profile.

Limitations

This operation has the following limitations:

- Does not support attachment, password field, and references other than URL reference.
- For chained field change rules, this operation does not perform cascading. For example, a rule defined on a field-A change leads to a field-B change, and another rule is defined to change field-C based on a field-B change. In this example, when you set the value for field-A using this operation, field-B will be changed as result of applying the rule, but field-C will not be affected.
- For required fields, the request will be created even the value for required field is not specified.
- There is no security check on the permission. After a user passes authentication, the user can create requests through Web service requests even if the user does not have the 'create request' permission.
- If the specified token of the field does not exist, there will be no error reported. The field will be simply ignored.

Related Information

- ["Request " on page 34](#)
- ["simpleFields \(type SimpleField\)" on page 36](#)
- ["tables \(type Table\)" on page 37](#)
- ["notes \(type Note\)" on page 38](#)
- ["fieldChangeNotes \(type FieldChangeNote\)" on page 38](#)
- ["URLReferences \(type URLReference\)" on page 39](#)
- ["remoteReference" on page 40](#)

Input

The Request object, with all the desired content filled, and with no request ID.

Return

A remote reference to the request.

Java Interface

```
CreateRequestResponseDocument  
createRequest(CreateRequestDocument in)
```

Parameters	Description
CreateRequestDocument	Wrapper of the request object.
CreateRequestResponseDocument	Wrapper of the remote reference to the newly created request.

Java Examples

Example: create a request of type Mybug, which is a copy of the out-of-box request type Bug, with the following changes:

- Adding a table component field on the request type:

Field TOKEN: STAKE_HOLDER

Number of columns: 2

Column 1 TOKEN: ID

Column 2 TOKEN: NAME

- Adding a auto-complete field on the request type, which allows multi-value

Field TOKEN: REVIEWER

Validation: PPM - User Id - Enabled

ExampleCreateRequest

```
public class ExampleCreateRequest{
    . . .
    private void createRequest(String serviceURL, String
requestType) throws Exception {

        // Request component
        Request req = Request.Factory.newInstance();
        req.setRequestType(requestType);
        SimpleField[] fields = new SimpleField[6];

        // simpleFields component

        // Set field 'Description'
        SimpleField field_A = SimpleField.Factory.newInstance();
        field_A.setToken("REQ.DESCRPTION");
        field_A.setStringValue1Array(new String[] { "WebService
Test" });
        fields[0] = field_A;

        // Set field 'Department'
        SimpleField field_B = SimpleField.Factory.newInstance();
        field_B.setToken("REQ.DEPARTMENT_NAME");
        field_B.setStringValue1Array(new String[] { "Finance"
});
        fields[1] = field_B;
```

```
// Set field 'Module'
SimpleField field_C = SimpleField.Factory.newInstance();
field_C.setToken("REQD.VP.MODULE");
field_C.setStringValue1Array(new String[] { "Module A"
});
fields[2] = field_C;

// Set field 'Platform'
SimpleField field_D = SimpleField.Factory.newInstance();
field_D.setToken("REQD.VP.PLATFORM");
field_D.setStringValue1Array(new String[] { "Unix" });
fields[3] = field_D;

// Set field 'Impact'
SimpleField field_E = SimpleField.Factory.newInstance();
field_E.setToken("REQD.VP.IMPACT");
field_E.setStringValue1Array(new String[] { "Warning"
});
fields[4] = field_E;

// Set field 'Reviewer', which allow multi-values
SimpleField field_F = SimpleField.Factory.newInstance();
field_F.setToken("REQD.REVIEWER");
field_F.setStringValue1Array(new String[] { "admin",
"userx" });
fields[5] = field_F;

// Add all the fields to request object
req.setSimpleFieldsArray(fields);

// tables component
Table t = req.addNewTables();
t.setToken("REQD.STAKE HOLDER"); // token of the field
Column c = t.addNewColumns();
c.setToken("T.STAKE HOLDER.ID"); // token of the column
c.setValuesArray(new String[] { "311", "312" }); //
value array
c = t.addNewColumns();
c.setToken("T.STAKE HOLDER.NAME");
c.setValuesArray(new String[] { "User1", "User2" });

// notes component
Note note = req.addNewNotes();
note.setAuthor("admin");
note.setContent("WebService Test Note");
note.setCreationDate(Calendar.getInstance());

// URLReferences component
URLReference refURL = req.addNewURLReferences();
refURL.setAddedBy("admin");
refURL.setCreationDate(Calendar.getInstance());
refURL.setDescription("This is a reference created thru
```

```

web service");
    refURL.setName("Reference URL");
    refURL.setRefURL("http://www.ref.com");

    // remoteReference component
    RemoteReference ref = req.addNewRemoteReferences();
    ref.setAddedBy("admin");
    ref.setCreationDate(Calendar.getInstance());
    ref.setDescription("This is a reference created thru web
service");
    ref.setName("Ticket#1234");
    ref.setDisplayURL("http://www.display.com");
    Identifier refId = Identifier.Factory.newInstance();
    refId.setId("31234");
    refId.setServerURL("http://server:port");
    ref.setIdentifier(refId);
    ref.setStatus("Assigned");

    // Get web service
    DemandServiceStub stub = new DemandServiceStub(ctx,
serviceURL);
    // Construct message to send
    CreateRequestDocument inDoc =
CreateRequestDocument.Factory.newInstance();
    CreateRequestDocument.CreateRequest createRequest =
inDoc.addNewCreateRequest();
    createRequest.setRequestObj(req);
    . . .
}
}

```

Errors and Exceptions

Message Code	Message	Cause(s)	Possible Corrective Action
N/A	Missing required element " {http://mercury.com/ppm/dm/1.0}requestType"(line - 1, col -1, in SOAP-message)	Missing Request type.	Check and provide the right request type.

setRequestFields

Purpose

This operation updates one or more simple fields in a request in PPM Center.

Function

This operation updates the content of a request, identified by the request ID in PPM Center. You can update the following fields of a request through this operation:

- Simple Field from request header type
- Simple Field from request type
- Simple Field from Field Group

Exception: The following field is not updatable once the request is created. Therefore, you cannot update them using this operation:

- REQ.CREATED_BY_NAME
- REQ.CREATED_BY_USERNAME
- REQ.CREATED_BY_EMAIL
- REQ.CREATION_DATE
- REQ.CREATED_BY
- REQ.LAST_UPDATED_BY_USERNAME
- REQ.LAST_UPDATED_BY_EMAIL
- REQ.LAST_UPDATE_DATE
- REQ.LAST_UPDATED_BY
- REQ.REQUEST_TYPE_NAME
- REQ.REQUEST_TYPE_ID

- REQ.REQUEST_ID
- REQ.WORKFLOW_NAME
- REQ.WORKFLOW_ID
- REQ.STATUS_NAME
- REQ.STATUS_ID
- REQ.CONTACT_PHONE_NUMBER
- REQ.CONTACT_EMAIL
- REQD.CREATION_DATE
- REQD.LAST_UPDATE_DATE
- REQD.WORKFLOW_ID
- REQD.CREATED_BY
- REQD.REQUEST_ID
- REQD.LAST_UPDATED_BY
- SYS.USERNAME
- SYS.USER_ID

During this operation, the rules defined on the fields are applied, in the order of the fields' appearance in the Web service request. Any field in which you enter a value will override the rule-derived value.

To update field with type of budget, benefit, or staffing profile, the name of the entity is the field value.

Limitations

This operation has the following limitations:

- Does not support updates of fields with attachments, password fields, and table component fields. Table component fields can be set only when you create a request. Once the request has been created, the Web service interface does not support update of table component fields.
- No additional security check including participant or field level access check on this operation. After the request passes the Web service authentication, the operation will try the update with no further security check.

Note: For more information, see:

- `addRequestNotes` - add new note to existing request
- `setRequestRemoteReferenceStatus` - update the status in remote reference
- `executeWFTransitions` - execute work flow transition for the request

Related Information

The following data type:

- ["simpleFields \(type SimpleField\)" on page 36](#)

Input

A collection of changed fields, and the ID of the request.

Return

Upon success, the operation returns the ID of the request.

Java Interface

`SetRequestFieldsResponseDocument`

`setRequestFields(SetRequestFieldsDocument in)`

Parameters	Description
<code>SetRequestFieldsDocument</code>	Wrapper of the collection of request fields, or <code>SimpleField []</code> .
<code>SetRequestFieldsResponseDocument</code>	Wrapper of the return code of this operation.

Java Examples

Example: update several simple fields of an existing request.

```

public class ExampleSetRequestFields {
    . . .
    private void setRequestFields(String serviceURL, String
requestId) throws Exception {
    . . .
        // create the array of changed fields
        SimpleField[] fields = new SimpleField[2];

        // Set field 'Description'
        SimpleField field_A = SimpleField.Factory.newInstance();
        field_A.setToken("REQ.DESCRPTION");
        field_A.setStringValue1Array(new String[] { "WebService
Test (Update)" });
        fields[0] = field_A;

        // Set field 'Department'
        SimpleField field_B = SimpleField.Factory.newInstance();
        field_B.setToken("REQ.DEPARTMENT_NAME");
        field_B.setStringValue1Array(new String[] { "IS" });
        fields[1] = field_B;

        // Get web service
        DemandServiceStub stub = new DemandServiceStub(ctx,
serviceURL);
        // Construct message to send
        SetRequestFieldsDocument inDoc =
SetRequestFieldsDocument.Factory.newInstance();
        SetRequestFieldsDocument.SetRequestFields
setRequestFields = inDoc
            .addNewSetRequestFields();
        setRequestFields.setRequestId(reqId);
        setRequestFields.setFieldsArray(fields);
    . . .
    }
}

```

Errors and Exceptions

Message Code	Message	Cause(s)	Possible Corrective Action
KNTA-11186	ry.com/ppm/common/1.0}token"(line -1, col -1, in SOAP-message)	You have not provided the token.	Provide a token name to update.

Message Code	Message	Cause(s)	Possible Corrective Action
N/A	An error occurred when updating request fields- java.lang.Exception: Error 2: Error validating value "ISSP" for field REQ.DEPARTMENT_NAME: no valid data was found for drop-down list.Error 2: Error validating value "ISSP" for field REQ.DEPARTMENT_NAME: no valid data was found for drop-down list.	You have provided an invalid value for the token.	Provide a valid token value for update.
N/A	Open API Exception 4: The Request could not be locked because another party currently holds the lock.	Update collision.	Retry the operation.
N/A	An error occurred when updating request fields- java.lang.NumberFormatException: For input string: "30361A	You have provided an invalid request ID.	Provide a valid request ID.
KNTA-11186	y.com/ppm/dm/service/1.0}requestId	You have not provided the request ID.	Provide the request ID.
N/A	Error in loadAndLockRequest Open API Exception 9: Request with ID 30391 was not found in the system. Import was not performed.	The request ID you provided does not exist in the system.	Provide an existing request ID.

setRequestRemoteReferenceStatus

Purpose

This operation updates the status of a remote reference in a request in PPM Center, as illustrated in the following example:

References		
Remote References		
Name	Status	Description
✘ 31448	Assigned	This is remote reference through web service.

Function

This operation provides the following functions.

- Update the status of the specified remote reference (the Source), in a PPM request (the Receiver). Remote reference is identified by the reference ID and server URL.
- Execute the workflow transition, if the given value of 'Status' exactly matches the transition name as appeared on the action buttons (case sensitive). This function is supported for backward compatibility. If the status does not match any of the transition names, no workflow transition will be performed. If you only want to perform workflow transition, use the ["executeWFTransitions" on page 64](#) operation.
- Update the simple fields of the request. This function is provided for backward compatibility. It is recommended to use the ["setRequestFields" on page 52](#) operation to update simple fields.

Limitations

This operation has the following limitations:

- Only updates the status field of the remote reference. There is no operation to update the other remote reference fields.
- No security check on workflow transition. Even though the user cannot perform the transition from PPM Center standard interfaces, the transition will be performed with this Web service call, as long as the status value matches the transition name.
- Even when there is no simple field change, a simple field collection is still required when you call this operation.

Note: For more information, see:

- `setRequestFields` - set the value for simple fields in the request.
- `executeWFTransitions` - execute work flow transition for the request.

Related Information

The following data types:

- ["Request " on page 34](#)
- ["simpleFields \(type SimpleField\)" on page 36](#)
- ["remoteReference" on page 40](#)

Input

The following inputs are required for this operation:

- Receiver ID, which is the Identifier for the request in PPM Center.
- Source ID, which is to identify the remote reference. ServerURL is required.
- Status
- Simple field array. Even if there is no update on simple fields, it is still required.

Return

Upon success, the operation returns the ID of the request.

Java Interface

```
SetRequestRemoteReferenceStatusResponseDocument  
setRequestRemoteReferenceStatu(SetRequestRemoteReferenceStatusDocument in)
```

Parameters	Description
SetRequestRemoteReferenceStatusDocument	Wrapper of the request ID, source ID,

Parameters	Description
	status, and the SimpleField[].
SetRequestRemoteReferenceStatusResponseDocument	Wrapper of the return code of this operation.

Java Examples

Example: update a remote reference of an existing request.

```
public class ExampleSetRequestFields {
    . . .
    private void setRequestFields(String serviceURL, String
requestId) throws Exception {
    . . .
        // create the array of changed fields
        SimpleField[] fields = new SimpleField[2];

        // Set field 'Description'
        SimpleField field_A = SimpleField.Factory.newInstance();
        field_A.setToken("REQ.DESCRPTION");
        field_A.setStringValue1Array(new String[] { "WebService
Test (Update)" });
        fields[0] = field_A;

        // Set field 'Department'
        SimpleField field_B = SimpleField.Factory.newInstance();
        field_B.setToken("REQ.DEPARTMENT_NAME");
        field_B.setStringValue1Array(new String[] { "IS" });
        fields[1] = field_B;

        // Get web service
        DemandServiceStub stub = new DemandServiceStub(ctx,
serviceURL);
        // Construct message to send
        SetRequestFieldsDocument inDoc =
SetRequestFieldsDocument.Factory.newInstance();
        SetRequestFieldsDocument.SetRequestFields
setRequestFields = inDoc
            .addNewSetRequestFields();
        setRequestFields.setRequestId(reqId);
        setRequestFields.setFieldsArray(fields);
    . . .
    }
}
```

Errors and Exceptions

Message Code	Message	Cause(s)	Possible Corrective Action
KNTA-11186	Internal error has occurred while calling PPM Web Service. Contact PPM support with the detail information if the problem persists. (KNTA-11186) Details: Missing required element {http://mercury.com/ppm/dm/service/1.0} fields	Missing simple fields.	Although this operation is not designed to update the request simple fields, a simple field array is required when calling this operation. Refer to the example provided in this section for how to add an empty simple field array.
N/A	Internal error has occurred while calling PPM Web Service. Contact PPM support with the detail information if the problem persists. (KNTA-11186) Details: Missing required element " {http://mercury.com/ppm/common/1.0} token"(line -1, col -1, in SOAP-message)	Missing token.	For a simple field, the token cannot be NULL. If for any reason we have to leave the token empty (for example, in case we have to supply an empty simple field array), we should set the token to an empty string.
N/A	An error occurred when updating remote reference status -java.lang.Exception: Error 3: Error applying remote reference: cannot apply a remote reference with NULL server URL	Missing ServerURL.	For identifying a remote reference, the ServerURL field of the Identifier of the remote reference is required.
KNTA-11186	//mercury.com/ppm/dm/1.0}id" start tag, found "{http://mercury.com/ppm/dm/1.0} serverURL" start tag (line -1, col -1, in SOAP-message).	You have not provided the request ID.	Provide the request ID.
N/A	Internal error has occurred while calling PPM Web Service. Contact PPM support with the detail information if the problem persists. (KNTA-11186) Details: Missing required element {http://mercury.com/ppm/dm/service/1.0} receiver	You have not set the receiver ID.	Provide the request ID for the receiver.
N/A	Internal error has occurred while calling PPM Web Service. Contact PPM support	You have not set the	Provide the Identifier with ID and server URL.

Message Code	Message	Cause(s)	Possible Corrective Action
	with the detail information if the problem persists. (KNTA-11186) Details: Missing required element {http://mercury.com/ppm/dm/service/1.0} source	Source.	
N/A	//mercury.com/ppm/dm/1.0}id" start tag, found "{http://mercury.com/ppm/dm/1.0} serverURL" start tag (line -1, col -1, in SOAP-message).	You have not provide the request ID.	Provide the reference request ID.
N/A	An error occurred when updating remote reference status -java.lang.Exception: Error 3: Error applying remote reference: cannot apply a remote reference with NULL server URL	You have not set the Server URL.	Provide the server URL for the identifier.
N/A	Internal error has occurred while calling PPM Web Service. Contact PPM support with the detail information if the problem persists. (KNTA-11186) Details: Missing required element {http://mercury.com/ppm/dm/service/1.0} status	You have not set the Status.	Provide the status for the remote reference.

addRequestNotes

Purpose

This operation adds one or more notes into a request in PPM Center.

Function

This operation adds one or more notes into a specified request. The following fields can be specified for each note:

- Author, which can be any string value and is displayed in the **originally addedby** entry. The value for the **Note Author** field is the user account used when you call the web service server.
- Creation Date
- Content

The notes are grouped into a daily bucket and displayed in date order. Request note cannot be edited or deleted once added.

Related Information

The following data type:

- ["notes \(type Note\)" on page 38](#)

Input

The following inputs are required for this operation:

- Request ID
- Collection of one or more notes

Return

Upon success, the operation returns the ID of the request.

Java Interface

```
AddRequestNotesResponseDocument  
addRequestNotes(AddRequestNotesDocument in)
```

Parameters	Description
AddRequestNotesDocument	Wrapper of the request ID and the Note[].
AddRequestNotesResponseDocument	Wrapper of the return code of this operation.

Java Examples

Example: add two notes to an existing request.

```

    . . .
    private void addRequestNotes(String serviceURL, String
    requestId) throws Exception {
    . . .
        // construct the note array
        Note[] notes = new Note[2];
        notes[0] = Note.Factory.newInstance();
        notes[0].setAuthor("Admin");
        notes[0].setCreationDate(Calendar.getInstance());
        notes[0].setContent("Note 1 added thru web service");
        notes[1] = Note.Factory.newInstance();
        notes[1].setAuthor("Admin");
        notes[1].setCreationDate(Calendar.getInstance());
        notes[1].setContent("Note 2 added thru web service");
        // get Webservice handle
        DemandServiceStub stub = new DemandServiceStub(ctx,
    serviceURL);
        // Construct message to send
        AddRequestNotesDocument inDoc =
    AddRequestNotesDocument.Factory.newInstance();
        AddRequestNotesDocument.AddRequestNotes addNotes =
    inDoc.addNewAddRequestNotes();
        addNotes.setRequestId(reqId);
        addNotes.setNotesArray(notes);
    . . .
    }
    }
    
```

Errors and Exceptions

Message Code	Message	Cause(s)	Possible Corrective Action
KNTA-11186	Internal error has occurred while calling PPM Web Service. Contact PPM support with the detail information if the problem persists. (KNTA-11186) Details: Missing required element {http://mercury.com/ppm/dm/service/1.0}requestId	Missing request ID	Add the required request ID.

Message Code	Message	Cause(s)	Possible Corrective Action
N/A	An error occurred when adding notes to request Error in loadAndLockRequest Open API Exception 9: Request with ID 30391 was not found in the system. Import was not performed	Request ID not found	Provide an existing request ID.
N/A	java.lang.NullPointerException	Missing author, creation date or content	Provide complete data (author, creation date and content).
N/A	java.lang.IllegalArgumentException: Array element null	Note array element is not initialized or is null	Initialize array element with a Note object that has complete data.

executeWFTransitions

Purpose

This operation executes the workflow transition on a request in PPM Center.

Function

This operation executes the workflow transition on a request.

Limitations

This operation has the following limitations:

- Supports only the decision step. It does not support the execution step.
- Does not support branching; it will fail if there is a branch on the same transition.
- No security check on workflow transition. Even though the user may not be allowed to perform the transition from PPM standard interfaces, or the workflow step may require reauthentication, the transition will be performed by this Web service call, as long as the status value matches the transition name.

Related Information

The following data type:

- ["Request " on page 34](#)

Input

The following inputs are required for this operation:

- Receiver ID, which is the identifier for the request in PPM Center.
- Transition name, which should exactly match the value on the action buttons (case sensitive).

Return

When the transition is valid and executed, the operation returns the following string:

```
Workflow transition was successful.
```

When the transition is invalid, the operation returns the following string:

```
There were no matches for the transition name provided.
```

Java Interface

```
ExecuteWFTransitionsResponseDocument  
executeWFTransitions(ExecuteWFTransitionsDocument in)
```

Parameters	Description
ExecuteWFTransitionsDocument	Wrapper of the request ID and the transition name.
ExecuteWFTransitionsResponseDocument	Wrapper of the return code of this operation.

Java Examples

Example: execute a workflow transition on an existing request.

```
public class ExampleSetExecuteWFTransitions {
    . . .
    private void executeWFTransitions(String serviceURL, String
    requestId,
        String transition) throws Exception {

        // construct the Identifiers
        Identifier receiverId =
    Identifier.Factory.newInstance();
        receiverId.setId(requestId);

        // get Webservice handle
        DemandServiceStub stub = new DemandServiceStub(ctx,
    serviceURL);

        // Construct message to send
        ExecuteWFTransitionsDocument inDoc =
    ExecuteWFTransitionsDocument.Factory
        .newInstance();
        ExecuteWFTransitionsDocument.ExecuteWFTransitions exec =
    inDoc
        .addNewExecuteWFTransitions();
        exec.setReceiver(receiverId);
        exec.setTransition(transition);

        . . .
    }
}
```

Errors and Exceptions

Message Code	Message	Cause(s)	Possible Corrective Action
N/A	Update remote reference status Succeeded.	The transition	Provide a

Message Code	Message	Cause(s)	Possible Corrective Action
	Return code: There were no matches for the transition name provided.	has been provided but there are no matches for the transition name.	matching transition for the request.
KNTA-11186	Internal error has occurred while calling PPM Web Service. Contact PPM support with the detail information if the problem persists. (KNTA-11186) Details: Missing required element {http://mercury.com/ppm/dm/service/1.0}transition	You have not provided the transition.	Provide the transition for the request
KNTA-11186	Internal error has occurred while calling PPM Web Service. Contact PPM support with the detail information if the problem persists. (KNTA-11186) Details: Missing required element {http://mercury.com/ppm/dm/service/1.0}receiver	You have not set the request ID.	Provide the request ID for the request.
KNTA-11186	Internal error has occurred while calling PPM Web Service. Contact PPM support with the detail information if the problem persists. (KNTA-11186) Details: null; nested exception is:java.lang.NullPointerException	The request ID value is invalid.	The request ID value has to be a numeric value. Correct the value and try again.

deleteRequests

Purpose

This operation deletes one or more requests in PPM Center.

Function

This operation is to delete one or more requests in PPM Center.

Limitations

No security check on this operation. The specified request(s) will be deleted by this Web service operation even if the user does not have permissions to delete the request.

Related Information

The following data type:

- ["Request " on page 34](#)

Input

A collection of request IDs.

Return

Return number of requests deleted.

Java Interface

```
DeleteRequestsResponseDocument  
deleteRequests(DeleteRequestsDocument in)
```

Parameters	Description
DeleteRequestsDocument	Wrapper of a list of request ID, or Identifier[].
DeleteRequestsResponseDocument	Wrapper of the return code of this operation.

Java Examples

Example: update a remote reference of an existing request.

```

public class ExampleDeleteRequests {
    . . .
    private void deleteRequests(String serviceURL, String[]
requestIds) throws Exception {
        // construct the Identifier array
        Identifier[] ids = new Identifier[requestIds.length];
        for (int i = 0; i < requestIds.length; i++) {
            ids[i] = Identifier.Factory.newInstance();
            ids[i].setId(requestIds[i]);
        }

        // get Webservice handle
        DemandServiceStub stub = new DemandServiceStub(ctx,
serviceURL);
        // Construct message to send
        DeleteRequestsDocument inDoc =
DeleteRequestsDocument.Factory.newInstance();
        DeleteRequestsDocument.DeleteRequests deletes =
inDoc.addNewDeleteRequests();
        deletes.setRequestIdsArray(ids);
    . . .
    }
}

```

Errors and Exceptions

Message Code	Message	Cause(s)	Possible Corrective Action
KNTA-11186	Internal error has occurred while calling PPM Web Service. Contact PPM support with the detail information if the problem persists. (KNTA-11186) Details: Non digit data was set for number field. Original exception message: For input string: "30364A".	The request ID is invalid (includes non-numeric data).	Provide a valid request ID.
KNTA-11186	Internal error has occurred while calling PPM Web Service. Contact PPM support with the detail information if the problem persists. (KNTA-11186) Details: Non digit data was set for number field. Original exception message: For input string: ""	The request ID is null.	Provide a valid request ID.
N/A	Delete Requests Succeeded. Return Code: 0	The request ID is not found in the system.	Provide an existing request ID.

Chapter 2: Financial Management (Cost Rules) Web Services

- ["Overview" below](#)
- ["References" below](#)
- ["Terms and Concepts" on the next page](#)
- ["Data Types" on the next page](#)
- ["Operations" on page 77](#)

Overview

Financial Management Web services for cost rules provides operations to create, update, read, and delete cost rules. Operations to set and get cost factors in PPM Center are also supported.

Cost rules are used to assign costs to various units of work, such as the time spent working on a project task. The applicability of a cost rule to a given transaction is determined by its cost factors. A cost rule may have different rates for one or more non-overlapping periods of time. Exactly one rule in the system is the default rule. The default rule may not be deleted and its set of cost factor values may not be altered.

Note: For more information about general Financial Management terms and concepts, see the *Financial Management User's Guide*.

References

Data types definition:

`webservice_toolkit\java\conf\wsdl60\Finance.xsd`

Operations definition:

`webservice_toolkit\java\conf\wsdl\FinanceService.wsdl`

Java sample code:

`webservice_toolkit\java\client\src\examples\fm`

.NET sample code:

```
webservice_toolkit\MicrosoftDotNet\FinancialSummaryTest
```

Terms and Concepts

Cost Rules

Cost rules encapsulate the business logic that determines how the labor cost is calculated.

Cost Factor

Cost factors are the factors that determine cost rates. The same factors generally determine cost rates in any organization. These common cost factors are Time period, Region, Role, Skills, specific Resource, and so on.

Cost Factor Value

A cost factor value is the specific value for a given cost factor."

For example, for the **Department** cost factor, the cost factor value could be "Finance," whereas for the **Region** cost factor, the value could be "America."

Data Types

Financial Management Web services for cost rules includes the following data types:

- ["WSCostRuleBean" on the next page](#)
- ["CostRateBean" on page 73](#)
- ["WSCostFactorValueBean" on page 74](#)

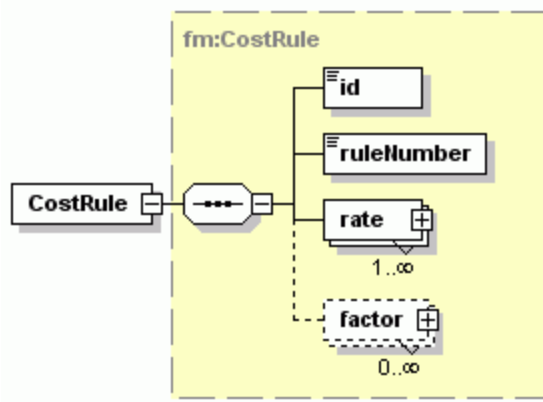
- ["WSCostFactorBean" on page 75](#)
- ["CostRuleSearchFilter" on page 76](#)

WSCostRuleBean

WSCostRuleBean is a value object representing a cost rule. This bean is used to create cost rules in the system along with cost rate and cost factor values.

Note: For more information about the WSCostRuleBean, see:

- WSCostRuleBean is used as the INPUT in the following operations:
 - Operation ["createCostRules" on page 78](#)
 - Operation ["updateCostRules" on page 83](#)
- WSCostRuleBean is used as the OUTPUT in the following operations:
 - Operation ["createCostRules" on page 78](#)
 - Operation ["getCostRules" on page 86](#)
 - Operation ["searchCostRules" on page 90](#)

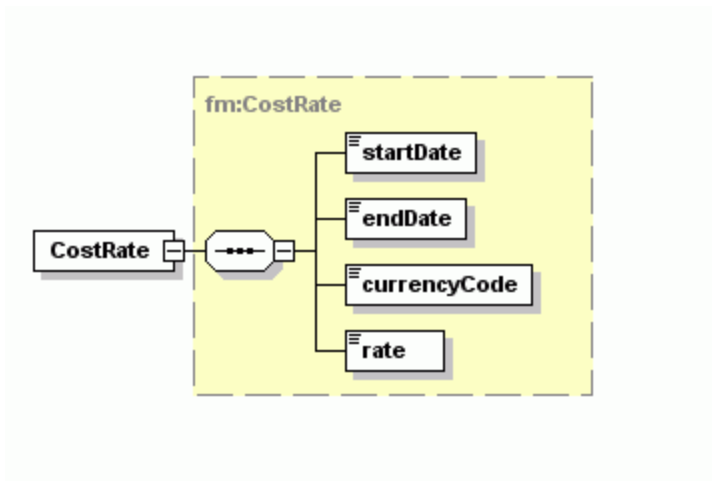


Property	Type	Description	Required	Default
Id	Long	Numeric value that is typically greater than 30,000. This value is automatically generated on the creation of a Cost Rule object in PPM Center.	No	N/A

Property	Type	Description	Required	Default
ruleNumber	int	Numeric value used to identify the cost rule. If you do not provide a value, the value is automatically incremented.	No	N/A
costRates	List	List of "CostRateBean" belows that contains the effective dates, rate, and currency code.	Yes	N/A
costFactorValues	List	list of "WSCostFactorValueBean" on the next pages that contains the cost factors values for the cost rule.	No	N/A

CostRateBean

CostRateBean is a value object representing the time-phased rate of a cost rule. A cost rule may have different rates for one or more non-overlapping periods of time.



Property	Type	Description	Required	Default
effectiveStartDate	Date	Date value that specifies the start date for the cost rate. The value can be blank to indicate that the effective date is from way before. Note that the effective dates should not be overlapping with the effective dates of other cost rates that exist for this rule.	No	N/A

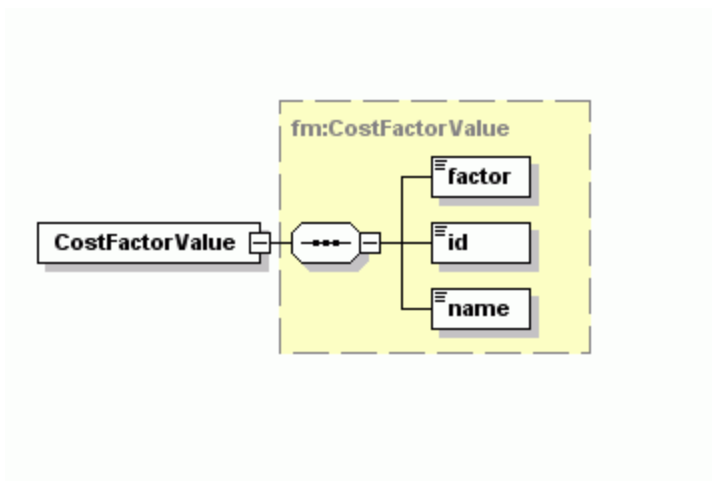
Property	Type	Description	Required	Default
effectiveEndDate	Date	Date value that specifies the end date for the cost rate. The value can be blank to indicate that there is no end date. Note that the effective dates should not be overlapping with the effective dates of other cost rates that exist for this rule.	No	N/A
currencyCode	String	Currency code to be used with this cost rate. For example, the USD currency code.	Yes	N/A
rate	BigDecimal	This property is the actual numeric value of the rate for this rule. This property is used in association with the currency code.	Yes	N/A

WSCostFactorValueBean

WSCostFactorValueBean is a value object that specifies which cost factor is used for the cost rule.

This value object also specifies the ID and the name associated with the cost factor.

For example, if the cost factor used for the cost rule is Resource, you should specify the resource ID or resource name on which the cost rule should be applied.



Property	Type	Description	Required	Default
costFactorName	String	Cost factor that is already set for the cost rules in the system. Both the id property and the name	Yes	N/A

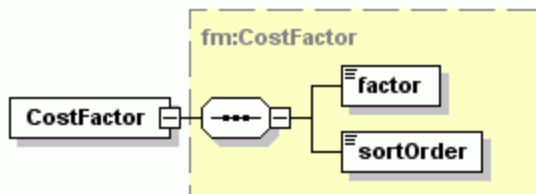
Property	Type	Description	Required	Default
		property are associated with this factor.		
id	String	Numeric value which is normally greater than 30,000. The value is associated with the selected cost factor.	No	N/A
name	String	Cost factor value that is associated with the selected cost factor.	No	N/A

WSCostFactorBean

WSCostFactorBean is used to select the cost factors and to specify the order by using the already seeded cost factors in PPM Center.

Note: For more information about the WSCostFactorBean, see:

- WSCostFactorBean is used as the INPUT in the following operation:
 - Operation "[getCostFactors](#)" on page 93
- WSCostFactorBean is used as the OUTPUT in the following operation:
 - Operation "[setCostFactors](#)" on page 95



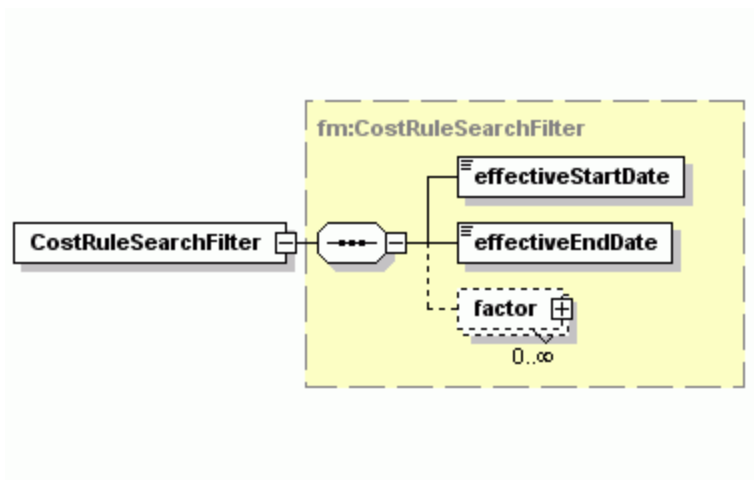
Property	Type	Description	Required	Default
costFactorName	String	One of the names of the cost factors that are already seeded in PPM Center. All the cost factors that you set through this method are applied for all the cost rules that are created in the system.	Yes	N/A
sortOrder	Integer	A numeric value used to identify the sort order. Lower numbers indicate more significance in the cost rule.	Yes	N/A

CostRuleSearchFilter

CostRuleSearchFilter bean is used to search for cost rules in the system.

Note: For more information about the CostRuleSearchFilter, see:

- CostRuleSearchFilter is used as the INPUT in the following operation:
 - Operation ["searchCostRules" on page 90](#)



Property	Type	Description	Required	Default
effectiveStartDate	Date	Specifies the effective start date for the cost rules that you search.	No	N/A

Property	Type	Description	Required	Default
effectiveEndDate	Date	Specifies the effective end date for the cost rules that you search	No	N/A
costFactorValues	List	List of WSCostFactorValueBeans. Specifies the cost factor values that are contained in the cost rule that you search.	No	N/A

Operations

Supported operations include reading, writing, creating, and deleting cost rules, as well as managing the global set of cost factors.

The system may perform many validations before you proceed with any operations or changes. For example, a rule cannot be created if it has the same cost factor values as another rule that already exists. However, the cost factor values themselves are not validated as this is the responsibility of the client.

The following operations are included in Financial Management Web services for cost rules:

- ["createCostRules" on the next page](#)
- ["updateCostRules" on page 83](#)
- ["getCostRules" on page 86](#)
- ["updateCostRules" on page 83](#)
- ["getCostRules" on page 86](#)
- ["deleteCostRules" on page 88](#)
- ["searchCostRules" on page 90](#)
- ["getCostFactors" on page 93](#)
- ["setCostFactors" on page 95](#)

createCostRules

Purpose

Creates a collection of new cost rules in the system.

Caution: Do not use the createCostRules operation to migrate a large volume of cost rule data from a legacy system into PPM Center as it will take too much time.

Creating a cost rule or any other operation is expected to take just slightly less time than the same operation run from within the PPM application.

Function

Creates a new set of cost rules in the system. The cost rule update service picks up these new cost rules and modifies the costs on any affected entities depending on the corresponding effective dates.

Limitations

The creation of cost rules using this operation occurs in a single transaction, where if one cost rule fails, the entire operation is rolled back.

Related Information

["updateCostRules" on page 83](#) updates a collection of cost rules.

Input

An array of ["WSCostRuleBean" on page 72](#).

Note: For date values entered, the format should be a parsable string by Java DateFormat class.

Otherwise a `ParseException` will be thrown by the `DateFormat` class, which will occur on the client side and not on the server side.

Return

An array of ["WSCostRuleBean" on page 72](#).

Java Interface

```
CreateCostRulesResponseDocument createCostRules
(CreateCostRulesDocument in)
```

Parameters	Description
CreateCostRulesDocument	<p>Wrapper for the <code>WSCostRuleBean</code> array.</p> <p>See the following example for the construction.</p> <p>The bean includes the following fields:</p> <ul style="list-style-type: none"> • Long id • int ruleNumber • List costRates// <code>List<CostRateBean></code> • List costFactorValues// <code>List<WSCostFactorValueBean></code>
CreateCostRulesResponseDocument	<p>Wrapper for the array of created cost rule ids. See the following example for retrieving the bean and its fields.</p>

Java Examples

Example: create a new set of cost rules.

```
public void testCreateCostRule() throws Exception {
    enableAllCostFactors();
    CostRule costRule = CostRule.Factory.newInstance();
    costRule.setFactorArray(new CostFactorValue[] {
        costFactorValue(DEPARTMENT, "Finance"),
        costFactorValue(REGION, "America"),
        costFactorValue(MISC_WORK_ITEMS, "Vacation"),
        costFactorValue(REQUEST_TYPE, "Bug")
    });
}
```

```
    }
    );
    costRule.setRateArray(new CostRate[] {
    costRate(null, "Oct 13, 1994", 1.0f),
    costRate("Oct 14, 1994", "Jan 1, 2000", 2.0f),
    costRate("Jan 1, 2001", "Jan 1, 2001", 3.0f),
    costRate("Jan 2, 2001", "June 28, 2006", 3.7f)
    }
    );

    createCostRules(new CostRule[] { costRule });
    }
    CostRule[] createCostRules(CostRule[] costRules) throws
Exception {
    FinanceServiceStub FM = new FinanceServiceStub(ctx, WSURL);
    CreateCostRulesDocument createCostRulesDoc =
CreateCostRulesDocument.Factory.newInstance();

    createCostRulesDoc.addNewCreateCostRules().setCostRuleArray(cos
tRules);
    CreateCostRulesResponseDocument response =
FM.createCostRules(createCostRulesDoc);
    return
response.getCreateCostRulesResponse().getCostRuleArray();
}
```

Errors and Exceptions

When an error occurs with this operation, you will see a description of the root cause in the log or in the response message.

The server log file content is similar to the following:

```
Exception in thread "main" org.apache.axis2.AxisFault:
<exception:exceptionDetails xmlns:exception="http://
www.mercury.com/ppm/ws/exception">
<exception:detail>[root cause description] </exception:detail>
</exception:exceptionDetails>
```

Root Cause Descriptions

Possible root cause descriptions:

Message Code	Message	Cause(s)	Possible Corrective Action
exception.authorization	You do not have the privilege to take this action. Please consult your PPM Administrator.	User not authorized to read the cost rule.	Check the user licenses and privileges (access grants).
error.ws.rateOverlap	One of the cost rules being created has two rates with overlapping effective ranges. Please check the rule and fix the rate overlap.	One of the rules has two or more rates with overlapping effective ranges.	Check the effective date ranges specified for the cost rate in the cost rule and make sure the dates are not overlapping.
error.conflictingCostFactors	You are adding a rule that has an illegal cost factor combination. No cost rule may have values for both the Resource and Role cost factor.	One of the rules has a combination of cost factors that is not allowed.	Avoid using the combination of cost factors mentioned in the message.
error.conflictingRules	You are adding a rule that has a conflict with an existing rule (Rule#30061). Please make sure the new rule has at least one cost factor with a different value.	One of the rules has the same set of cost factors as another rule in the system.	Change the cost factor values or use different cost factors.
validationError.nameNotFound	Could not find a Department whose name is MANUFACTUR. Please verify that a Department with this name exists in the system. Alternatively, you may specify the Department by ID instead of name.	One of the rules has a cost factor that is not in the system wide selected cost factor set.	Verify the cost factor values given for a cost rule. These values should be already seeded in the system.
ex.InvalidCurrency	Could not find a Currency with id USWD. Please verify that a Currency with this id exists in	Specified currency does not	Verify that a currency with this ID exists in

Message Code	Message	Cause(s)	Possible Corrective Action
	the system.	exist in the PPM system.	the system. Alternatively, you may specify the Currency by name instead of ID.
ex.InvalidDateRange	The effective end date {0} is before the start date {1}.	Specified start date is after the end date.	Specify the start date before the end date.
error.ws.notSelectedCostFactor	The cost factor {Project} is not selected for use. The selected cost factors are {Resource, Role, Region.}	Specified cost factor is not selected for use.	Use selected cost factors.
validationError.nameNotFound	Could not find a {0} whose name is {1} or the {0} with name {1} is disabled. Please verify that a {0} with this name exists in the system or enable it. Alternatively, you may specify the {0} by id instead of name.	Specified cost factor value you selected does not exist or disable.	Verify that a cost factor value with this name exists in the system.
validationError.nameAndIdNull	When specifying a {0} you must provide either its name or id. If you do not want to specify a {0} please omit it entirely.	The user selects a cost factor, but does not input the factor id or name.	Specify a value either for the name of the ID.

updateCostRules

Purpose

Updates the cost rates of the collection of existing cost rules in the system.

Function

Updates a collection of cost rules. Each rule to update is identified by its primary key and must exist in the database. The cost factor values of a rule cannot be changed after the rule is created. Only the collection of cost rates may be altered. You must record the updated rates as `CostRuleUpdates` so that the updates are visible to the Cost Rate Rule Update service, which triggers the service to update the costs of any affected entities.

Limitations

Updates can be made only to the cost rate for an existing cost rule. The cost factors cannot be updated. This is a functional limitation.

Related Information

["createCostRules" on page 78](#) creates a collection of cost rules.

Input

An object of an array of ["WSCostRuleBean" on page 72](#), which holds the cost rule IDs and the rates to update.

Return

None

Java Interface

UpdateCostRulesResponseDocument updateCostRules (UpdateCostRulesDocument in)

Parameters	Description
UpdateCostRulesDocument	Wrapper for WSCostRuleBean array. See the following example for the construction. The following fields in the bean are required: <ul style="list-style-type: none">• Long id• List costRates// List<CostRateBean>

Java Examples

Example: update a given set of cost rules.

```
public void testUpdateCostRule() throws Exception {  
    CostRule costRule = CostRule.Factory.newInstance();  
    costRule.setId (30010); //Set the cost rule id.  
    costRule.setRateArray(new CostRate[] {  
        costRate(null, "Oct 13, 1994", 1.0f),  
        costRate("Oct 14, 1994", "Jan 1, 2000", 2.0f),  
        costRate("Jan 1, 2001", "Jan 1, 2001", 3.0f),  
        costRate("Jan 2, 2001", "June 28, 2006", 3.7f)  
    }  
    );  
    updateCostRules(new CostRule[] { costRule });  
}  
void updateCostRules(CostRule[] costRules) throws Exception  
{  
    FinanceServiceStub FM = new FinanceServiceStub(ctx, WSURL);  
    UpdateCostRulesDocument updateCostRulesDoc =  
    UpdateCostRulesDocument.Factory.newInstance();  
    updateCostRulesDoc.addNewUpdateCostRules().setCostRuleArray(cos
```

```

tRules);
    FM.updateCostRules(updateCostRulesDoc);
}

```

Errors and Exceptions

When an error occurs on this operation, you will see a description of the root cause in the log or in the response message.

The server log file content is similar to the following:

```

Exception in thread "main" org.apache.axis2.AxisFault:
<exception:exceptionDetails
xmlns:exception="http://www.mercury.com/ppm/ws/exception">
<exception:detail>[root cause description] </exception:detail>
</exception:exceptionDetails>

```

Root Cause Descriptions

Possible root cause descriptions:

Message Code	Message	Cause(s)	Possible Corrective Action
exception.authorization	You do not have the privilege to take this action. Consult your PPM Administrator.	User is not authorized to read the cost rule.	Check the user licenses and privileges (access grants).
error.ws.rateOverlap	One of the cost rules being created has two rates with overlapping effective ranges. Check the rule and fix the rate overlap.	One of the rules has two or more rates with overlapping effective ranges.	Check the effective date ranges specified for the cost rate in the cost rule and make sure the dates are not overlapping.
error.ws.update.idRequired	A cost rule ID is required when updating cost rules. Please fill in the ID for each of the rules you wish to modify.	The rule ID is not specified for the updated cost rule.	Specify the rule ID.
ex.ObjectNotFound	Cannot load the {0} with	Cost rule is not	Check to read the cost

Message Code	Message	Cause(s)	Possible Corrective Action
	the specified ID: {1}. {0} may be deleted by another user.	found for any of the IDs in the costRuleBeans.	rule with the same ID to see if the cost rule exists.

getCostRules

Purpose

Gets the details of the cost rules for the given set of IDs.

Function

For a given set of IDs, this operation reads the details of the cost rules that are persisted in the system.

Limitations

Performance might be an issue if more than a few thousand cost rules read at one time.

Input

An object with an array of existing cost rule IDs.

Return

An object with an array of "[WSCostRuleBean](#)" on [page 72](#) corresponding to the given set of IDs.

Java Interface

`GetCostRulesResponseDocument getCostRules (GetCostRulesDocument in)`

Parameters	Description
<code>GetCostRulesDocument</code>	Wrapper for cost rule ids array. See the following example for the construction. The following field is required: <ul style="list-style-type: none">• <code>Long[] ids</code>
<code>GetCostRulesResponseDocument</code>	Wrapper for the array of cost rule beans (<code>WSCostRuleBean</code>) obtained performing this operation.

Java Examples

Example: Read cost rules for a given set of IDs.

```
CostRule[] getCostRules(long[] ids) throws Exception {
    FinanceServiceStub FM = new FinanceServiceStub(ctx, WSURL);
    GetCostRulesDocument getCostRulesDoc =
    GetCostRulesDocument.Factory.newInstance();

    getCostRulesDoc.addNewGetCostRules().setCostRuleIdArray(ids);
    GetCostRulesResponseDocument response =
    FM.getCostRules(getCostRulesDoc);
    return
    response.getGetCostRulesResponse().getCostRuleArray();
}
```

Errors and Exceptions

When an error occurs on this operation, you will see a description of the root cause in the log or in the response message.

The server log file content is similar to the following:

```
Exception in thread "main" org.apache.axis2.AxisFault:
<exception:exceptionDetails
xmlns:exception="http://www.mercury.com/ppm/ws/exception">
```

```
<exception:detail>[root cause description] </exception:detail>
</exception:exceptionDetails>
```

Root Cause Descriptions

Possible root cause descriptions:

Message Code	Message	Cause(s)	Possible Corrective Action
exception.authorization	You do not have the privilege to take this action. Consult your PPM Administrator.	User is not authorized to read the cost rule.	Check the user licenses and privileges (access grants).
ex.ObjectNotFound	Cannot load the {0} with the specified ID: {1}. {0} may be deleted by another user.	Cost rule is not found for any of the IDs in the array.	This means the cost rule with this ID does not exist in the PPM system. Try to read with another existing ID.

deleteCostRules

Purpose

Deletes the cost rules for a given set of cost rule IDs.

Function

Deletes the given cost rules. If a rule is not found for a given ID it is silently ignored. The default rule cannot be deleted, and attempts to delete it are ignored.

After the cost rule is deleted, the costs related with this cost rule will be recalculated for all entities in the system. This operation is not performed in the "frozen" state. This is done asynchronously by the Cost Rate Rule Update service.

Related Information

["getCostRules" on page 86](#) reads a collection of cost rules.

Input

An object with an array of cost rule IDs.

Return

None

Java Interface

`deleteCostRules (DeleteCostRulesDocument in)`

Parameters	Description
DeleteCostRulesDocument	Wrapper for cost rule ids array. See the following example for the construction. The following field is required: <ul style="list-style-type: none">• Long[] ids

Java Examples

Example: delete a set of cost rules.

```
void deleteCostRules(long[] ids) throws Exception {  
    FinanceServiceStub FM = new FinanceServiceStub(ctx, WSURL);  
    DeleteCostRulesDocument deleteCostRulesDoc =  
DeleteCostRulesDocument.Factory.newInstance();  
  
deleteCostRulesDoc.addNewDeleteCostRules().setCostRuleIdArray(i  
ds);
```

```
FM.deleteCostRules(deleteCostRulesDoc);
}
```

Errors and Exceptions

When an error occurs on this operation, you will see a description of the root cause in the log or in the response message.

The server log file content is similar to the following:

```
Exception in thread "main" org.apache.axis2.AxisFault:
<exception:exceptionDetails
xmlns:exception="http://www.mercury.com/ppm/ws/exception">
<exception:detail>[root cause description] </exception:detail>
</exception:exceptionDetails>
```

Root Cause Descriptions

Possible root cause descriptions:

Message Code	Message	Cause(s)	Possible Corrective Action
exception.authorization	You do not have the privilege to take this action. Consult your PPM Administrator.	User is not authorized to delete the cost rule	Check the user licenses and privileges (access grants).

Note: If the ID you provide to delete a cost rule is not found, the operation is silently ignored, and no exception is thrown.

searchCostRules

Purpose

Searches all the existing cost rules in the system that satisfy the search criteria provided.

Function

For a given search criteria, this operation gets all the cost rules satisfying the criteria. The criteria include the effective dates and the cost rate for the cost rules.

Limitations

The maximum number of results cannot exceed 1,000.

Related Information

["getCostRules" on page 86](#) gets all the cost rules for a given set of IDs.

Input

["CostRuleSearchFilter" on page 76](#)

Return

An object with an array of ["WSCostRuleBean" on page 72](#) corresponding to the given search filter.

Java Interface

```
SearchCostRulesResponseDocument searchCostRules (SearchCostRulesDocument in)
```

Parameters	Description
SearchCostRulesDocument	Wrapper for cost rule search filter (CostRuleSearchFilter).
SearchCostRulesResponseDocument	Wrapper for the array of cost rule beans (WSCostRuleBean) that the caller obtains by performing the search operation.

Java Examples

Example: Search cost rules for a given set of IDs.

```
CostRule[] searchAllCostRules() throws Exception {
    FinanceServiceStub FM = new FinanceServiceStub(ctx, WSURL);
    SearchCostRulesDocument searchCostRulesDoc =
SearchCostRulesDocument.Factory.newInstance();
    searchCostRulesDoc.addNewSearchCostRules().addNewFilter();
    // empty filter finds ALL rules
    SearchCostRulesResponse searchResponse =
FM.searchCostRules(searchCostRulesDoc).getSearchCostRulesRespon
se();
    return searchResponse.getCostRuleArray();
}
```

Errors and Exceptions

When an error occurs on this operation, you will see a description of the root cause in the log or in the response message.

The server log file content is similar to the following:

```
Exception in thread "main" org.apache.axis2.AxisFault:
<exception:exceptionDetails xmlns:exception="http://
www.mercury.com/ppm/ws/exception">
<exception:detail>[root cause description] </exception:detail>
</exception:exceptionDetails>
```

Root Cause Descriptions

Possible root cause descriptions:

Message Code	Message	Cause(s)	Possible Corrective Action
exception.authorization	You do not have the privilege to take this action. Consult your PPM Administrator.	User is not authorized to search cost rules.	Check the user licenses and privileges (access grants).

getCostFactors

Purpose

Gets the cost factors that are in PPM Center.

Function

Gets the cost factors defined in the system. Cost factors are returned in a list with the most significant factor first and the least significant factor last. For more information about the significance of cost factors, refer to the *Financial Management User's Guide*.

Limitations

None

Related Information

["setCostFactors" on page 95](#) sets the cost factors in PPM Center.

Input

None

Return

An object with an array of ["WSCostFactorBean" on page 75](#).

Java Interface

GetCostRulesResponseDocument getCostRules (GetCostRulesDocument in)

Parameters	Description
GetCostFactorsResponseDocument	Wrapper for the array of cost factor beans (WSCostFactorBean) that the caller obtains by performing this operation.

Java Examples

Example: Read the cost factors that are in PPM Center.

```
CostFactor[] getCostFactors() throws Exception {
    FinanceServiceStub FM = new FinanceServiceStub(ctx, WSURL);
    GetCostFactorsDocument getCostFactorsDoc =
    GetCostFactorsDocument.Factory.newInstance();
        getCostFactorsDoc.addNewGetCostFactors();
    GetCostFactorsResponseDocument response =
    FM.getCostFactors(getCostFactorsDoc);
    return
    response.getGetCostFactorsResponse().getCostFactorArray();
}
```

Errors and Exceptions

When an error occurs on this operation, you will see a description of the root cause in the log or in the response message.

The server log file content is similar to the following:

```
Exception in thread "main" org.apache.axis2.AxisFault:
<exception:exceptionDetails
xmlns:exception="http://www.mercury.com/ppm/ws/exception">
<exception:detail>[root cause description] </exception:detail>
</exception:exceptionDetails>
```

Root Cause Descriptions

Possible root cause descriptions:

Message Code	Message	Cause(s)	Possible Corrective Action
exception.authorization	You do not have the privilege to take this action. Consult your PPM Administrator.	User is not authorized to read the cost factors.	Check the user licenses and privileges (access grants).

setCostFactors

Purpose

Sets the cost factors in PPM Center to be used by cost rules.

Function

Sets the cost factors for the system. The precedence of cost factors is determined by their ordering in the list, the most significant cost factor is the first element in the list and the least significant factor is the last.

Caution: Setting of cost factors does not mean adding cost factors to an already existing list. This operation removes the existing list of cost factors and creates an entirely new list.

Changing the order of cost factors necessitates that costs be recalculated for all entities in the system that are not in a "frozen" state. This is done asynchronously by the Cost Rate Rule Update service. Costs do not need to be recalculated when cost factors are added or deleted. Adding new cost factors does not affect costing because no rule in the system has a value for the new factor. If a rule is created with a non-null value for the new factor, a CostRuleUpdate is created for it by PPM Center automatically. Deleting cost factors also does not require cost recalculation because only unused cost factors may be removed.

Related Information

["getCostFactors" on page 93](#) gets the cost factors in PPM Center.

Input

An object with an array of ["WSCostFactorBean" on page 75](#).

Return

None

Java Interface

SetCostFactorsResponseDocument
setCostFactors(SetCostFactorsDocument in)

Parameters	Description
SetCostFactorsDocument	Wrapper for the array of cost factor beans (WSCostFactorBean) that the caller obtains by performing this operation.

Java Examples

Example: Set the cost factors in PPM Center.

```
CostFactor[] setCostFactors() throws Exception {  
    SetCostFactorsDocument setCostFactorsDoc =  
    SetCostFactorsDocument.Factory.newInstance();  
    SetCostFactors setCostFactors =  
    setCostFactorsDoc.addNewSetCostFactors();  
    List newFactors = shuffleCostFactors();  
    populate(setCostFactors, newFactors);  
    if (VERBOSE) print(setCostFactors);  
}
```



```
FM.setCostFactors(setCostFactorsDoc);
}

List shuffleCostFactors() {
List factors = new ArrayList();
Random r = new Random();
for (int i = 0; i < COST_FACTORS.length; i++) {
if (r.nextFloat() > .2) { // more likeley to keep it then
not
factors.add(COST_FACTORS[i]);
}
}

Collections.shuffle(factors);
return factors;
}

void populate(SetCostFactors setCostFactors, List factors)
{
for (int i = 0; i < factors.size(); i++ ) {
CostFactor f = setCostFactors.addNewCostFactor();
f.setFactor((String) factors.get(i));
f.setSortOrder(BigInteger.valueOf(i));
}
}
}
```

Errors and Exceptions

When an error occurs on this operation, you will see a description of the root cause in the log or in the response message.

```
Exception in thread "main" org.apache.axis2.AxisFault:
<exception:exceptionDetails xmlns:exception="http://
www.mercury.com/ppm/ws/exception">
<exception:detail>[root cause description] </exception:detail>
</exception:exceptionDetails>
```

Root Cause Descriptions

Possible root cause descriptions:

Message Code	Message	Cause(s)	Possible Corrective Action
ex.Authorization	You do not have the privilege to take this action. Consult your PPM Administrator.	User is not authorized to read the cost factors.	Check the user licenses and privileges (access grants).
ex.CostFactorInUse	You are removing a cost factor - Request Type - that is still in use by cost rules in the system. Please remove these rules before proceeding.	An attempt is made to deactivate a cost factor that is being used by one or more cost rules. In other words, a CostFactorInUseException is thrown if one or more rules exist in the system with a non-null value for this factor.	<p>Do NOT deactivate or remove the cost factors that are being used by cost rules.</p> <p>Identify the cost rule associated with this cost factor. Then, determine if the cost rule should be removed. If the cost rule can be removed, do so, and then remove the cost factor.</p> <p>Note that when you are trying to set the cost factors, it is as though you are deleting the existing list and setting a whole new set of factors.</p>

Chapter 3: Financial Management (Finance Data) Web Services

- "Overview" below
- "References" below
- "Data Types" on the next page
- "Operations" on page 113

Overview

Financial Management Web services for finance data provides interfaces for accessing and updating finance data in PPM Center. These include operations for modifying cost rules, cost factors, financial summaries, and financial data. For more information about general Financial Management terms and concepts, see the *Financial Management User's Guide*.

Note: Financial summary Web services does not update the Financial summary forecast/actual lines if the lines are set to synchronize with a staffing profile.

References

Data types definition:

`webservice_toolkit\java\conf\xsd\Finance.xsd`

Operations definition:

`webservice_toolkit\java\conf\wsdl\FinanceService.wsdl`

Java sample code:

- `webservice_toolkit\java\client\src\examples\fm\FinanceServiceClient.java`
- `webservice_toolkit\java\client\src\examples\fm\FinancialSummaryClient.java`
- `webservice_toolkit\java\client\src\examples\fm\FinancialDataClient.java`

.NET sample code:

- `webservice_toolkit\MicrosoftDotNet\FinancialDataTest`
- `webservice_toolkit\MicrosoftDotNet\FinancialSummaryTest`

Data Types

Financial Management Web services for finance includes the following data types:

- ["FinancialSummaryInfo" below](#)
- ["FinancialDataInfo" on the next page](#)
- ["FinancialSummarySnapshotInfo" on page 102](#)
- ["SnapshotInfo" on page 103](#)
- ["UserDataInfo" on page 104](#)
- ["ApprovedBudgetInfo" on page 105](#)
- ["ForecastActualInfo" on page 106](#)
- ["BenefitInfo" on page 107](#)
- ["FinancialLineCellInfo" on page 108](#)
- ["CostPeriodSumCellInfo" on page 108](#)
- ["BenefitPeriodSumCellInfo" on page 111](#)
- ["ParentInfo" on page 111](#)
- ["FinancialDataParentInfo" on page 112](#)
- ["PeriodInfo" on page 112](#)
- ["AccessListInfo" on page 112](#)

FinancialSummaryInfo

This type defines the financial summary.

Property	Type	Description	Required	Default
id	Integer	Financial summary ID.	No	N/A

Property	Type	Description	Required	Default
name	String	Financial summary name.	No	N/A
description	String	Description of the financial summary.	No	N/A
parent	parentInfo	The parent entity associated with this financial summary. A parent entity can be one of the following: <ul style="list-style-type: none"> • Project • Proposal • Asset • Program • Org Unit 	No	N/A
localCurrencyCode	String	Local Currency Code.	No	N/A
baseCurrencyCode	String	Base Currency Code.	No	N/A
approvedBudgets	approvedBudgetInfo []	A list of approved budget entries.	No	N/A
forecastActual	forecastActualInfo	Planned and actual costs.	No	N/A
benefit	benefitInfo	Benefits.	No	N/A
snapshots	snapshotInfo[]	A list of snapshot headers.	No	N/A
discountRate	Double	Discount rate	No	N/A
Npv	Decimal	Net Present Value	No	N/A
Tnr	Decimal	Return on investment (ROI)	No	N/A

FinancialDataInfo

This type defines the financial data.

Property	Type	Description	Required	Default
id	Integer	Financial data ID.	No	N/A

Property	Type	Description	Required	Default
name	String	Financial data name.	No	N/A
description	String	Description of the financial data.	No	N/A
parent	financialDataParentInfo	The parent entity associated with this financial data. A parent entity can be one of the following: <ul style="list-style-type: none"> • Project • Proposal • Asset • Program • Org Unit 	No	N/A
localCurrencyCode	String	Currency Code.	No	N/A
baseCurrencyCode	String	Currency Code.	No	N/A
forecastActual	forecastActualInfo	Planned and actual costs.	No	N/A
benefit	benefitInfo	Benefits.	No	N/A

FinancialSummarySnapshotInfo

This type defines the financial summary snapshot.

Property	Type	Description	Required	Default
id	Integer	Financial summary snapshot ID.	No	N/A
name	String	Financial summary snapshot name.	No	N/A
description	String	Description of the financial summary snapshot.	No	N/A
parent	parentInfo	The parent entity associated with this financial summary. A parent entity can be one of the following: <ul style="list-style-type: none"> • Project • Proposal • Asset • Program • Org Unit 	No	N/A

Property	Type	Description	Required	Default
		<ul style="list-style-type: none"> • Project • Proposal • Asset • Program • Org Unit 		
localCurrencyCode	String	Currency Code.	No	N/A
baseCurrencyCode	String	Currency Code.	No	N/A
activeFSId	Integer	Active Financial Summary ID.	No	N/A
isPlanOfRecord	Boolean	Indicates if this snapshot is the plan of record	No	N/A
creationDate	Date	Date when the snapshot is taken.	No	N/A
approvedBudgets	approvedBudgetInfo []	List of approved budget entries.	No	N/A
forecastActual	forecastActualInfo	Planned and actual costs.	No	N/A
benefit	benefitInfo	Benefits.	No	N/A

SnapshotInfo

This type defines the financial summary snapshot.

Property	Type	Description	Required	Default
id	Integer	Financial summary snapshot ID.	No	N/A
name	String	Financial summary snapshot name	No	N/A
description	String	Description of the financial summary snapshot.	No	N/A
isPlanOfRecord	Boolean	Indicates if this snapshot is the plan of record.	No	N/A
creationDate	Date	Date when the snapshot is taken.	No	N/A

UserDataInfo

This type defines the user data type.

Property	Type	Description	Required	Default
userData1	String	User data string values.	No	N/A
userData2	String	User data string values.	No	N/A
userData3	String	User data string values.	No	N/A
userData4	String	User data string values.	No	N/A
userData5	String	User data string values.	No	N/A
userData6	String	User data string values.	No	N/A
userData7	String	User data string values.	No	N/A
userData8	String	User data string values.	No	N/A
userData9	String	User data string values.	No	N/A
userData10	String	User data string values.	No	N/A
userData11	String	User data string values.	No	N/A
userData12	String	User data string values.	No	N/A
userData13	String	User data string values.	No	N/A
userData14	String	User data string values.	No	N/A
userData15	String	User data string values.	No	N/A
userData16	String	User data string values.	No	N/A
userData17	String	User data string values.	No	N/A
userData18	String	User data string values.	No	N/A
userData19	String	User data string values.	No	N/A
userData20	String	User data string values.	No	N/A
visUserData1	String	Visible user data string values.	No	N/A
visUserData2	String	Visible user data string values.	No	N/A

Property	Type	Description	Required	Default
visUserData3	String	Visible user data string values.	No	N/A
visUserData4	String	Visible user data string values.	No	N/A
visUserData5	String	Visible user data string values.	No	N/A
visUserData6	String	Visible user data string values.	No	N/A
visUserData7	String	Visible user data string values.	No	N/A
visUserData8	String	Visible user data string values.	No	N/A
visUserData9	String	Visible user data string values.	No	N/A
visUserData10	String	Visible user data string values.	No	N/A
visUserData11	String	Visible user data string values.	No	N/A
visUserData12	String	Visible user data string values.	No	N/A
visUserData13	String	Visible user data string values.	No	N/A
visUserData14	String	Visible user data string values.	No	N/A
visUserData15	String	Visible user data string values.	No	N/A
visUserData16	String	Visible user data string values.	No	N/A
visUserData17	String	Visible user data string values.	No	N/A
visUserData18	String	Visible user data string values.	No	N/A
visUserData19	String	Visible user data string values.	No	N/A
visUserData20	String	Visible user data string values.	No	N/A

ApprovedBudgetInfo

This type defines the approved budget:

Property	Type	Description	Required	Default
id	Integer	Budget ID.	No	N/A
name	String	Budget name.	No	N/A
expenseType	String	Expense type of the budget.	No	Operating

Property	Type	Description	Required	Default
		Valid values are: <ul style="list-style-type: none"> • Operating • Capital 		
description	String	Budget description.	No	N/A
period	periodInfo	Budget date period.	No	N/A
amountLCL	Double	Budget amount in local currency.	No	N/A
amountBSE	Double	Budget amount in base currency.	No	N/A
approvalDate	Date	Approval date of the budget.	No	N/A
approvedBy	String	The user name of the user who approves the budget.	No	N/A

ForecastActualInfo

This type defines the forecast actual.

Property	Type	Description	Required	Default
id	Integer	Forecast actual ID.	No	N/A
actualRollupCode	String	<ul style="list-style-type: none"> • MANUAL • AUTO • PARTIAL 	No	N/A
isCapexOpexEnabled	Boolean	Indicates if planned and actual costs are capitalized.	No	N/A
lines	line[]	Actual lines. A line contains the following: <ul style="list-style-type: none"> • laborType: String • category: String • expenseType: String • syncSourceCode: <ul style="list-style-type: none"> • NO_SYNC (Default) 	No	N/A

Property	Type	Description	Required	Default
		<ul style="list-style-type: none"> • STAFFING_PROFILE • PROJECT • ASSET • PROPOSAL • cells: financialLineCellInfo[] • userData: userDataInfo <p>See Finance.xsd for more information.</p>		
periodSum	costPeriodSumCellInfo []	Sum of planned and actual costs by type (Labor, Non-Labor) and expenseType (Capitalized, Operating) over a given period	No	N/A
userData	userDataInfo	User data information.	No	N/A

BenefitInfo

This type defines the financial benefit.

Property	Type	Description	Required	Default
id	Integer	Financial benefit ID.	No	N/A
lines	line[]	<p>Actual lines. A line contains:</p> <ul style="list-style-type: none"> • ID: Integer • benefitType: String • category: String • cells: financialLineCellInfo[] • userData: userDataInfo <p>See Finance.xsd for more information</p>	No	N/A
periodSum	costPeriodSumCellInfo	Sum of benefit planned and actual	No	N/A

Property	Type	Description	Required	Default
	[]	values by period.		
userData	userDataInfo	User data information.	No	N/A

FinancialLineCellInfo

This type defines financial benefit or actual cells:

Property	Type	Description	Required	Default
period	periodInfo	Date period.	No	N/A
actualValueBSE	Double	Actual cost or benefit value in the base currency.	No	N/A
actualValueLCL	Double	Actual cost or benefit value in a local currency.	No	N/A
planValueBSE	Double	Planned cost or benefit value in the base currency.	No	N/A
planValueLCL	Double	Planned cost or benefit value in a local currency.	No	N/A

CostPeriodSumCellInfo

This type defines total planned and actual costs for a given period summed by type (labor, non-Labor) and expenseType (capitalized, operating) in the base currency and the local currency.

Property	Type	Description	Required	Default
period	periodInfo	Date period.	No	N/A
projectedTotal	Double	Total cost.	No	N/A
capexProjectedTotal	Double	CAPEX (Capitalized) total cost.	No	N/A
opexProjectedTotal	Double	OPEX (Operating) total cost.	No	N/A
planTotalLCL	Double	Planned total cost or benefit value in local currency.	No	N/A

Property	Type	Description	Required	Default
planTotalBSE	Double	Planned cost or benefit value in base currency.	No	N/A
actualTotalLCL	Double	Actual total cost or benefit value in local currency.	No	N/A
actualTotalBSE	Double	Actual total cost or benefit value in local currency.	No	N/A
planCapexTotalLCL	Double	Planned total CAPEX cost or benefit value in local currency.	No	N/A
planCapexTotalBSE	Double	Planned total CAPEX cost or benefit value in base currency.	No	N/A
planOpexTotalLCL	Double	Planned total OPEX in local currency.	No	N/A
planOpexTotalBSE	Double	Planned total OPEX in base currency.	No	N/A
planLaborTotalLCL	Double	Planned total labor cost value in local currency.	No	N/A
planLaborTotalBSE	Double	Planned total labor cost value in base currency.	No	N/A
planNonLaborTotalLCL	Double	Planned total non-labor cost value in local currency.	No	N/A
planNonLaborTotalBSE	Double	Planned total non-labor cost value in base currency.	No	N/A
actualCapexTotalLCL	Double	Actual CAPEX cost or benefit value in local currency.	No	N/A
actualCapexTotalBSE	Double	Actual CAPEX cost or benefit value in base currency.	No	N/A
actualOpexTotalLCL	Double	Actual OPEX cost or benefit value in local currency.	No	N/A
actualOpexTotalBSE	Double	Actual OPEX cost or benefit value in base currency.	No	N/A
actualLaborTotalLCL	Double	Actual total labor cost value in local currency.	No	N/A
actualLaborTotalBSE	Double	Actual total labor cost value in base currency.	No	N/A

Property	Type	Description	Required	Default
actualNonLaborTotalLCL	Double	Actual total non-labor cost value in local currency.	No	N/A
actualNonLaborTotalBSE	Double	Actual total non-labor cost value in base currency.	No	N/A
planCapexLaborLCL	Double	Planned CAPEX labor cost value in local currency.	No	N/A
planCapexLaborBSE	Double	Planned CAPEX labor cost value in base currency	No	N/A
planOpexLaborLCL	Double	Planned OPEX labor cost value in local currency.	No	N/A
planOpexLaborBSE	Double	Planned OPEX labor cost value in base currency	No	N/A
planCapexNonLaborLCL	Double	Planned CAPEX non-labor cost value in local currency.	No	N/A
planCapexNonLaborBSE	Double	Planned CAPEX non-labor cost value in base currency	No	N/A
planOpexNonLaborLCL	Double	Planned OPEX non-labor cost value in local currency.	No	N/A
planOpexNonLaborBSE	Double	Planned OPEX non-labor cost value in base currency	No	N/A
actualCapexLaborLCL	Double	Actual CAPEX non-labor cost value in local currency.	No	N/A
actualCapexLaborBSE	Double	Actual CAPEX non-labor cost value in base currency.	No	N/A
actualOpexLaborLCL	Double	Actual OPEX labor cost value in local currency.	No	N/A
actualOpexLaborBSE	Double	Actual OPEX labor cost value in base currency.	No	N/A
actualCapexNonLaborLCL	Double	Actual CAPEX non-labor cost value in local currency.	No	N/A
actualCapexNonLaborBSE	Double	Actual CAPEX non-labor cost value in base currency.	No	N/A
actualOpexNonLaborLCL	Double	Actual OPEX non-labor cost value in local currency.	No	N/A

Property	Type	Description	Required	Default
actualOpexNonLaborBSE	Double	Actual OPEX non-labor cost value in base currency.	No	N/A

BenefitPeriodSumCellInfo

This type defines the benefit period summary cell.

Property	Type	Description	Required	Default
period	periodInfo	Date period	No	N/A
actualValueBSE	Double	Actual cost or benefit value in the base currency	No	N/A
actualValueLCL	Double	Actual cost or benefit value in a local currency	No	N/A
planValueBSE	Double	Planned cost or benefit value in the base currency	No	N/A
planValueLCL	Double	Planned cost or benefit value in a local currency	No	N/A
projectedTotal	Double	Total benefit	No	N/A

ParentInfo

This type is used by the parent reference.

Property	Type	Description	Required	Default
parentType	String	Enum of: <ul style="list-style-type: none"> • PROPOSAL • PROJECT • ASSET • ORG_UNIT • PROGRAM 	No	N/A

Property	Type	Description	Required	Default
parentIdentifier	Long/String	ID or name that uniquely identifies the associated parent entity.	No	N/A

FinancialDataParentInfo

This type is used by the parent reference for financial data.

Property	Type	Description	Required	Default
parentType	String	Enum of: <ul style="list-style-type: none"> REQUEST 	No	N/A
parentId	Long	Parent ID.	No	N/A
token	String	The token that uniquely identifies the financial data field on a demand management request type.	No	N/A

PeriodInfo

This type defines the period.

Property	Type	Description	Required	Default
periodType	String	A string representing the fiscal period type. Currently, a value of "MONTH" is supported.	Yes	N/A
periodStartDate	Date	Period Start Date.	Yes	N/A

AccessListInfo

This type defines the access list.

Property	Type	Description	Required	Default
entry	(list)	<p>A list of the following entities:</p> <ul style="list-style-type: none"> • username: String or securityGroupName: String • viewCosts: Boolean • editCosts: Boolean • viewBenefits: Boolean • editBenefits: Boolean • viewApprovedBudget: Boolean • editApprovedBudget: Boolean • setPlanOfRecord: Boolean • editAccess: Boolean 	No	N/A

Operations

The following operations are included in Financial Management Web services for finance data:

- ["readFinancialSummary" on the next page](#)
- ["readFinancialSummarySnapshot" on page 115](#)
- ["updateFinancialSummary" on page 116](#)
- ["createFinancialSummarySnapshot" on page 117](#)
- ["readFinancialSummaryACL" on page 118](#)
- ["updateFinancialSummaryACL" on page 119](#)
- ["readFinancialData" on page 120](#)
- ["updateFinancialData" on page 121](#)
- ["createFinancialData" on page 122](#)
- ["readFinancialDataACL" on page 123](#)
- ["updateFinancialDataACL" on page 124](#)

readFinancialSummary

Purpose

This operation reads a financial summary.

Related Information

Data Type: ["ParentInfo" on page 111](#), ["FinancialSummaryInfo" on page 100](#)

Input

This operation requires the following input:

- financialSummaryParent: ParentInfo, with the following required information:
 - parentType
 - parentIdentifier

Return

["FinancialSummaryInfo" on page 100](#)

A financial summary will be returned if this operation is successful.

Java Examples

See `webservice_toolkit`:

```
java\client\src\examples\fm\FinancialSummaryClient.java  
readFinancialSummary ()
```

```
ReadFinancialSummaryDocument requestDoc =
ReadFinancialSummaryDocument.Factory.newInstance();
ReadFinancialSummary fs =
requestDoc.addNewReadFinancialSummary();

// Set up parent information.
ParentInfo parent =
populateParentInfo(fs.addNewFinancialSummaryParent(),
parentType, parentId);

// Read the financial summary.
FinanceServiceStub stub = new FinanceServiceStub(context,
serviceURL);
ReadFinancialSummaryResponseDocument responseDoc =
stub.readFinancialSummary(requestDoc);
ReadFinancialSummaryResponse response =
responseDoc.getReadFinancialSummaryResponse();

return response.getFinancialSummary();

ParentInfo populateParentInfo (ParentInfo parentInfo, String
parentType, Long parentId)
{
    ParentInfo.ParentType.Enum parentTypeEnum =
ParentInfo.ParentType.Enum.forString(parentType);
    parentInfo.setParentType(parentTypeEnum);

    ParentIdentifier identifier =
ParentIdentifier.Factory.newInstance();
    identifier.setId(new BigInteger(parentId.toString()));
    parentInfo.setParentIdentifier(identifier);
    return parentInfo;
}
```

readFinancialSummarySnapshot

Purpose

This operation reads a financial summary snapshot.

Related Information

Data type: ["ParentInfo" on page 111](#)

Input

This operation requires the following inputs:

- financialSummaryParent: ParentInfo, with the following required information:
 - parentType
 - parentIdentifier
- planOfRecord: Boolean
 - creationDate: Date

Return

["FinancialSummarySnapshotInfo" on page 102](#)

Java Examples

See webservice_toolkit:

```
java\client\src\examples\fm\FinancialSummaryClient.java  
readSnapshot ()
```

updateFinancialSummary

Purpose

This operation updates an existing financial summary.

Related Information

Operation: ["readFinancialSummary" on page 114](#)

Input

This operation requires the following input:

- financialSummary: ["FinancialSummaryInfo" on page 100](#)

Return

If the operation is successful, nothing will be returned to the client.

Java Examples

See `webservice_toolkit`:

```
java\client\src\examples\fm\FinancialSummaryClient.java
```

```
updateFinancialSummary ()
```

createFinancialSummarySnapshot

Purpose

This operation creates a financial summary snapshot.

Input

This operation requires the following inputs:

- financialSummaryParent: parentInfo (required)
- snapshot: snapshotInfo

Return

If the operation is successful, nothing will be returned to the client.

Java Examples

See `webservice_toolkit`:

```
java\client\src\examples\fm\FinancialSummaryClient.java  
createSnapshot ()
```

readFinancialSummaryACL

Purpose

This operation reads a financial summary access control list (ACL).

Related Information

Operation: ["readFinancialSummary" on page 114](#)

Input

This operation requires the following input:

- `financialSummaryParent`: ["ParentInfo" on page 111](#)

Return

["AccessListInfo" on page 112](#)

Java Examples

See `webservice_toolkit: java\client\src\examples\fm\FinancialSummaryClient.java`

```
readACL()
```

```
ReadFinancialSummaryACLDocument requestDoc =  
ReadFinancialSummaryACLDocument.Factory.newInstance();  
ReadFinancialSummaryACL root =  
requestDoc.addNewReadFinancialSummaryACL();  
  
ParentInfo parent =  
populateParentInfo(root.addNewFinancialSummaryParent(),  
parentType, parentId);  
  
FinanceServiceStub stub = new FinanceServiceStub(context,  
serviceURL);  
ReadFinancialSummaryACLResponseDocument responseDoc =  
stub.readFinancialSummaryACL(requestDoc);  
ReadFinancialSummaryACLResponse response =  
responseDoc.getReadFinancialSummaryACLResponse();  
  
return response.getAccessList();
```

updateFinancialSummaryACL

Purpose

This operation updates a financial summary ACL.

Input

This operation requires the following input:

- `financialSummaryParent`: ["ParentInfo" on page 111](#)
- `accessList`: ["AccessListInfo" on page 112](#)

Return

If the operation is successful, nothing will be returned to the client.

Java Examples

See `webservice_toolkit`:

```
java\client\src\examples\fm\FinancialSummaryClient.java  
updateACL()
```

readFinancialData

Purpose

This operation reads financial data.

Input

This operation requires the following input:

- `financialDataParent`: ["FinancialDataParentInfo" on page 112](#)

Return

Data as defined in ["FinancialDataInfo" on page 101](#).

Java Examples

See `webservice_toolkit`:


```
java\client\src\examples\fm\FinancialDataClient.java
```

```
readFinancialData ()
```

updateFinancialData

Purpose

This operation updates financial data.

Related Information

Operation: ["readFinancialData" on the previous page](#)

Input

This operation requires the following input:

- financialData: ["FinancialDataInfo" on page 101](#)

Return

If the operation is successful, nothing will be returned to the client.

Java Examples

See `webservice_toolkit`:

```
java\client\src\examples\fm\FinancialDataClient.java
```

```
updateFinancialData ()
```

```
// Modify the financial summary name and description.
fd.setName("Toolkit Financial Data");
fd.setDescription(makeUnique("Last Modified: "));

// Set the local currency.
fd.setLocalCurrencyCode("USD");

// Create or update benefits.
BenefitInfo benefit = null;
BenefitInfo.Lines benefitLines = fd.getBenefit().getLines();
if (isEmpty(benefitLines)) {
    benefit = createBenefit(fd.getBenefit());
}
else {
    benefit = updateBenefit(fd.getBenefit());
}
fd.setBenefit(benefit);

UpdateFinancialDataDocument requestDoc =
UpdateFinancialDataDocument.Factory.newInstance();
requestDoc.addNewUpdateFinancialData().setFinancialData(fd);

FinanceServiceStub stub = new FinanceServiceStub(context,
serviceURL);
stub.updateFinancialData(requestDoc);
```

createFinancialData

Purpose

This operation creates financial data.

Related Information

Operations: ["readFinancialData" on page 120](#), ["updateFinancialData" on the previous page](#)

Input

This operation requires the following inputs:

- financialData: ["FinancialDataInfo" on page 101](#)
- regionName: String

Return

If the operation is successful, nothing will be returned to the client.

Java Examples

See webservice_toolkit:

```
java\client\src\examples\fm\FinancialDataClient.java  
createFinancialData ()
```

readFinancialDataACL

Purpose

This operation reads a financial data ACL.

Related Information

Operation: ["readFinancialSummaryACL" on page 118](#)

Input

This operation requires the following input:

- financialDataParent: ["FinancialDataParentInfo" on page 112](#)

Return

Data as defined in ["AccessListInfo" on page 112](#).

Java Examples

See `webservice_toolkit`:

```
java\client\src\examples\fm\FinancialDataClient.java
```

```
readACL ()
```

updateFinancialDataACL

Purpose

This operation updates a financial data ACL.

Related Information

Operation: ["updateFinancialSummaryACL" on page 119](#)

Input

This operation requires the following inputs:

- `financialDataParent`: ["FinancialDataParentInfo" on page 112](#)
- `accessList`: ["AccessListInfo" on page 112](#)

Return

If the operation is successful, nothing will be returned to the client.

Java Examples

See `webservice_toolkit`:

```
java\client\src\examples\fm\FinancialDataClient.java
```

```
updateACL()
```

Chapter 4: Portfolio Management Web Services

- "Overview" below
- "References" below
- "Operations available in PPM Center" on the next page
- "Data Types" on the next page
- "Operations" on page 131

Overview

Portfolio Management Web Services provides operations to create, update, read, and delete hierarchical portfolios.

Note: For more information about general Portfolio Management terms and concepts, see the *Portfolio Management User's Guide*.

References

Data types definition:

`webservice_toolkit\java\conf\xsd\Portfolio.xsd`

Operations definition:

`webservice_toolkit\java\conf\wsdl\PortfolioService.wsdl`

Java sample code:

`webservice_toolkit\java\client\src\examples\pfm`

.NET sample code:

`webservice_toolkit\MicrosoftDotNet\PortfolioServiceTest`

Operations available in PPM Center

The following operations are available in PPM Center:

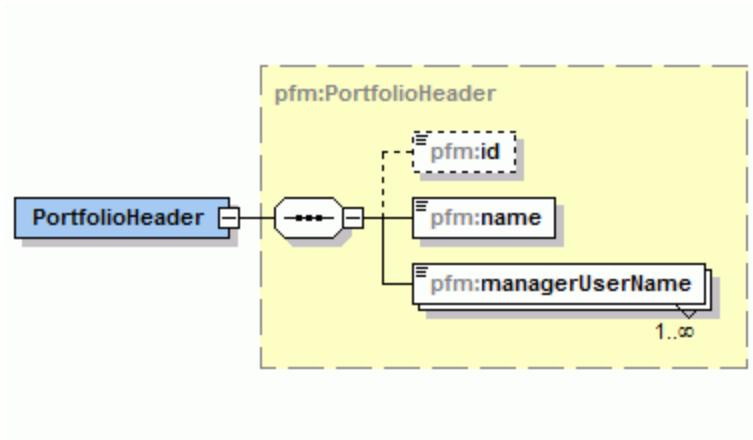
- ["createPortfolio" on page 132](#)
- ["updatePortfolio" on page 135](#)
- ["readPortfolio" on page 138](#)
- ["deletePortfolio" on page 141](#)
- ["addPortfolioChildren" on page 144](#)
- ["removePortfolioChildren" on page 147](#)
- ["addPortfolioPrograms" on page 149](#)
- ["removePortfolioPrograms" on page 152](#)
- ["addPortfolioContent" on page 155](#)
- ["removePortfolioContent" on page 158](#)

Data Types

The following data types are available in PPM Center:

- ["PortfolioWSHeaderBean" below](#)
- ["PortfolioContentItemBean" on page 129](#)
- ["PortfolioProgramBean" on page 129](#)
- ["PortfolioWSBean" on page 130](#)

PortfolioWSHeaderBean



PortfolioWSHeaderBean represents the header of a portfolio. A portfolio header includes the following elements:

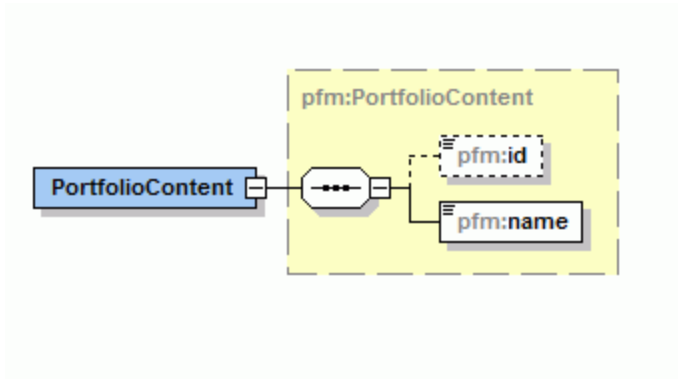
- Portfolio ID
- Portfolio name
- Portfolio managers

Note: PortfolioWSHeaderBean is used as the INPUT in the following operations:

- ["createPortfolio" on page 132](#)
- ["readPortfolio" on page 138](#)
- ["updatePortfolio" on page 135](#)
- ["deletePortfolio" on page 141](#)
- ["addPortfolioChildren" on page 144](#)
- ["removePortfolioChildren" on page 147](#)

Property	Type	Description	Required	Default
id	Long	Portfolio ID Numeric value which is normally greater than 30,000. Generated automatically on the creation of a portfolio object in PPM Center.	No	N/A
name	String	Name of the portfolio	Yes	N/A
portfolioManagers	Set	This property is a set of usernames of the PPM users who are the managers of this portfolio.	Yes	N/A

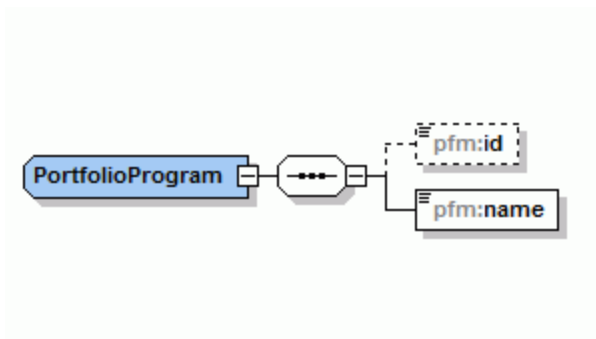
PortfolioContentItemBean



PortfolioContentItemBean represents each life cycle entity (proposals/projects/assets) that belongs to the portfolio. This data type includes the ID and the name for each entity.

Property	Type	Description	Required	Default
requestId	Long	ID of the request for the life cycle entity	Yes	N/A
name	String	Request name	Yes	N/A

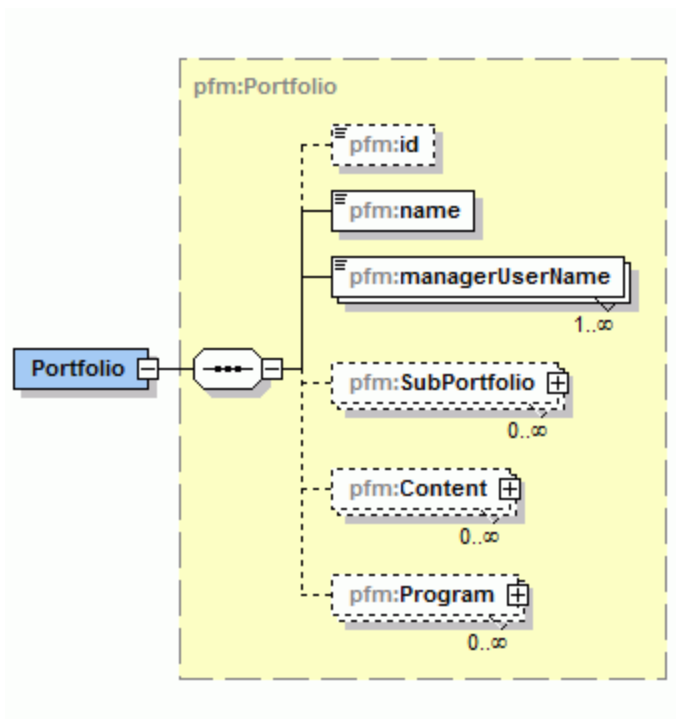
PortfolioProgramBean



PortfolioProgramBean represents a program that belongs to the portfolio. This data type includes the program ID and name.

Property	Type	Description	Required	Default
programId	Long	ID of the program that belongs to the portfolio	Yes	N/A
name	String	Program name	Yes	N/A

PortfolioWSBean



PortfolioWSBean represents a portfolio entity in its entirety. This data type consists of the header and contents of the portfolio, which include its child portfolios, programs, projects, proposals and assets.

Note: PortfolioWSBean is used as the OUTPUT in the following operations:

- ["createPortfolio" on page 132](#)
- ["readPortfolio" on page 138](#)
- ["updatePortfolio" on page 135](#)

Property	Type	Description	Required	Default
id	Long	Portfolio ID	No	N/A

Property	Type	Description	Required	Default
		Numeric value which is normally greater than 30,000. Generated automatically on the creation of a portfolio object in PPM Center.		
name	String	Name of the portfolio	Yes	N/A
portfolioManagers	Set	This property is a set of usernames of PPM users who are the managers of the portfolio.	Yes	N/A
SubPortfolios	Set	Set of "PortfolioWSHeaderBean" on page 127s , which are headers of the child portfolios	No	N/A
Contents	Set	Set of "PortfolioContentItemBean" on page 129s that represent the life cycle entities that belong to the portfolio	No	N/A
Programs	Set	Set of "PortfolioProgramBean" on page 129s that represent the programs that belong to the portfolio	No	N/A

Operations

The Portfolio Management Web Services implementation supports the basic Create/Read/Update/Delete operations for portfolios, including adding and removing child portfolios, programs and proposals/projects and assets to a portfolio.

Note: The Portfolio Management Web Services is not built for the purpose of migrating large volume of portfolio data from other systems or an earlier version of PPM Center into the current PPM Center version.

Note: For more information about the licenses and privileges (access grants) required for this operation, see the Security Model Guide and Reference and the Portfolio Management Configuration Guide.

The following operations are available in PPM Center:

- ["createPortfolio" on the next page](#)
- ["updatePortfolio" on page 135](#)
- ["readPortfolio" on page 138](#)
- ["deletePortfolio" on page 141](#)

- ["addPortfolioChildren" on page 144](#)
- ["removePortfolioChildren" on page 147](#)
- ["addPortfolioPrograms" on page 149](#)
- ["removePortfolioPrograms" on page 152](#)
- ["addPortfolioContent" on page 155](#)
- ["removePortfolioContent" on page 158](#)

createPortfolio

Purpose

This operation creates a portfolio.

Function

You can use this operation to create a new portfolio in PPM Center.

Related Information

The following related operations:

- ["readPortfolio" on page 138](#)
- ["updatePortfolio" on page 135](#)
- ["deletePortfolio" on page 141](#)

Input

This operation requires the following input:

- The portfolio name
- The usernames of the portfolio managers

Return

A ["PortfolioWSBean" on page 130](#) object with the ID, name and the set of managers populated.

Java Interface

```
CreatePortfolioResponseDocument  
createPortfolio(CreatePortfolioDocument param
```

Parameters	Description
CreatePotfolioDocument	Wrapper for PortfolioWSBean. See the Java example for construction. The bean includes the following fields: <ul style="list-style-type: none">• String name• String[] managers //array of the user names of the managers
CreatePortfolioResponseDocument	Wrapper for PortfolioWSBean. See the Java example for how to retrieve the bean and its fields.

Java Examples

```
public Portfolio testCreatePortfolio(String portfolioName,  
String[] managers) throws Exception {  
    System.out.println("testCreatePortfolio started ...");  
    PortfolioServiceStub stub = new PortfolioServiceStub(ctx,  
WSURL);  
    CreatePortfolioDocument createPortfolioDocument =  
CreatePortfolioDocument.Factory.newInstance();  
    final CreatePortfolioInput newPortfolio =  
createPortfolioDocument.addNewCreatePortfolio().addNewPortfolio  
();
```

```

        newPortfolio.setName(portfolioName);
        newPortfolio.setManagerUserNameArray(managers);
        CreatePortfolioResponseDocument response =
stub.createPortfolio(createPortfolioDocument);
        Portfolio createdPortfolio =
response.getCreatePortfolioResponse().getPortfolio();
        System.out.println("Created Portfolio " + portfolioName +
" with id " + createdPortfolio.getId());
        System.out.println("testCreatePortfolio completed");
        return createdPortfolio;
    }

```

Errors and Exceptions

When an error occurs on this operation, you will see a description of the root cause in the server log or in the response message:

The server log content is similar to the following:

```

Exception in thread "main" org.apache.axis2.AxisFault:
<exception:exceptionDetails xmlns:exception="http://
www.mercury.com/ppm/ws/exception">
<exception:detail>[root cause description] </exception:detail>
</exception:exceptionDetails>

```

Possible root cause descriptions:

Message Code	Message	Cause(s)	Possible Corrective Action
exception.authorization	You do not have the privilege to take this action. Please consult your PPM Administrator.	You are not authorized to create a portfolio.	Check the user licenses and privileges (access grants)
com.mercury.itg.pfm.service.PFMDDataValidationException	Portfolio Name already exists.	A portfolio with the specified name	Change the portfolio name.

Message Code	Message	Cause(s)	Possible Corrective Action
		already exists.	
com.mercury.itg.pfm.service.PFMDDataValidationException	One or more of the portfolio managers do not exist in PPM	One or more of the specified usernames do not exist in PPM Center.	Check the usernames.

updatePortfolio

Purpose

This operation updates the portfolio header details (name, set of managers) for a portfolio.

Function

You can use this operation to update the name or the managers of a portfolio.

Limitations

When the input data contains a list of portfolio managers, the existing list of managers is replaced with the list that you specify in the input of this operation. This means, the operation does not add new managers to the original list or delete existing managers from the original list. This has to be done by specifying a new list of managers for the portfolio.

Related Information

The following related operation:

- ["createPortfolio" on page 132](#)

Input

This operation requires the following input:

- The portfolio ID
- A new set of managers and/or the new name of the portfolio

Return

A ["PortfolioWSBean" on page 130](#) object with the ID, name and managers fields populated.

Java Interface

```
UpdatePortfolioResponseDocument updatePortfolio (UpdatePortfolioDocument in)
```

Parameters	Description
UpdatePortfolioDocument	Wrapper for PortfolioWSBean array. See the Java example for construction. The following fields in the bean are required: <ul style="list-style-type: none">• Long id• String name and /or String[] managers

Java Examples

```
public Portfolio testUpdatePortfolio(long portfolioId, String
portfolioName, String[] managers) throws Exception {
    System.out.println("testUpdatePortfolio started ...");
    PortfolioServiceStub stub = new PortfolioServiceStub(ctx,
WSURL);
    UpdatePortfolioDocument updatePortfolioDocument =
UpdatePortfolioDocument.Factory.newInstance();
    final UpdatePortfolioInput updatePortfolio =
updatePortfolioDocument.addNewUpdatePortfolio().addNewUpdatePor
tfolio();
    updatePortfolio.setId(portfolioId);
    updatePortfolio.setName(portfolioName);
    updatePortfolio.setManagerUserNameArray(managers);
    UpdatePortfolioResponseDocument response =
stub.updatePortfolio(updatePortfolioDocument);
    Portfolio updatedPortfolio =
response.getUpdatePortfolioResponse().getUpdatePortfolio();
    System.out.println("Updated Portfolio " + portfolioName +
" with id " + updatedPortfolio.getId());
    System.out.println("testUpdatePortfolio completed");
    return updatedPortfolio;
}
```

Errors and Exceptions

When an error occurs on this operation, you will see a description of the root cause in the server log or in the response message:

The server log content is similar to the following:

```
Exception in thread "main" org.apache.axis2.AxisFault:
<exception:exceptionDetails xmlns:exception="http://
www.mercury.com/ppm/ws/exception">
<exception:detail>[root cause description] </exception:detail>
</exception:exceptionDetails>
```

Possible root cause descriptions:

Message Code	Message	Cause(s)	Possible Corrective Action
exception.authorization	You do not have the privilege to take this action. Please consult your PPM Administrator.	You are not authorized to create a portfolio.	Check the user licenses and privileges (access grants)
com.mercury.itg.pfm.service.PFMDDataValidationException	Portfolio Name already exists.	A portfolio with the specified name already exists.	Change the portfolio name.
com.mercury.itg.pfm.service.PFMDDataValidationException	One or more of the portfolio managers do not exist in PPM	One or more of the specified usernames do not exist in PPM Center.	Check the usernames.

readPortfolio

Purpose

This operation reads the details of an existing portfolio specified by its name or ID.

Function

By specifying the name of ID of a portfolio, you can use this operation to retrieve all the details of the portfolio, including managers, child portfolios, programs and proposals/project and assets that belong to this portfolio.

Related Information

The following related operation:

- ["createPortfolio" on page 132](#)
- ["updatePortfolio" on page 135](#)
- ["deletePortfolio" on page 141](#)

Input

One of the following field is required:

- The portfolio ID
- The portfolio name

Return

A ["PortfolioWSBean" on page 130](#) object which includes all the details of the portfolio.

Java Interface

```
ReadPortfolioResponseDocument  
readPortfolio(ReadPortfolioDocument in)
```

Parameters	Description
ReadPortfolioDocument	Wrapper for the input of this operation. See the Java example for construction. One of the following fields in the bean is required: <ul style="list-style-type: none">• Long id• String name
ReadPortfolioResponseDocument	Wrapper for the PortfolioWSBean retrieved by performing this operation.

Java Examples

```
public Portfolio testReadPortfolio(String portfolioName) throws
Exception {
    System.out.println("testReadPortfolio started ...");
    PortfolioServiceStub stub = new
PortfolioServiceStub(ctx, WSURL);
    ReadPortfolioDocument readPortfolioDocument =
ReadPortfolioDocument.Factory.newInstance();
    ReadPortfolio readPortfolio =
readPortfolioDocument.addNewReadPortfolio();
    ReadPortfolioInput portfolioInput =
readPortfolio.addNewPortfolio();
    portfolioInput.setName(portfolioName);
    ReadPortfolioResponseDocument response =
stub.readPortfolio(readPortfolioDocument);
    Portfolio portfolio =
response.getReadPortfolioResponse().getPortfolio();
    System.out.println("Read Portfolio " +
portfolio.getName() + " with id " + portfolio.getId());
    System.out.println("testReadPortfolio completed");
    return portfolio;
}
```

Errors and Exceptions

When an error occurs on this operation, you will see a description of the root cause in the server log or in the response message:

The server log content is similar to the following:

```
Exception in thread "main" org.apache.axis2.AxisFault:  
<exception:exceptionDetails xmlns:exception="http://  
www.mercury.com/ppm/ws/exception">  
<exception:detail>[root cause description] </exception:detail>  
</exception:exceptionDetails>
```

Possible root cause descriptions:

Message Code	Message	Cause(s)	Possible Corrective Action
exception.authorization	You do not have the privilege to take this action. Please consult your PPM Administrator.	You are not authorized to create a portfolio.	Check the user licenses and privileges (access grants)
com.mercury.itg.exceptions.ObjectNotFoundException	Cannot load the {0} with the specified id: {1}. {0} may be deleted by another user.	A portfolio with the specified name or ID cannot be found.	Check the portfolio name or ID.

deletePortfolio

Purpose

This operation deletes an existing portfolio specified by the name or ID.

Function

By specifying the name or ID of a portfolio, you can use this operation to delete the portfolio. Portfolios that have child portfolios, programs and proposals/projects/assets cannot be deleted. You must use the Remove operations to remove the contents of the portfolio before deleting it.

Related Information

The following related operations:

- ["removePortfolioChildren" on page 147](#)
- ["removePortfolioPrograms" on page 152](#)
- ["removePortfolioContent" on page 158](#)

Input

One of the following field is required:

- The portfolio ID
- The portfolio name

Return

This operation does not return any data.

Java Interface

`deletePortfolio (DeletePortfolioDocument in)`

Parameters	Description
DeletePortfolioDocument	Wrapper for the input of this operation. See the Java example for construction. One of the following fields is required: <ul style="list-style-type: none">• Long id• String name

Java Examples

```
public void testDeletePortfolioById(Long portfolioId) throws
Exception {
    System.out.println("testDeletePortfolioById started
...");
    PortfolioServiceStub stub = new PortfolioServiceStub(ctx,
WSURL);
    DeletePortfolioDocument deletePortfolioDocument =
DeletePortfolioDocument.Factory.newInstance();
    DeletePortfolio deletePortfolio =
deletePortfolioDocument.addNewDeletePortfolio();
    ReadPortfolioInput portfolioInput =
deletePortfolio.addNewPortfolio();
    portfolioInput.setId(portfolioId);
    DeletePortfolioResponseDocument response =
stub.deletePortfolio(deletePortfolioDocument);
    System.out.println("testDeletePortfolioById completed");
}
public void testDeletePortfolioByName(String portfolioName)
throws Exception {
    System.out.println("testDeletePortfolioByName started
...");
    PortfolioServiceStub stub = new PortfolioServiceStub(ctx,
WSURL);
    DeletePortfolioDocument deletePortfolioDocument =
DeletePortfolioDocument.Factory.newInstance();
    DeletePortfolio deletePortfolio =
deletePortfolioDocument.addNewDeletePortfolio();
    ReadPortfolioInput portfolioInput =
deletePortfolio.addNewPortfolio();
    portfolioInput.setName(portfolioName);
    DeletePortfolioResponseDocument response =
stub.deletePortfolio(deletePortfolioDocument);
    System.out.println("testDeletePortfolioByName
completed");
}
```

Errors and Exceptions

When an error occurs on this operation, you will see a description of the root cause in the server log or in the response message:

The server log content is similar to the following:

```
Exception in thread "main" org.apache.axis2.AxisFault:  
<exception:exceptionDetails xmlns:exception="http://  
www.mercury.com/ppm/ws/exception">  
<exception:detail>[root cause description] </exception:detail>  
</exception:exceptionDetails>
```

Possible root cause descriptions:

Message Code	Message	Cause(s)	Possible Corrective Action
exception.authorization	You do not have the privilege to take this action. Please consult your PPM Administrator.	You are not authorized to create a portfolio.	Check the user licenses and privileges (access grants)
com.mercury.itg.exceptions.ObjectNotFoundException	Cannot load the {0} with the specified id: {1}. {0} may be deleted by another user.	A portfolio with the specified name or ID cannot be found.	Check the portfolio name or ID.

addPortfolioChildren

Purpose

This operation adds one or more child portfolios to a parent portfolio.

Function

You can use this operation to add a set of child portfolios to a parent portfolio.

Limitations

The maximum number of child portfolios that can be added to a parent is 40.

Related Information

The following related operation:

- ["removePortfolioChildren" on page 147](#)

Input

The following fields are required:

- The ID of the parent portfolio
- An array of IDs of the child portfolios that need to be added to the parent portfolio

Return

This operation does not return any data.

Java Interface

```
AddPortfolioChildrenResponseDocument addPortfolioChildren  
(AddPortfolioChildrenDocument in)
```

Parameters	Description
AddPortfolioChildrenDocument	Wrapper for the input of this operation. The input includes the following fields: <ul style="list-style-type: none">• Long id //parent portfolio ID• Long[] childrenIds //IDs of child portfolios

Java Examples

```
public void testAddPortfolioChildren(long parentPortfolioId,
long[] ids) throws Exception {
    System.out.println("testAddPortfolioChildren started
...");
    PortfolioServiceStub PFM = new PortfolioServiceStub(ctx,
WSURL);
    AddPortfolioChildrenDocument addChildrenDoc =
AddPortfolioChildrenDocument.Factory.newInstance();
    AddPortfolioChildren children =
addChildrenDoc.addNewAddPortfolioChildren();
    children.setPortfolioIdsArray(ids);
    children.setParentPortfolioId(parentPortfolioId);
    PFM.addPortfolioChildren(addChildrenDoc);
    System.out.println("testAddPortfolioChildren
completed");
}
```

Errors and Exceptions

When an error occurs on this operation, you will see a description of the root cause in the server log or in the response message:

The server log content is similar to the following:

```
Exception in thread "main" org.apache.axis2.AxisFault:
<exception:exceptionDetails xmlns:exception="http://
www.mercury.com/ppm/ws/exception">
<exception:detail>[root cause description] </exception:detail>
</exception:exceptionDetails>
```

Possible root cause descriptions:

Message Code	Message	Cause(s)	Possible Corrective Action
exception.authorization	You do not have the privilege to take this action. Please	You are not authorized to create a portfolio.	Check the user licenses and privileges

Message Code	Message	Cause(s)	Possible Corrective Action
	consult your PPM Administrator.		(access grants)
com.mercury.itg.exceptions.ObjectNotFoundException	Cannot load the {0} with the specified id: {1}. {0} may be deleted by another user.	A portfolio with the specified name or ID cannot be found.	Check the portfolio name or ID.

removePortfolioChildren

Purpose

This operation removes one or more child portfolios from a parent portfolio.

Function

You can use this operation to remove a set of child portfolios to a parent portfolio.

Related Information

The following related operation:

- ["addPortfolioChildren" on page 144](#)

Input

The following fields are required:

- The ID of the parent portfolio
- An array of IDs of the child portfolios that need to be removed from the parent portfolio

Return

This operation does not return any data.

Java Interface

```
RemovePortfolioChildrenResponseDocument removePortfolioChildren  
(RemovePortfolioChildrenDocument in)
```

Parameters	Description
RemovePortfolioChildrenDocument	Wrapper for the input of this operation. The input includes the following fields: <ul style="list-style-type: none">• Long id //parent portfolio ID• Long[] childrenIds //IDs of child portfolios

Java Examples

```
public void testRemovePortfolioChildren(long parentPortfolioId,  
long[] ids) throws Exception {  
    System.out.println("testRemovePortfolioChildren started  
...");  
    PortfolioServiceStub PFM = new PortfolioServiceStub(ctx,  
WSURL);  
    RemovePortfolioChildrenDocument delChildrenDoc =  
RemovePortfolioChildrenDocument.Factory.newInstance();  
    RemovePortfolioChildren children =  
delChildrenDoc.addNewRemovePortfolioChildren();  
    children.setPortfolioIdsArray(ids);  
    children.setParentPortfolioId(parentPortfolioId);  
    PFM.removePortfolioChildren(delChildrenDoc);  
    System.out.println("testRemovePortfolioChildren  
completed");  
}
```

Errors and Exceptions

When an error occurs on this operation, you will see a description of the root cause in the server log or in the response message:

The server log content is similar to the following:

```
Exception in thread "main" org.apache.axis2.AxisFault:  
<exception:exceptionDetails xmlns:exception="http://  
www.mercury.com/ppm/ws/exception">  
<exception:detail>[root cause description] </exception:detail>  
</exception:exceptionDetails>
```

Possible root cause descriptions:

Message Code	Message	Cause(s)	Possible Corrective Action
exception.authorization	You do not have the privilege to take this action. Please consult your PPM Administrator.	You are not authorized to create a portfolio.	Check the user licenses and privileges (access grants)
com.mercury.itg.exceptions.ObjectNotFoundException	Cannot load the {0} with the specified id: {1}. {0} may be deleted by another user.	A portfolio with the specified name or ID cannot be found.	Check the portfolio name or ID.

addPortfolioPrograms

Purpose

This operation adds one or more programs to a parent portfolio.

Function

You can use this operation to add a set of programs to a parent portfolio.

Limitations

The maximum number of programs that can be added to a parent is 250.

Related Information

The following related operation:

- ["removePortfolioPrograms" on page 152](#)

Input

The following fields are required:

- Long id - the ID of the parent portfolio
- Long[] ids - an array of IDs of programs that need to be added to the parent portfolio

Return

This operation does not return any data.

Java Interface

```
AddPortfolioProgramsResponseDocument  
addPortfolioPrograms(AddPortfolioProgramsDocument in)
```

Parameters	Description
AddPortfolioProgramsDocument	Wrapper for the input of this operation. The input includes the following fields: <ul style="list-style-type: none">• Long id //parent portfolio ID• Long[] ids //IDs of the programs to be added to the parent portfolio

Java Examples

```
public void testAddPortfolioProgram(long portfolioId, long[]
ids) throws Exception {
    System.out.println("testAddPortfolioProgram started
...");
    PortfolioServiceStub PFM = new PortfolioServiceStub(ctx,
WSURL);
    AddPortfolioProgramDocument addProgramDoc =
AddPortfolioProgramDocument.Factory.newInstance();
    AddPortfolioProgram program =
addProgramDoc.addNewAddPortfolioProgram();
    program.setPortfolioId(portfolioId);
    program.setProgramIdsArray(ids);
    PFM.addPortfolioProgram(addProgramDoc);
    System.out.println("testAddPortfolioProgram completed");
}
```

Errors and Exceptions

When an error occurs on this operation, you will see a description of the root cause in the server log or in the response message:

The server log content is similar to the following:

```
Exception in thread "main" org.apache.axis2.AxisFault:
<exception:exceptionDetails
xmlns:exception="http://www.mercury.com/ppm/ws/exception">
<exception:detail>[root cause description] </exception:detail>
</exception:exceptionDetails>
```

Possible root cause descriptions:

Message Code	Message	Cause(s)	Possible Corrective Action
exception.authorization	You do not have the privilege to take this action. Please consult your PPM Administrator.	You are not authorized to create a portfolio.	Check the user licenses and privileges (access grants)
com.mercury.itg.exceptions.ObjectNotFoundException	Cannot load the {0} with the specified id: {1}. {0} may be deleted by another user.	A portfolio/program with the specified name or ID cannot be found.	Check the portfolio/program name or ID.

removePortfolioPrograms

Purpose

This operation removes one or more programs to a parent portfolio.

Function

You can use this operation to remove a set of programs from a parent portfolio.

Related Information

The following related operation:

- ["addPortfolioPrograms" on page 149](#)

Input

The following fields are required:

- Long id - the ID of the parent portfolio
- Long[] ids - an array of IDs of programs that need to be removed from the parent portfolio

Return

This operation does not return any data.

Java Interface

```
RemovePortfolioProgramsResponseDocument  
removePortfolioPrograms(RemovePortfolioProgramsDocument in)
```

Parameters	Description
RemovePortfolioProgramsDocument	Wrapper for the input of this operation. The input includes the following fields: <ul style="list-style-type: none">• Long id //parent portfolio ID• Long[] ids //IDs of the programs to be removed from the parent portfolio

Java Examples

```
public void testRemovePortfolioProgram(long portfolioId, long[]  
ids) throws Exception {  
    System.out.println("testRemovePortfolioProgram started  
...");  
    PortfolioServiceStub PFM = new PortfolioServiceStub(ctx,  
WSURL);  
    RemovePortfolioProgramDocument removeProgramDoc =  
RemovePortfolioProgramDocument.Factory.newInstance();
```

```

    RemovePortfolioProgram program =
removeProgramDoc.addNewRemovePortfolioProgram();
    program.setPortfolioId(portfolioId);
    program.setProgramIdsArray(ids);
    PFM.removePortfolioProgram(removeProgramDoc);
    System.out.println("testRemovePortfolioProgram
completed");
  }

```

Errors and Exceptions

When an error occurs on this operation, you will see a description of the root cause in the server log or in the response message:

The server log content is similar to the following:

```

Exception in thread "main" org.apache.axis2.AxisFault:
<exception:exceptionDetails xmlns:exception="http://
www.mercury.com/ppm/ws/exception">
<exception:detail>[root cause description] </exception:detail>
</exception:exceptionDetails>

```

Possible root cause descriptions:

Message Code	Message	Cause(s)	Possible Corrective Action
exception.authorization	You do not have the privilege to take this action. Please consult your PPM Administrator.	You are not authorized to create a portfolio.	Check the user licenses and privileges (access grants)
com.mercury.itg.exceptions.ObjectNotFoundException	Cannot load the {0} with the specified id: {1}. {0} may be deleted by another user.	A portfolio/program with the specified name or ID cannot be found.	Check the portfolio/program name or ID.

addPortfolioContent

Purpose

This operation adds one or more proposals/projects and assets to a parent portfolio.

Function

You can use this operation to add a set of content (proposals, projects, or assets) to a parent portfolio.

Limitations

The maximum number of proposal/projects and assets that can be added to a parent is 1000.

Related Information

The following related operation

- ["removePortfolioContent" on page 158](#)

Input

The following fields are required:

- The ID of the parent portfolio
- An array of request IDs of proposals/projects and assets

Return

This operation does not return any data.

Java Interface

AddPortfolioContentResponseDocument
addPortfolioContent(AddPortfolioContentDocument in)

Parameters	Description
AddPortfolioContentDocument	Wrapper for the input of this operation. The input includes the following fields: <ul style="list-style-type: none">• Long portfolioId - parent portfolio ID• Long[] requestIds - request IDs of the content to be added

Java Examples

```
public void testAddPortfolioContent(long portfolioId, long[]
ids) throws Exception {
    System.out.println("testAddPortfolioContent started
...");
    PortfolioServiceStub PFM = new PortfolioServiceStub(ctx,
WSURL);
    AddPortfolioContentDocument addContentDoc =
AddPortfolioContentDocument.Factory.newInstance();
    AddPortfolioContent content =
addContentDoc.addNewAddPortfolioContent();
    content.setPortfolioId(portfolioId);
    content.setRequestIdsArray(ids);
    PFM.addPortfolioContent(addContentDoc);
    System.out.println("testAddPortfolioContent completed");
}
```

Errors and Exceptions

When an error occurs on this operation, you will see a description of the root cause in the server log or in the response message:

The server log content is similar to the following:

```
Exception in thread "main" org.apache.axis2.AxisFault:
<exception:exceptionDetails xmlns:exception="http://
```

```
www.mercury.com/ppm/ws/exception">
<exception:detail>[root cause description] </exception:detail>
</exception:exceptionDetails>
```

Possible root cause descriptions:

Message Code	Message	Cause(s)	Possible Corrective Action
exception.authorization	You do not have the privilege to take this action. Please consult your PPM Administrator.	You are not authorized to create a portfolio.	Check the user licenses and privileges (access grants)
com.mercury.itg.pfm.service.PFMDataValidationException	One or more of the requests you specified are not valid portfolio content.	One of the following causes: <ul style="list-style-type: none"> • One or more request IDs specified for content cannot be found in PPM Center • One or more of the contents are not proposals, projects, or assets. • One or 	Check the request IDs of the contents.

Message Code	Message	Cause(s)	Possible Corrective Action
		more of the contents already have a parent portfolio or program.	
com.mercury.itg.pfm.service.PFMDDataValidationException	One or more of the portfolio managers do not exist in PPM	One or more of the specified usernames do not exist in PPM Center.	Check the usernames.
com.mercury.itg.exceptions.ObjectNotFoundException	Cannot load the {0} with the specified id: {1}. {0} may be deleted by another user.	A portfolio with the specified name or ID cannot be found.	Check the parent portfolio name or ID.

removePortfolioContent

Purpose

This operation removes one or more proposals/projects and assets to a parent portfolio.

Function

You can use this operation to remove a set of content (proposals, projects, or assets) to a parent portfolio.

Related Information

The following related operation

- ["addPortfolioContent" on page 155](#)

Input

The following fields are required:

- The ID of the parent portfolio
- An array of request IDs of proposals/projects and assets

Return

This operation does not return any data.

Java Interface

RemovePortfolioContentResponseDocument

RemovePortfolioContent(RemovePortfolioContentDocument in)

Parameters	Description
RemovePortfolioContentDocument	Wrapper for the input of this operation. The input includes the following fields: <ul style="list-style-type: none">• Long portfolioId - parent portfolio ID• Long[] requestIds - request IDs of the content to be removed

Java Examples

```
public void testRemovePortfolioContent(long portfolioId, long[]
ids) throws Exception {
    System.out.println("testRemovePortfolioContent started
...");
    PortfolioServiceStub PFM = new PortfolioServiceStub(ctx,
WSURL);
    RemovePortfolioContentDocument removeContentDoc =
RemovePortfolioContentDocument.Factory.newInstance();
    RemovePortfolioContent content =
removeContentDoc.addNewRemovePortfolioContent();
    content.setPortfolioId(portfolioId);
    content.setReqIdsArray(ids);
    PFM.removePortfolioContent(removeContentDoc);
    System.out.println("testRemovePortfolioContent
completed");
}
```

Errors and Exceptions

When an error occurs on this operation, you will see a description of the root cause in the server log or in the response message:

The server log content is similar to the following:

```
Exception in thread "main" org.apache.axis2.AxisFault:
<exception:exceptionDetails xmlns:exception="http://
www.mercury.com/ppm/ws/exception">
<exception:detail>[root cause description] </exception:detail>
</exception:exceptionDetails>
```

Possible root cause descriptions:

Message Code	Message	Cause(s)	Possible Corrective Action
exception.authorization	You do not have the privilege to take this action. Please	You are not authorized to create a portfolio.	Check the user licenses and privileges

Message Code	Message	Cause(s)	Possible Corrective Action
	consult your PPM Administrator.		(access grants)
com.mercury.itg.exceptions.ObjectNotFoundException	Cannot load the {0} with the specified id: {1}. {0} may be deleted by another user.	A portfolio with the specified name or ID cannot be found.	Check the portfolio name or ID.

Chapter 5: Project Management Web Services

- ["Overview" below](#)
- ["References" on page 177](#)
- ["Data Types" below](#)
- ["Operations" on page 177](#)

Overview

Project Management Web services provides interfaces for accessing, creating, updating, and transitioning projects through workflow steps in PPM Center. Operations for creating work plans, adding tasks to work plans, accessing task information and modifying actual time for various task assignments, are also supported. Additionally, operations such as searching for tasks and projects are provided.

Note: For more information about general Project Management terms and concepts (such as, work plan, and project), see the *Project Management User's Guide*.

Data Types

Project Management Web services includes the following data types:

- ["ProjectType" on the next page](#)
- ["CreateProjectResultType" on page 165](#)
- ["ProjectInputType" on page 166](#)
- ["WorkPlanInputType" on page 166](#)
- ["AnchorType" on page 166](#)
- ["taskAnchors" on page 167](#)

- ["AddTaskResultType" on page 167](#)
- ["AddTaskResultType" on page 167](#)
- ["SearchProjectPreferenceType" on page 167](#)
- ["SearchTaskPreferenceType" on page 169](#)
- ["AssignmentType" on page 170](#)
- ["UpdateActualsInput" on page 170](#)
- ["ResourceType" on page 171](#)
- ["TaskType" on page 171](#)
- ["ActivityType" on page 174](#)
- ["ScheduleInfo" on page 174](#)
- ["TaskActualType" on page 175](#)
- ["MoneyInfo" on page 175](#)
- ["CostBean" on page 175](#)
- ["RoleInfo" on page 176](#)
- ["DependencyInfo" on page 176](#)

ProjectType

This common data type is used in many project-related operations.

Property	Type	Description	Required	Default
projectName	String	Name of the project.	No	N/A
projectTypeName ^a	String	Name of the project type for the project.	No	N/A
projectTypeId ^a	Long	Numeric value used to identify a project type.	No	N/A
projectManagerUserName	String[]	List of all the project managers for the project.	No	N/A

Property	Type	Description	Required	Default
plannedStartPeriodFullName	String	Full name of the project start period.	No	N/A
plannedFinishPeriodFullName	String	Full name of the project finish period.	No	N/A
regionName	String	Name of the project's region.	No	N/A
staffingProfileName	String	Name of the associated staffing profile.	No	N/A
budgetName	String	Name of the associated budget.	No	N/A
financialBenefitName	String	Name of the associated financial benefit.	No	N/A
simpleFields	SimpleFields	An array list of the following elements: <ul style="list-style-type: none"> • token • stringValue • dateValue 	No	N/A
tables	Table	An array list of token and columns elements.	No	N/A
pmReferences	pmReference	A list of references for the project. References contain the following information: <ul style="list-style-type: none"> • targetId (Integer) • targetTypeCode (Integer) • refRelationshipId (Integer) • referenceName (String) • relBehaviorCode (String): 	No	N/A

Property	Type	Description	Required	Default
		<ul style="list-style-type: none"> • SUCCESSOR • INFORMATIONAL • PREDECESSOR • FF_PREDECESSOR • AUTO_UPDATE • UPDATED_BY_STEP • refDescription (String) <p>The 'attachment' reference type is not yet supported through web services.</p>		
notes	Note	Refer to Demand Management.	No	N/A
fieldChangeNotes	fieldChangeNote	Refer to Demand Management.	No	N/A

a. When you make a web service call, you can either specify the projectTypeName property or the projectTypeId property, but not both of them. When the projectType object is returned in response to the web service call, only the projectTypeName is populated.

CreateProjectResultType

The CreateProjectResult data type returned when a new project is created.

Property	Type	Description	Required	Default
projectName ^a	String	Name of the project.	No	N/A
projectId	Long	A numeric value used to identify a project.	Yes	N/A
requestId	Integer	A numeric value used to identify a request corresponding to the newly created project.	Yes	N/A

Property	Type	Description	Required	Default
a. The projectName property is only present when you perform a bulkImportProject operation. This property is not set when you create a single project.				

ProjectInputType

This data type is used to identify a project.

Property	Type	Description	Required	Default
projectName ^a	String	Name of the project.	No	N/A
projectId ^a	Long	Numeric value used to identify a project.	No	N/A
requestId ^a	Integer	Numeric value used to identify a request corresponding to the newly-created project.	No	N/A

a. You can either specify the projectName property, the projectId property, or the requestId property associated with the project, but not all of them. None of these properties is "required", but at least one must be specified.

WorkPlanInputType

The WordPlanInputType data type is typically used for operations on a project's work plan.

Property	Type	Description	Required	Default
projectName ^a	String	Name of the project.	No	N/A
projectId ^a	Long	Numeric value used to identify a project.	No	N/A

a. You can either specify the projectName property or the projectId property, but not both of them. Neither of these properties is "required", but at least one must be specified.

Anchortype

This data type is used to order and outline tasks.

Property	Type	Description	Required	Default
taskSequenceNumber	Integer	Sequence number of the task. Task sequence starts from 0, which is the root task.	Yes	N/A
outLineLevel	Long	The outline level of the task. Task outLineLevel starts from 1, which is the root task.	Yes	N/A

taskAnchors

The TaskAnchors data type is used when you specify anchors for following operation:
addingTasksToExistingWorkPlan.

Property	Type	Description	Required	Default
topAnchor	anchorType	Top anchor task. New tasks should be added after the top anchor task.	Yes	N/A
bottomAnchor	anchorType	Bottom anchor task. New tasks should be added before the bottom anchor task.	No	N/A

AddTaskResultType

The AddTaskResultType data type is the returned result for tasks that are added to a work plan.

Property	Type	Description	Required	Default
taskSequenceNumber	Long	Sequence number of the new task in the work plan.	Yes	N/A
taskId	Long	Numeric value used to identify the task that is being added to the work plan.	Yes	N/A

SearchProjectPreferenceType

The SearchProjectPreference type is used for project search.

Property	Type	Description	Required	Default
startSearchPosition	Integer	Start position of the project in the search result. This is used in paged search. Starts at position 1.	Yes	0
maximumProjectsToShow	Integer	Number of projects to be returned for the search criteria. Values range from 1 to 1,000 inclusively.	Yes	N/A
projectNameContains	String	String contained in the project name of the projects that you want to search for.	No	N/A
projectTypeNames	String[]	List of project type names you want to search for.	No	N/A
projectRegionNames	String[]	List of project region names of the projects that you want to search for.	No	N/A
associatedProgramNames	String[]	List of program names associated with the projects that you want to search for.	No	N/A
projectManagerUserNames	String	List of project manager user names of the projects you want to search for.	No	N/A
isIncludeFinishProject	boolean	Determines whether the search result should include finished projects.	No	N/A
plannedStartFrom	dateTime	Start from date for the project you want to search for.	No	N/A
plannedStartTo	dateTime	Start to date for the project you want to search for.	No	N/A
plannedFinishFrom	dateTime	Finish from date for the project you want to search for.	No	N/A
plannedFinishTo	dateTime	Finish to date for the project you want to search for.	No	N/A
projectHealth	String[]	List of project health information you want to search for. Valid values for project health are: <ul style="list-style-type: none"> • RED • YELLOW 	No	N/A

Property	Type	Description	Required	Default
		<ul style="list-style-type: none"> GREEN NONE 		

SearchTaskPreferenceType

The SearchTaskPreferenceType data type is used for task search:

Property	Type	Description	Required	Default
startSearchPosition	Integer	Start position of the task in the search result. This is used in paged search. Starts at position 1.	Yes	N/A
maximumTasksToShow	Integer	Number of tasks to be returned for the search criteria. Values range from 1 to 1,000 inclusively.	Yes	N/A
taskNamePrefix	String	String the task name starts with.	No	N/A
projectNames	String[]	List of project names of the tasks that you want to search for.	No	N/A
resourceUserNames	String[]	List of resource user names of the tasks that you want to search for.	No	N/A
showOnlyMileStone	boolean	Determines whether the search result should show only milestone tasks.	No	N/A
showOnlyExceptionTask	boolean	Determines whether the search result should show only tasks with exceptions.	No	N/A
includeFinishedTask	boolean	Determines whether the search result should include finished tasks.	No	N/A
scheduledStartFrom	dateTime	Schedules the Start from date for the task you want to search for.	No	N/A
scheduledStartTo	dateTime	Schedules the Start to date for the task you want to search for.	No	N/A
scheduledFinishFrom	dateTime	Schedules the Finish from date for the task you want to search for.	No	N/A
scheduledFinishTo	dateTime	Schedules the Finish to date for the task you want to search for.	No	N/A

AssignmentType

The Assignment Type data type is the returned result for tasks that are added to a work plan. This type is part of the "UpdateActualsInput" below data type.

Property	Type	Description	Required	Default
taskId	Long	ID of the task, to which the assignment belongs.	No	N/A
scheduledEffort	Double	Scheduled effort for the assignment.	No	N/A
resource	ResourceType	Resources assigned to the task.	No	N/A
actualDuration	Double	Duration for the assignment.	No	N/A
percentComplete	Double	Completion percentage of the assignment, which is a value ranging from 0 to 100.	No	N/A
actualEffort	Double	Actual effort of the assignment.	No	N/A
estimatedRemainingEffort	Double	Estimated remaining effort of the assignment	No	N/A
actualStart	DateTime	Actual start date for the assignment.	No	N/A
actualFinish	DateTime	Actual finish date for the assignment.	No	N/A
estimatedFinish	DateTime	Estimated finish date for the assignment.	No	N/A

UpdateActualsInput

The UpdateActualsInput data type defines the assignment's (or work unit's) actuals information in relation to a task:

Property	Type	Description	Required	Default
assignment	AssignmentType	Assignment of the task, for which you want to update actuals.	Yes	N/A
taskId	Long	ID of the task, for which you want to update actuals.	Yes	N/A

ResourceType

The ResourceType data type is the returned result for tasks being added to a work plan.

Property	Type	Description	Required	Default
resourceId	Long	ID of the resource assigned to a task.	Yes	N/A
firstName	String	First name of the resource assigned to a task.	No	N/A
lastName	String	Last name of the resource assigned to a task.	No	N/A

TaskType

The TaskType data type defines tasks in a work plan.

Property	Type	Description	Required	Default
taskId	Long	ID of the task.	No	N/A
taskName	String	Name of the task.	Yes	N/A
taskSequence	Long	Sequence number of the task.	Yes	N/A
outlineLevel	Long	Outline level of a task within a work plan.	Yes	N/A
taskStatus	String	The status of a task with the	No	N/A

Property	Type	Description	Required	Default
		following valid values: <ul style="list-style-type: none"> • in-progress • completed • in-planning • ready • on-hold • active • cancelled • pending-predecessor • completed-pending-predecessor • completed-pending-request • pending-request 		
description	String	Description of the task.	No	N/A
priority	Long	Task priority.	No	N/A
isMilestone	boolean	Whether this task is a milestone.	No	N/A
isMajorMilestone	boolean	Whether this task is a major milestone.	No	N/A
isMilestoneAutomaticallyCompletes	boolean	Whether this Milestone automatically completes.	No	N/A
isMilestoneManualConversion	boolean	Whether this milestone is a	No	N/A

Property	Type	Description	Required	Default
		manual conversion.		
isRequired	boolean	Whether this task is required for the work plan.	No	N/A
activity	activityType	Activity for this task.	No	N/A
owners	resourceType	A list of owners of the task.	No	N/A
workUnits	assignmentType	A list of assignments for the task.	No	N/A
actuals	taskActualType	Task level actuals.	No	N/A
roleBean	roleInfo	Task role.	No	N/A
schedulingBean	scheduleInfo	Task schedule.	Yes	N/A
costBean	costInfo	Task cost.	No	N/A
predecessors	dependencyInfo	Task predecessors.	No	N/A
notificationSetupBean	notificationSetupInfo	Task notification setup.	No	N/A
userData	userDataInfo	Task user Data.	No	N/A
skillProficiencies	skillProficiencyInfo	Task skill proficiencies.	No	N/A
notes	taskNoteInfo	Task notes.	No	N/A
pmReferences	pmReference	Task references. Note that attachment is not a supported reference.	No	N/A

ActivityType

This type defines the activity for a task.

Property	Type	Description	Required	Default
activityId	Long	ID of the activity.	Yes	N/A
activityName	String	Name of the activity.	No	N/A

ScheduleInfo

The ScheduleInfo data type defines schedule information for a task.

Property	Type	Description	Required	Default
scheduledDuration	Double	Scheduled duration of a task.	No	N/A
scheduledEffort	Double	Scheduled effort of a task.	Yes	N/A
scheduledStart	DateTime	Scheduled start date of a task.	No	N/A
scheduledFinish	DateTime	Scheduled finish date of a task.	No	N/A
constraintType	String	Constraint Type of a task with the following valid values: <ul style="list-style-type: none">• as-soon-as-possible• as-late-as-possible• start-no-earlier-than• start-no-later-than• finish-no-earlier-than• finish-no-later-than• must-start-on• must-finish-on	No	N/A
constraintDate	DateTime	Constraint date of the task constraint.	No	N/A

TaskActualType

The TaskActualType data type defines the actual information for a task.

Property	Type	Description	Required	Default
actualDuration	Double	Actual duration of a task.	No	N/A
percentComplete	Double	Completion percentage of a task.	No	N/A
actualEffort	Double	Actual effort of a task.	No	N/A
estimatedRemainingEffort	Double	Estimated remaining effort of a task	No	N/A
actualStart	DateTime	Actual start date of task.	No	N/A
estimatedFinish	DateTime	Estimated finish date of task.	No	N/A
actualFinish	DateTime	Actual finish date of task.	No	N/A

MoneyInfo

The MoneyInfo data type defines the money information related to task cost.

Property	Type	Description	Required	Default
localValue	Double	Money value corresponding to the currency code.	Yes	N/A
localCurrencyCode	String	Currency code used for the value.	Yes	N/A

CostBean

The CostBean data type defines the task cost information.

Property	Type	Description	Required	Default
actualCapLaborMoney	MoneyInfo	Actual capitalized labor money of	No	N/A

Property	Type	Description	Required	Default
		the task.		
actualCapNonLaborMoney	MoneyInfo	Actual capitalized non-labor money for the task.	No	N/A
actualOpLaborMoney	MoneyInfo	Actual operating labor money of the task.	No	N/A
actualOpNonLaborMoney	MoneyInfo	Actual operating non-labor money for the task.	No	N/A
plannedCapLaborMoney	MoneyInfo	Planned capitalized labor money for the task.	No	N/A
plannedCapNonLaborMoney	MoneyInfo	Planned capitalized non-labor money for the task.	No	N/A
plannedOpNonLaborMoney	MoneyInfo	Planned operating non-labor money for the task.	No	N/A
plannedOpLaborMoney	MoneyInfo	Planned operating labor money for the task.	No	N/A

RoleInfo

The RoleInfo data type defines the role information for a task.

Property	Type	Description	Required	Default
roleId	Long	ID of the task role.	Yes	N/A
roleName	String	Name of the task role.	No	N/A

DependencyInfo

The DependencyInfo data type defines the role information for a task.

Property	Type	Description	Required	Default
taskName	String	Name of the predecessor task.	No	N/A

Property	Type	Description	Required	Default
taskId	Long	ID of the predecessor task.	No	N/A
isExternalPredecessor	boolean	Whether it is an external predecessor.	No	N/A
predTaskSeq	Long	Sequence number of the predecessor task.	Yes	N/A
predRelationType	String	Predecessor relationship with the following valid values: <ul style="list-style-type: none">• start-to-finish• finish-to-finish• start-to-start• finish-to-start	Yes	N/A
lagInDays	Double	Lag in days for the predecessor.	No	N/A

References

Data types definition:

webservice_toolkit\java\conf\xsd**Project.xsd**

Operations definition:

webservice_toolkit\java\conf\wsdl**ProjectService.wsdl**

Java sample code:

webservice_toolkit\java\client\src\examples\pm**ProjectServiceClient.java**

.NET sample code:

webservice_toolkit\MicrosoftDotNet\ProjectServiceTest

Operations

The following operations are included in Project Management Web services:

- ["createProject" on the next page](#)
- ["bulkImportProjects" on page 180](#)

- ["updateProject" on page 181](#)
- ["getProjectDetails" on page 182](#)
- ["executeWorkflowTransition" on page 184](#)
- ["createWorkPlanFromTemplate" on page 185](#)
- ["createBlankWorkPlan" on page 186](#)
- ["importWorkPlanTasks" on page 187](#)
- ["addTasksToExistingWorkPlan" on page 189](#)
- ["exportWorkPlanFromProject" on page 191](#)
- ["readTasks" on page 192](#)
- ["updateTaskActuals" on page 193](#)
- ["updateWorkPlanStatus" on page 194](#)
- ["searchProjects" on page 195](#)
- ["searchTasks" on page 196](#)

createProject

Purpose

Create a project and its associated request in PPM Center.

Function

Create a project with name, project type, and period.

Related Information

Data Type: ["ProjectType" on page 163](#)

Input

ProjectType with the following required information:

- Project name
- Project type name or ID
- A list of project manager user names. At least one user name should be provided.
- Planned start period full name
- Planned finish period full name
- Region name

All other information is optional. Specify the information only when you need to create a project with the corresponding information.

Return

["CreateProjectResultType" on page 165](#)

The project ID and request ID are returned if the project is created successfully.

Java Examples

See `webservice_toolkit`:

```
java\client\src\examples\pm\ProjectServiceClient.java
```

createProject()

```
CreateProjectDocument createProjDoc =  
CreateProjectDocument.Factory.newInstance();  
ProjectType projectBean =  
createProjDoc.addNewCreateProject().addNewProjectBean();  
  
projectBean.setProjectTypeName("Enterprise");  
projectBean.setRegionName("US West Coast");  
projectBean.addProjectManagerUserName("user1");  
projectBean.setPlannedStartPeriodFullName("January 2007");
```

```
projectBean.setPlannedFinishPeriodFullName("June 2007");  
projectBean.setProjectName(projectName);  
  
ProjectServiceStub stub = new ProjectServiceStub(ctx, WSURL);  
CreateProjectResponseDocument createProjectResponseDoc =  
stub.createProject(createProjDoc);  
  
CreateProjectResultType cpResult =  
createProjectResponseDoc.getCreateProjectResponse().getReturn()  
;  
  
return cpResult;
```

bulkImportProjects

Purpose

Create multiple projects at one time.

Function

This operation creates projects and their associated requests according to the given input. All projects are created in one transaction.

There is no limitation on the maximum number of projects that can be imported at the same time. However, if too many projects are created at the same time (for example, more than 1,000 projects), the operation can fail when there is a network issue.

Related Information

Operation: ["createProject" on page 178](#)

Return

["CreateProjectResultType" on page 165](#)

A list of project IDs, request IDs, and the corresponding project names are returned if all projects are created successfully. Otherwise, a SOAP fault with a detailed error message is returned.

Java Examples

```
BulkImportProjectsDocument importProjDoc =
BulkImportProjectsDocument.Factory.newInstance();
BulkImportProjects bip =
importProjDoc.addNewBulkImportProjects();
ProjectType projectBean = bip.addNewProjects();
... // Set projectBean for the first project

// Add the second input project bean
projectBean = bip.addNewProjects();
... // Set projectBean for the second project

BulkImportProjectsResponseDocument importProjectsResponseDoc =
stub.bulkImportProjects(importProjDoc);
// check the response
CreateProjectResultType[] cpResults =
importProjectsResponseDoc.getBulkImportProjectsResponse().getRe
turnArray();
```

updateProject

Purpose

Update an existing project.

Related Information

Operation: ["createProject" on page 178](#)

Input

The following inputs are required:

- One of the following:
 - Project name
 - Project ID
- ProjectType detail

Return

If the operation is successful, no results are returned to the client.

Java Examples

See `webservice_toolkit`:

```
java\client\src\examples\pm\ProjectServiceClient.java
```

updateProject()

```
UpdateProjectDocument updateProjDoc =  
UpdateProjectDocument.Factory.newInstance();  
UpdateProject up = updateProjDoc.addNewUpdateProject();  
ProjectInputType pit = up.addNewProjectInput();  
// Set the project name for update  
pit.setProjectName(projectName);  
  
ProjectType pb = up.addNewProjectBean();  
  
// Update the project info  
pb.addProjectManagerUserName("admin");  
pb.setPlannedStartPeriodFullName("March 2007");  
  
ProjectServiceStub stub = new ProjectServiceStub(ctx, WSURL);  
stub.updateProject(updateProjDoc);
```

getProjectDetails

Purpose

This operation collects detailed information for a project.

Function

Not all information related to a project and its request is returned. Only the information relevant to ProjectType (refer to the "[ProjectType](#)" on page 163 data type) is returned. If ProjectType details for a project do not exist, no results are returned.

Input

One of the following inputs is required:

- Project name
- Project ID

Return

The project together with its associated request detailed information is returned.

Java Examples

```
GetProjectDetailsDocument gpdDoc =
GetProjectDetailsDocument.Factory.newInstance();
GetProjectDetails gpd = gpdDoc.addNewGetProjectDetails();
ProjectInputType pit = gpd.addNewProjectInput();
pit.setProjectName(projectName);

GetProjectDetailsResponseDocument gpdResDoc =
stub.getProjectDetails(gpdDoc);
ProjectType pt =
gpdResDoc.getGetProjectDetailsResponse().getReturn();
```

executeWorkflowTransition

Purpose

This operation executes one workflow transition step from the current available steps.

Input

The following inputs are required:

- Transition step name

The step name must be one of the visible step names available on the **Project Details** tab in the PPM Center user interface.

- One of the following:
 - Project ID
 - Project name
 - Request ID

Return

If the operation is successful, no results are returned to the client.

Java Examples

See `webservice_toolkit`:

```
java\client\src\examples\pm\ProjectServiceClient.java
```

```
executeWorkflowTransition ()
```

```
String transition = "Launch";
```



```
ExecuteWorkflowTransitionDocument updateProjStatusDoc =  
    ExecuteWorkflowTransitionDocument.Factory.newInstance();  
  
ExecuteWorkflowTransition ups =  
updateProjStatusDoc.addNewExecuteWorkflowTransition();  
  
// create projectInputType  
ProjectInputType pit = ups.addNewProjectInput();  
pit.setProjectName(projectName);  
  
ups.setTransition(transition);  
  
stub.executeWorkflowTransition(updateProjStatusDoc)
```

createWorkPlanFromTemplate

Purpose

This operation creates a work plan for a specified project from a given template.

Input

The following inputs are required:

- Work plan template name.
- One of the following:
 - Project ID
 - Project name

Return

If the operation is successful, no results are returned to the client.

Java Examples

```
String templateName = "Template 1";  
String projectName = "Project template 1";  
  
CreateWorkPlanFromTemplateDocument crtblkWpDoc =  
CreateWorkPlanFromTemplateDocument.Factory.newInstance();  
CreateWorkPlanFromTemplate cbwp =  
crtblkWpDoc.addNewCreateWorkPlanFromTemplate();  
  
WorkPlanInputType wpit = cbwp.addNewProjectInput();  
// Set the value for WorkPlanInputType element's projectName  
wpit.setProjectName(projectName);  
  
// Set the value for CreateWorkPlanFromTemplate element's  
templateName  
cbwp.setTemplateName(templateName);  
// Invoke service method  
stub.createWorkPlanFromTemplate(crtblkWpDoc);
```

createBlankWorkPlan

Purpose

This operation creates a blank work plan for the specified project.

Input

One of the following:

- Project ID
- Project name

Return

If the operation is successful, no results are returned to the client.

Java Examples

See `webservice_toolkit`:

```
java\client\src\examples\pm\ProjectServiceClient.java
```

createBlankWorkPlan ()

```
String projectName = "Project 1";  
  
// Create input document  
CreateBlankWorkPlanDocument crtblkWpDoc =  
CreateBlankWorkPlanDocument.Factory.newInstance();  
  
// create and add an empty CreateBlankWorkPlan element  
CreateBlankWorkPlan cbwp =  
crtblkWpDoc.addNewCreateBlankWorkPlan();  
  
// Create and add an empty WorkPlanInputType element  
WorkPlanInputType wpit = cbwp.addNewProjectInput();  
  
// Set the value for this element  
wpit.setProjectName(projectName);  
  
stub.createBlankWorkPlan(crtblkWpDoc);
```

importWorkPlanTasks

Purpose

This operation creates and imports all the tasks to the work plan for a specified project.

Function

This operation requires that the specified project not have a work plan created. If a work plan already exists, a SOAP fault is returned to the client. In this operation, you do not need to specify actual values in `TaskActualType`. Instead, actual values should be specified in the assignment for each resource. The assignment actual values are then finally rolled up to the task.

Input

The following inputs are required:

- One of the following:
 - Project ID
 - Project name
- TaskType details with the following required information:
 - Task name
 - Task sequence number
 - Task outline level
 - Task schedule information

Return

["AddTaskResultType" on page 167](#)

If the operation is successful, a list of task sequence and task IDs corresponding to the tasks that are newly imported to the work plan is returned.

Java Examples

```
String projectName = "Project 1";

ImportWorkPlanTasksDocument impWPDoc =
ImportWorkPlanTasksDocument.Factory.newInstance();
ImportWorkPlanTasks impWP =
impWPDoc.addNewImportWorkPlanTasks();
WorkPlanInputType wpit = impWP.addNewWorkPlanInput();

// Set the value for WorkPlanInput element's projectName
wpit.setProjectName(projectName);

// Create and add an empty tasks element
TaskType task1 = impWP.addNewTasks();
```

```
// Set required properties for task
task1.setOutlineLevel(2);
task1.setTaskSequence(1);
task1.setTaskName("pm ws test importTask1" + uniqueSuffix);

// create and add task scheduling bean to task.
ScheduleInfo sif = task1.addNewSchedulingBean();
sif.setScheduledDuration(4);
sif.setScheduledEffort(34);
sif.setScheduledStart(taskScheduleStart);
sif.setScheduledFinish(taskScheduleFinish);
sif.setConstraintType(ConstraintType.AS_SOON_AS_POSSIBLE);

AssignmentType assignment = task1.addNewWorkUnits();
// Add resource to assignment
ResourceType resource = assignment.addNewResource();
resource.setResourceId(resourceId2.longValue());

assignment.setScheduledEffort(10);
assignment.setActualEffort(0);
assignment.setEstimatedRemainingEffort(10);
assignment.setPercentComplete(0);

// Create and add another empty tasks
TaskType task2 = impWP.addNewTasks();
... // Info for task2

// Calling service
ImportWorkPlanTasksResponseDocument impResponseDoc =
stub.importWorkPlanTasks(impWPDoc);

// Check the response and make sure we are getting it back ok
AddTaskResultType[] addedTasks =
impResponseDoc.getImportWorkPlanTasksResponse().getReturnArray(
);
```

addTasksToExistingWorkPlan

Purpose

This operation adds tasks to the work plan of the project.

Function

The position is defined by a top anchor and a bottom anchor. The top anchor is always required, but the bottom anchor can be null if you want to add tasks after the last existing task in the work plan.

This operation requires that the project have an existing work plan. In this operation, you do not need to specify actual values in `TaskActualType`. Instead, actual values should be specified in the assignment for each resource. The assignment actual values are then finally rolled up to the task.

Related Information

Operation: ["importWorkPlanTasks" on page 187](#)

Input

The following inputs are required:

- Project name or project ID
- TaskType details with the following required information:
 - Task name
 - Task sequence number
 - Task outline level
 - Task schedule information
- Position (topAnchor and bottomAnchor)

Return

["AddTaskResultType" on page 167](#)

If the operation is successful, a list of task sequence and task IDs corresponding to the tasks newly added to the work plan is returned.

Java Examples

See `webservice_toolkit`:

```
java\client\src\examples\pm\ProjectServiceClient.java
```

```
addTaskToExistingWorkPlanWithTopAnchor ()
```

exportWorkPlanFromProject

Purpose

This operation exports all the tasks in a work plan.

Function

Rather than all data, only those properties that are defined in the `TaskType` data type are exported.

Input

One of the following:

- Project name
- Project ID

Return

["TaskType" on page 171](#)

If the operation is successful, a list of tasks as defined in the `TaskType` is returned to the client.

Java Examples

```
ExportWorkPlanFromProjectDocument expWPDoc =  
ExportWorkPlanFromProjectDocument.Factory.newInstance();  
  
ExportWorkPlanFromProject expWP =  
expWPDoc.addNewExportWorkPlanFromProject();  
WorkPlanInputType wpit = expWP.addNewWorkPlanInput();  
  
wpit.setProjectName(projectName);  
  
ExportWorkPlanFromProjectResponseDocument expWPResDoc =  
stub.exportWorkPlanFromProject(expWPDoc);  
  
TaskType[] tasks =  
expWPResDoc.getExportWorkPlanFromProjectResponse().getReturnArr  
ay();
```

readTasks

Purpose

This operation reads a list of tasks information corresponding to the given task IDs.

Input

The input specifies a list of task IDs for which you want to collect task information. The specified tasks do not need to belong to the same project.

Return

["TaskType" on page 171](#)

If the operation is successful, a list of tasks as defined in the TaskType is returned to the client.

Java Examples

See `webservice_toolkit`:

```
java\client\src\examples\pm\ProjectServiceClient.java
```

readTasks ()

```
long taskId = 30000;  
  
ReadTasksDocument readTskDoc =  
ReadTasksDocument.Factory.newInstance();  
  
ReadTasks readTsks = readTskDoc.addNewReadTasks();  
readTsks.addTaskId(taskId);  
  
ReadTasksResponseDocument readTskresponseDoc =  
stub.readTasks(readTskDoc);  
  
TaskType[] tasks =  
readTskresponseDoc.getReadTasksResponse().getReturnArray();
```

updateTaskActuals

Purpose

This operation allows the actuals of task assignments to be updated.

Function

Some actual values cannot be changed if the resource is a Time Management user and has actuals logged in PPM Center already. In this case, refer to the user interface behavior in PPM Center.

Input

The following inputs are required:

- Task ID
- Resource ID of the resource to which the assignment belongs
- Other actual values

Return

If the operation is successful, no results are returned to the client.

Java Examples

See `webservice_toolkit`:

```
java\client\src\examples\pm\ProjectServiceClient.java
```

```
updateTaskActuals ()
```

updateWorkPlanStatus

Purpose

This operation updates the status of the root task of a given project's work plan.

Function

You cannot give arbitrary values for the status. Basically, only those statuses available in the root tasks of the PPM Center user interface are allowed. If the status is invalid, a SOAP fault with detailed error message is returned to the client.

Input

The following inputs are required:

- One of the following:
 - Project name
 - Project ID
- Task status (Note: this is the status you want the root task of the work plan to be.)

Return

If the operation is successful, no results are returned to the client.

Java Examples

```
UpdateWorkPlanStatusDocument uwpsDoc =  
UpdateWorkPlanStatusDocument.Factory.newInstance();  
UpdateWorkPlanStatus uwps=  
uwpsDoc.addNewUpdateWorkPlanStatus();  
WorkPlanInputType wpi = uwps.addNewWorkPlanInput();  
  
wpi.setProjectName(projectName);  
// new status to set  
uwps.setNewStatus("active");  
stub.updateWorkPlanStatus(uwpsDoc);
```

searchProjects

Purpose

This operation returns project IDs based on the search criteria.

Input

The input specifies the project search criteria. In addition to the search criteria, you must specify the `startSearchPosition` property, which should begin with 1 and the `maximumProjectsToShow` property, which should be within the range of 1 to 1,000.

Return

long[]

A list of project IDs corresponding to the search criteria.

Java Examples

```
SearchProjectsDocument schProjDoc =
SearchProjectsDocument.Factory.newInstance();

SearchProjects schProjs = schProjDoc.addNewSearchProjects();
SearchProjectPreferenceType searchPreference =
schProjs.addNewSearchPreferences();
// Set the search preferences
searchPreference.setStartSearchPosition(start);
searchPreference.setMaximumProjectsToShow(maxResult);
// searchPreference.setProjectNameContains("pm ws");

SearchProjectsResponseDocument schProjRespDoc =
stub.searchProjects(schProjDoc);

long[] projIds =
schProjRespDoc.getSearchProjectsResponse().getProjectIdsArray()
;
```

searchTasks

Purpose

This operation returns task IDs based on the search criteria.

Input

The input specifies the task search criteria. In addition to the search criteria, you must specify the `startSearchPosition` property, which should begin with 1, and the `maximumTasksToShow` property,

which should be within the range of 1 to 1,000.

Return

long[]

A list of task IDs corresponding to the search criteria.

Java Examples

See `webservice_toolkit`:

```
java\client\src\examples\pm\ProjectServiceClient.java
```

searchTasks ()

```
SearchTasksDocument schTskDoc =  
SearchTasksDocument.Factory.newInstance();  
SearchTasks schTsk = schTskDoc.addNewSearchTasks();  
  
// Create SearchPreference element  
SearchTaskPreferenceType searchPreference =  
schTsk.addNewSearchPreferences();  
  
// searchPreference  
searchPreference.setMaximumTasksToShow(10);  
searchPreference.setStartSearchPosition(1);  
  
searchPreference.setScheduledStartFrom(taskScheduleStart);  
// searchPreference  
searchPreference.setScheduledStartTo(taskScheduleFinish);  
  
SearchTasksResponseDocument schTskRespDoc =  
stub.searchTasks(schTskDoc);  
long[] taskids =  
schTskRespDoc.getSearchTasksResponse().getTaskIdArray();
```

Chapter 6: Program Management Web Services

- "Overview" below
- "References" below
- "Operations available in PPM Center" on the next page
- "Data Types" on the next page
- "Operations" on page 212

Overview

Web services for Program Management provides interfaces for accessing, creating, updating programs in PPM Center. An operation for searching programs is also provided.

Note: For more information about Program Management terms and concepts (such as Program, Portfolio, Proposal, Project, Asset), see the *Program Management User's Guide*.

References

Data types definition:

webservice_toolkit\java\conf\xsd\Program.xsd

Operations definition:

webservice_toolkit\java\conf\wsdl\ProgramService.wsdl

Java sample code:

webservice_toolkit\java\client\src\examples\pgm\ProgramServiceClient.java

.Net sample code:

webservice_toolkit\MicrosoftDotNet\ProgramServiceTest

Operations available in PPM Center

The following operations are available in PPM Center:

- ["createProgram" on page 212](#)
- ["readProgram" on page 214](#)
- ["updateProgram" on page 216](#)
- ["deleteProgram" on page 218](#)
- ["searchProgram" on page 219](#)

Data Types

Program Management Web services includes the following data types:

- ["ProgramInfo" on the next page](#)
- ["StatusCodeInfo" on page 201](#)
- ["HealthInfo" on page 202](#)
- ["ContentItemIdentifierInfo" on page 202](#)
- ["ContentItemInfo" on page 203](#)
- ["ContentInfo" on page 203](#)
- ["CostInfo" on page 204](#)
- ["ProgramCostInfo" on page 205](#)
- ["IssueInfo" on page 205](#)
- ["RiskInfo" on page 206](#)
- ["ScopeChangeInfo" on page 207](#)
- ["BusinessObjectiveIdentifierInfo" on page 207](#)
- ["BusinessObjectiveInfo" on page 207](#)
- ["WeightInfo" on page 208](#)

- ["SettingsInfo" on page 208](#)
- ["ProgramSearchFiltersInfo" on page 210](#)
- ["Note" on page 211](#)

ProgramInfo

ProgramInfo is a common type used in many program-related operations. This data type contains the following properties:

Property	Type	Description	Required	Default
id	Integer	Program ID	No	N/A
name	String	Program name	Yes	N/A
programTypeName ^a	String	Program type name	Yes	N/A
programTypeId ^a	Long	Program type ID	Yes	N/A
managers	common:User[]	Program managers (users)	No	N/A
priority	Integer	Priority of the program	No	N/A
score	Integer	Score of the program	No	N/A
statusCode	"StatusCodeInfo" on the next page	Status of the program	No	N/A
health	"HealthInfo" on page 202	Program health	No	N/A
issueHealth	"HealthInfo" on page 202	Program issue health	No	N/A
riskHealth	"HealthInfo" on page 202	Program risk health	No	N/A
scopeChangeHealth	"HealthInfo" on page 202	Program scope change health	No	N/A

Property	Type	Description	Required	Default
description	String	Description of the program	No	N/A
benefit	String	Program benefit of the program	No	N/A
statusNotes	String	Status notes of the program	No	N/A
costs	"ProgramCostInfo" on page 205	Program costs of the program	No	N/A
content	"ContentInfo" on page 203	Program content (Proposal, Project, or Asset)	No	N/A
issues	"IssueInfo" on page 205 []	Issues of the program	No	N/A
risks	"RiskInfo" on page 206 []	Risks of the program	No	N/A
scopeChanges	"ScopeChangeInfo" on page 207 []	Scope changes	No	N/A
businessObjectives	"BusinessObjectiveInfo" on page 207 []	Business objectives	No	N/A
userData	common:UserData	User data	No	N/A
notes	"Note" on page 211	Note of the program	No	N/A
settings	"SettingsInfo" on page 208	Program settings	No	N/A

a. You can either specify the programTypeName property or the programTypeid property, but not both of them. When the programType object is returned in response to the web service call, only the programTypeName is populated.

StatusCodeInfo

StatusCodeInfo is an enumeration data type for program status values. This data type contains the following properties:

Property	Type	Description	Required	Default
<enumeration>	String	Values: <ul style="list-style-type: none"> • NEW • ACTIVE • ON_HOLD • COMPLETED • CANCELLED 	No	N/A

HealthInfo

HealthInfo is an enumeration data type for program health values.

Property	Type	Description	Required	Default
<enumeration>	String	Values: <ul style="list-style-type: none"> • RED • YELLOW • GREEN • NONE 	No	N/A

ContentIdentifierInfo

ContentIdentifierInfo defines program contents (Proposal/Project/Asset). This data type contains the following properties:

Property	Type	Description	Required	Default
requestId	Integer	Request id of the content	No	N/A
name	String	Name of the content	No	N/A
type	String	Enumeration of the following:	No	N/A

Property	Type	Description	Required	Default
		<ul style="list-style-type: none"> • PROPOSAL • PROJECT • ASSET 		

ContentItemInfo

ContentItemInfo type extends "[ContentItemIdentifierInfo](#)" on the previous page. This data type contains the following properties:

Property	Type	Description	Required	Default
managers	common:User[]	Managers of the content	No	N/A
status	String	Status of the content	No	N/A
scheduledStartDate	dateTime	Schedule start date	No	N/A
scheduledFinishDate	dateTime	Schedule finish date	No	N/A
percentComplete	Integer	Percent completed	No	N/A
health	"HealthInfo" on the previous page	Health of the content	No	N/A
costHealth	"HealthInfo" on the previous page	Health of the content cost	No	N/A
costs	"CostInfo" on the next page	Costs	No	N/A

ContentInfo

ContentInfo defines program contents. This data type contains the following property:

Property	Type	Description	Required	Default
contentItem	"ContentItemInfo" above	List of content items	No	N/A

CostInfo

CostInfo defines cost information. This data type contains the following properties:

Property	Type	Description	Required	Default
currencyCode	String	Currency code	No	N/A
planTotal	Double	The total planned value of a financial summary.	No	N/A
actualTotal	Double	The total actual value of a financial summary.	No	N/A
planLaborTotal	Double	The total planned labor cost of a financial summary.	No	N/A
planNonLaborTotal	Double	The total planned non-labor cost of a financial summary	No	N/A
actualLaborTotal	Double	The total actual labor cost of a financial summary.	No	N/A
actualNonLaborTotal	Double	The total actual non-labor cost of a financial summary.	No	N/A
planCapexTotal	Double	The total planned CAPEX labor cost of a financial summary.	No	N/A
planOpexTotal	Double	The total planned OPEX labor cost of a financial summary.	No	N/A
actualCapexTotal	Double	The total actual CAPEX value of a financial summary.	No	N/A
actualOpexTotal	Double	The total actual OPEX value of a financial summary.	No	N/A
planCapexLabor	Double	The total planned CAPEX value of a financial summary.	No	N/A
planOpexLabor	Double	The total planned OPEX value of a financial summary.	No	N/A
planCapexNonLabor	Double	The total planned CAPEX non-labor cost of a financial summary.	No	N/A

Property	Type	Description	Required	Default
planOpexNonLabor	Double	The total planned OPEX non-labor cost of a financial summary.	No	N/A
actualCapexLabor	Double	The total actual CAPEX labor cost of a financial summary.	No	N/A
actualOpexLabor	Double	The total actual OPEX labor cost of a financial summary.	No	N/A
actualCapexNonLabor	Double	The total actual CAPEX non-labor cost of a financial summary.	No	N/A
planOpexNonLabor	Double	The total planned OPEX non-labor cost of a financial summary.	No	N/A
actualCapexLabor	Double	The total actual CAPEX labor cost of a financial summary.	No	N/A
actualOpexLabor	Double	The total actual OPEX labor cost of a financial summary.	No	N/A
actualCapexNonLabor	Double	The total actual CAPEX non-labor cost of a financial summary.	No	N/A
actualOpexNonLabor	Double	The total actual OPEX non-labor cost of a financial summary.	No	N/A

ProgramCostInfo

ProgramCostInfo extends "[CostInfo](#)" on the previous page. This data type contains the following properties:

Property	Type	Description	Required	Default
lastSyncedOn	dateTime	Last synchronization date	No	N/A
needsSync	Boolean	If it is true, the cost information needs synchronizing. Otherwise, the cost information is up to date.	No	N/A

IssueInfo

IssueInfo defines related issues. This data type contains the following properties:

Property	Type	Description	Required	Default
id	Integer	Issue ID	Yes	N/A
statusCode	String	Issue status	Yes	N/A
description	String	Issue description	No	N/A
projectId	Integer	ID of the related project Populated if the issue is logged on the project	No	N/A
projectName	String	Name of the related project Populated if the issue is logged on the project	No	N/A
priorityCode	String	Issue priority	No	N/A
escalationLevelCode	String	Escalation level	No	N/A

RiskInfo

RiskInfo defines related risks. This data type contains the following properties:

Property	Type	Description	Required	Default
id	Integer	Risk ID	Yes	N/A
statusCode	String	Risk status	Yes	N/A
description	String	Risk description	No	N/A
projectId	Integer	ID of the related project Populated if the risk is logged on the project	No	N/A
projectName	String	Name of the related project Populated if the risk is logged on the project	No	N/A
priorityCode	String	Risk priority	No	N/A
impactLevelCode	String	Impact level	No	N/A
probabilityCode	String	Probability	No	N/A

ScopeChangeInfo

ScopeChangeInfo represents scope changes. This data type contains the following properties:

Property	Type	Description	Required	Default
id	Integer	Scope change request ID	Yes	N/A
statusCode	String	Scope change Status	Yes	N/A
description	String	Scope change description	No	N/A
projectId	Integer	ID of the related project	No	N/A
projectName	String	Name of the related project	No	N/A
priorityCode	String	Scope change priority	No	N/A
crLevelCode	String	Scope change level	No	N/A
severityCode	String	Scope change severity	No	N/A

BusinessObjectiveIdentifierInfo

BusinessObjectiveIdentifierInfo identifies business objectives. This data type contains the following properties:

Property	Type	Description	Required	Default
id	Integer	Business objective ID	No	N/A
name	String	Business objective name	No	N/A

BusinessObjectiveInfo

BusinessObjectiveInfo extends type "[BusinessObjectiveIdentifierInfo](#)" above. This data type contains the following properties:

Property	Type	Description	Required	Default
description	String	Business objective description	No	N/A
owners	String	Owners of the Business objective	No	N/A
priority	Integer	Business objective priority	No	N/A
statusCode	String	Value of: <ul style="list-style-type: none"> • IN_PROGRESS • ACHIEVED • CANCELLED 	No	IN_PROGRESS

WeightInfo

WeightInfo defines the weight code value map. This data type is used in program settings. This data type contains the following properties:

Property	Type	Description	Required	Default
code	String	Weight code	Yes	N/A
weight	Integer	Weight value	No	N/A

SettingsInfo

SettingsInfo defines program settings. This data type contains the following properties:

Property	Type	Description	Required	Default
financialManagementEnabled	Boolean	Financial management enable flag. Set to true if you enable Financial management for the program.	No	N/A
portfolioManagementEnabled	Boolean	Portfolio management	No	N/A

Property	Type	Description	Required	Default
		enable flag. Set to true if you enable Portfolio management for the program.		
capitalizationEnabled	Boolean	Capitalization enable flag. Read-only property Specifies whether the Capitalization tracking for the program is enabled or not.	No	N/A
issueEnabled	Boolean	Issue track enable flag. Set to true if you enable Issue tracking for the program.	No	N/A
issuePriorityWeights	"WeightInfo" on the previous page	Issue priority weight values	No	N/A
issueYellowThreshold	Integer	Issue yellow threshold value	No	N/A
issueRedThreshold	Integer	Issue red threshold value	No	N/A
riskEnabled	Boolean	Risk track enable flag. Set to true if you enable Risk tracking for the program.	No	N/A
riskProbabilityWeights	"WeightInfo" on the previous page	Risk probability weight values	No	N/A
riskImpactWeights	"WeightInfo" on the previous page	Risk impact weight values	No	N/A
riskYellowThreshold	Integer	Risk yellow threshold value	No	N/A
riskRedThreshold	Integer	Risk red threshold value	No	N/A
scopeChangeEnabled	Boolean	Scope change track enable flag	No	N/A

Property	Type	Description	Required	Default
scopeChangeSeverityWeights	"WeightInfo" on page 208	Scope change severity weight values	No	N/A
scopeChangeYellowThreshold	Integer	Scope change yellow threshold value	No	N/A
scopeChangeRedThreshold	Integer	Scope change red threshold value	No	N/A

ProgramSearchFiltersInfo

ProgramSearchFilterInfo defines schedule information for a task. This data type contains the following properties:

Property	Type	Description	Required	Default
startRow	Integer	Start row	No	1
maxResults	Integer	Values range from 1 to 1,000 inclusively.	No	50
nameStartsWith	String	Text that the project name start with	No	N/A
programTypeName ^a	String	Program Type Name	No	N/A
programTypeId ^a	Long	Program type ID	No	N/A
statusCodes	"StatusCodeInfo" on page 201	Program status	No	N/A
managers	common:User	Program managers	No	N/A
createdBy	common:User	Create user	No	N/A
containsContent	"ContentItemIdentifierInfo" on page 202	Content	No	N/A

Property	Type	Description	Required	Default
containsBusinessObjectives	"BusinessObjectiveIdentifier Info" on page 207	Business objective	No	N/A
showOnlyPortfolioManagementEnabled	Boolean	Portfolio management enabled flag. If this flag is set to true, the searchProgram operation only returns the programs that have the Portfolio Management enabled.	No	N/A
priorityFrom	Integer	Value of the Priority From field	No	N/A
priorityTo	Integer	Value of the Priority To field	No	N/A
creationDateFrom	Date	Value of the Create Date From field	No	N/A
creationDateTo	Date	Value of the Create Date To field	No	N/A
a. You can either specify the programTypeName property or the programTypeId property, but not both of them.				

Note

Note defines note information. This data type contains the following properties:

Property	Type	Description	Required	Default
userName	String	Name of user who creates the note	No	N/A
date	Date	Create date and time of the note	No	N/A
text	String	Note text	No	N/A

Operations

The following operations are included in Program Management Web services:

- ["createProgram" below](#)
- ["readProgram" on page 214](#)
- ["updateProgram" on page 216](#)
- ["deleteProgram" on page 218](#)
- ["searchProgram" on page 219](#)

createProgram

Purpose

This operation creates a program in PPM Center.

Function

This operation creates a program, identified by the program name and program type name (or program type ID).

To perform this operation, the user must meet one of the following conditions:

- Have the Edit All Programs access grant.
- Have both the Create Programs access grant and the Edit Programs access grant.

Related Information

Data Type: ["ProgramInfo" on page 200](#)

Input

A ProgramInfo object with the following required property:

- name
- program type name (or program type ID)

All the other properties are optional. Specify those properties only when you need to create a program with the corresponding information.

Return

Program ID will be returned if the program is created successfully.

Java Examples

See `webservice_toolkit`:

```
java\client\src\examples\pgm\ProgramServiceClient.java
```

testCreateProgram()

```
ProgramServiceStub stub = new ProgramServiceStub(ctx, WSURL);
CreateProgramDocument doc =
CreateProgramDocument.Factory.newInstance();
ProgramInfo program =
doc.addNewCreateProgram().addNewProgram();
program.setName(PROGRAM_NAME);
Managers managers = program.addNewManagers();
User manager = managers.addNewManager();
// Set a manager (user with ID=1) on the program. (Optional)
manager.setId(BigInteger.valueOf(1));
managers.setManagerArray(new User[] { manager });
program.setProgramTypeName("Enterprise");
```

```
CreateProgramResponseDocument responseDoc =  
stub.createProgram(doc);  
long programId =  
responseDoc.getCreateProgramResponse().getProgramId();
```

Errors and Exceptions

Possible root cause descriptions:

Exception/Message Code	Message	Cause(s)	Possible Corrective Action
NameAlreadyExistsException	A Program with the name <i><name></i> already exists. Please enter a different name.	The name is already used.	Use another name.
AuthorizationException	You do not have the privilege to take this action. Please consult your PPM Administrator.	The user has no access granted to create programs.	Grant the appropriate access to the user. For more information, see "Function" on page 212
ITGException	Unable to get program type for program <i><program name></i>	Program type ID or name is invalid.	Validate program type ID or name.

readProgram

Purpose

This operation reads a program in PPM Center.

Function

This operation reads a program and the related information, identified by the program ID.

To perform this operation, the user must have one of the following access grants:

- View Programs
- Edit Programs
- Edit All Programs

Related Information

Operation: ["createProgram" on page 212](#)

Input

Program ID

Return

A ["ProgramInfo" on page 200](#) object.

Java Examples

```
ProgramServiceStub stub = new ProgramServiceStub(ctx, WSURL);
ReadProgramDocument doc =
ReadProgramDocument.Factory.newInstance();
ReadProgram readProgram = doc.addNewReadProgram();
readProgram.setProgramId(programId);
ReadProgramResponseDocument responseDoc =
stub.readProgram(doc);

return responseDoc.getReadProgramResponse().getProgram();
```

Errors and Exceptions

Possible root cause descriptions:

Exception/Message Code	Message	Cause(s)	Possible Corrective Action
ObjectNotFoundException	No program found	The program ID or name is invalid.	Validate the program ID or name.
AuthorizationException	You do not have the privilege to take this action. Please consult your PPM Administrator.	The user has no access granted to read programs.	Grant the appropriate access to the user. For more information, see "Function" on page 214

updateProgram

Purpose

This operation updates a program in PPM Center.

Function

This operation updates a program according to the ProgramInfo object you input.

To perform this operation, the user must meet one of the following conditions:

- Have the Edit All Programs access grant.
- Be the manager of the program and have the Edit Programs access grant.

Related Information

Operation: ["createProgram" on page 212](#)

Input

A ["ProgramInfo" on page 200](#) object.

Return

If the operation is successful, nothing will be returned to the client.

Java Examples

See `webservice_toolkit`:

```
java\client\src\examples\pgm\ProgramServiceClient.java
```

```
updateProgram()
```

Errors and Exceptions

Possible root cause descriptions:

Exception/Message Code	Message	Cause(s)	Possible Corrective Action
ObjectNotFoundException	No program found	The program ID or name is invalid.	Validate the program ID or name.
ITGException	Unable to get program type for program <i><program name></i>	Program type ID or name is invalid.	Validate the program type ID or name.
AuthorizationException	You do not have the privilege to take this action. Please consult your PPM Administrator.	The user has no access granted to update programs.	Grant the appropriate access to the user. For more information, see "Function" on the previous page

Exception/Message Code	Message	Cause(s)	Possible Corrective Action
ProgramRollupableConversionException/ pgm.exception.non-rollupable	Portfolio Management cannot be disabled for this program because it is already included in a portfolio.	Program is already included in a portfolio.	Remove the program from the associated portfolio.
ProgramRollupableConversionException/ pgm.exception.rollupable	Portfolio Management cannot be enabled for this program because some of its content is already included in a portfolio or in another program that has Portfolio Management enabled.	Projects in this program is already included in a portfolio.	Validate the associated projects for this program.

deleteProgram

Purpose

This operation deletes a program in PPM Center.

Function

This operation deletes a program, identified by the program ID.

To perform this operation, the user must meet one of the following conditions:

- Have the Edit All Programs access grant.
- Be the manager of the program and have the Edit Programs access grant.

Input

Program ID

Return

If the operation is successful, nothing will be returned to the client.

Java Examples

```
ProgramServiceStub stub = new ProgramServiceStub(ctx, WSURL);
DeleteProgramDocument doc =
DeleteProgramDocument.Factory.newInstance();
DeleteProgram deleteProgram = doc.addNewDeleteProgram();
deleteProgram.setProgramId(programId);
DeleteProgramResponseDocument responseDoc =
stub.deleteProgram(doc);
```

Errors and Exceptions

Possible root cause descriptions:

Exception/Message Code	Message	Cause(s)	Possible Corrective Action
ObjectNotFoundException	No program found	The program ID or name is invalid.	Validate the program ID or name.
AuthorizationException	You do not have the privilege to take this action. Please consult your PPM Administrator.	The user has no access granted to delete programs.	Grant the appropriate access to the user. For more information, see "Function" on the previous page

searchProgram

Purpose

This operation searches a program by parameters in PPM Center.

Function

This operation searches programs by using the ["ProgramSearchFiltersInfo" on page 210](#) object you specify.

To perform this operation, the user must have one of the following access grants:

- View Programs
- Edit Programs
- Edit All Programs

Input

A ["ProgramSearchFiltersInfo" on page 210](#) object.

Return

A list of the program IDs that meet the search criteria.

Java Examples

See `webservice_toolkit`:

```
java\client\src\examples\pgm\ProgramServiceClient.java  
testSearchPrograms ()
```

Errors and Exceptions

Possible root cause descriptions:

Exception/Message Code	Message	Cause(s)	Possible Corrective Action
AuthorizationException	You do not have the privilege to take this action. Please consult your PPM Administrator.	The user has no access granted to search programs.	Grant the appropriate access to the user. For more information, see "Function" on the previous page

Chapter 7: Resource Management Web Services

- "Overview" below
- "References" below
- "Data Types" on the next page
- "Operations" on page 236

Overview

Resource Management Web services provides the following operations in PPM Center:

- Create, update, search, and fetch resource pools
- Get and set resource participation in resource pools
- Create roles and skills

Note: For more information about general Resource Management terms and concepts, see the *Resource Management User's Guide*.

References

- Data types:
`webservice_toolkit\java\conf\xsd\Resource.xsd`
- Operations:
`webservice_toolkit\java\conf\wsdl\ResourceService.wsdl`
- Sample code:
`webservice_toolkit\java\client\src\examples\rm`

Data Types

Resource Management Web services includes the following data types:

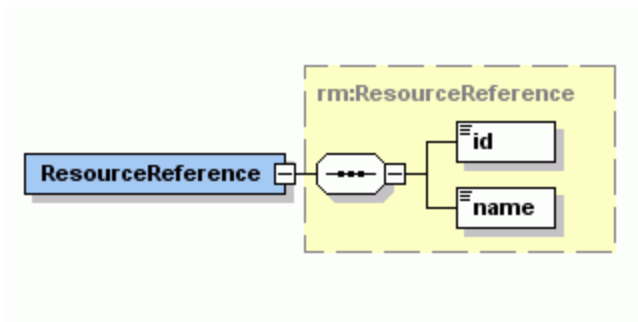
- ["ResourceReference" below](#)
- ["RegionReference" on the next page](#)
- ["OrgUnitReference" on page 225](#)
- ["RoleReference" on page 225](#)
- ["SkillReference" on page 226](#)
- ["ResourcePoolReference" on page 226](#)
- ["ResourcePoolAccessControlBean" on page 227](#)
- ["ResourcePool" on page 228](#)
- ["ResourcePoolSearchFilter" on page 230](#)
- ["Role" on page 232](#)
- ["Skill" on page 233](#)
- ["ResourceParticipation" on page 234](#)
- ["ResourceDistributionGroup" on page 235](#)
- ["ResourcePoolDistribution" on page 235](#)

ResourceReference

ResourceReference is a value object that specifies the ID and the name of a resource. It is used in ["ResourcePool" on page 228](#) and ["ResourceParticipation" on page 234](#).

Note: WSCostRuleBean is used as the INPUT in the following operation:

- ["getResourceParticipation" on page 255](#)

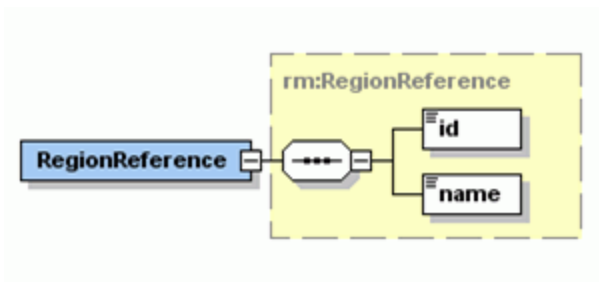


Property	Type	Description	Required	Default
id	Long	ID of the resource. A value that is typically greater than 30,000.	No	N/A
name	String	Name of the resource.	No	N/A

RegionReference

RegionReference is a value object that specifies the ID and the name of a region. It is used in ["ResourcePoolSearchFilter" on page 230](#) and ["ResourcePool" on page 228](#).

Note: For more information about the RegionReference, see ["ResourcePool" on page 228](#).

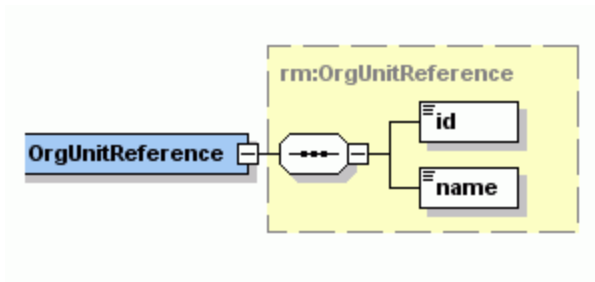


Property	Type	Description	Required	Default
id	Long	ID of the region.	No	N/A
name	String	Name of the region.	No	N/A

OrgUnitReference

OrgUnitReference is a value object that specifies the ID and the name of an organizational unit. It is used in "[ResourcePoolSearchFilter](#)" on page 230 and "[ResourcePool](#)" on page 228.

Note: For more information about the OrgUnitReference, see "[ResourcePool](#)" on page 228.



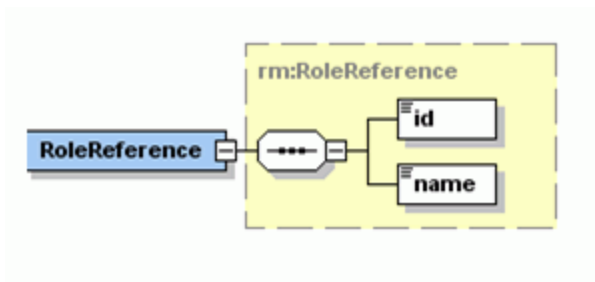
Property	Type	Description	Required	Default
id	Long	ID of the org unit.	No	N/A
name	String	Name of the org unit.	No	N/A

RoleReference

RoleReference is a value object that specifies the ID and the name of a role.

Note: RoleReference is used as the OUTPUT in the following operation:

- Operation "[createRoles](#)" on page 263



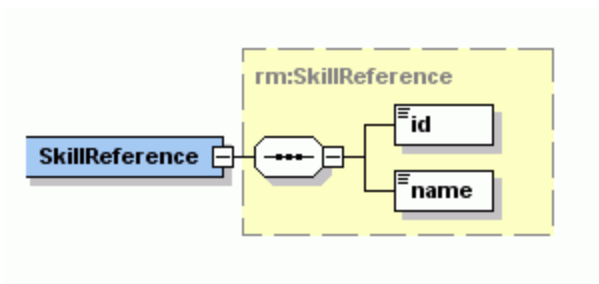
Property	Type	Description	Required	Default
id	Long	ID of the role.	No	N/A
name	String	Name of the role.	No	N/A

SkillReference

SkillReference is a value object that specifies the ID and the name of a skill.

Note: SkillReference is used as the OUTPUT in the following operation:

- Operation "[createRoles](#)" on page 263



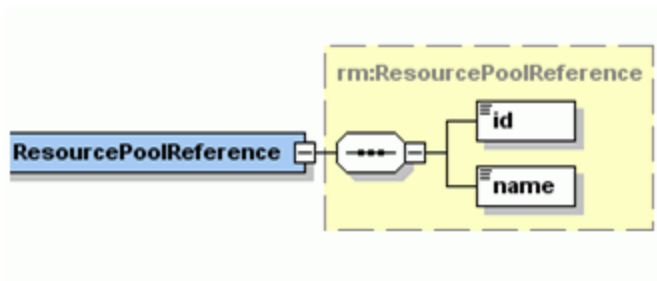
Property	Type	Description	Required	Default
id	Long	ID of the skill.	No	N/A
name	String	Name of the skill.	No	N/A

ResourcePoolReference

ResourcePoolReference is a value object that specifies the ID and the name of a resource pool.

Note: ResourcePoolReference is used as the OUTPUT in the following operation:

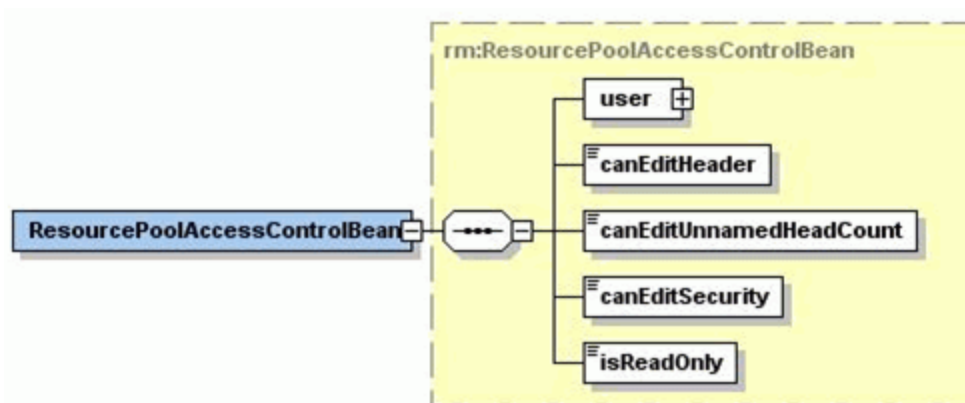
- Operation "[createResourcePools](#)" on page 237



Property	Type	Description	Required	Default
id	Long	ID of the resource pool.	No	N/A
name	String	Name of the resource pool.	No	N/A

ResourcePoolAccessControlBean

ResourcePoolAccessControlBean is a value object that specifies the user access privileges of a resource pool.



Property	Type	Description	Required	Default
user	ResourceReference	User/resource whose privileges the bean describes.	Yes	N/A
canEditHeader	boolean	Value that specifies whether the user can edit the resource pool header.	No	N/A

Property	Type	Description	Required	Default
canEditUnnamedHeadCount	boolean	Value that specifies whether the user can edit unnamed head count in resource pool.	No	N/A
canEditSecurity	boolean	Value that specifies whether the user can edit resource pool security settings.	No	N/A
isReadOnly	boolean	Value that specifies whether the user is read only.	No	N/A

ResourcePool

ResourcePool is a value object that specifies the resource pool information. The bean is used to create resource pools in the system.

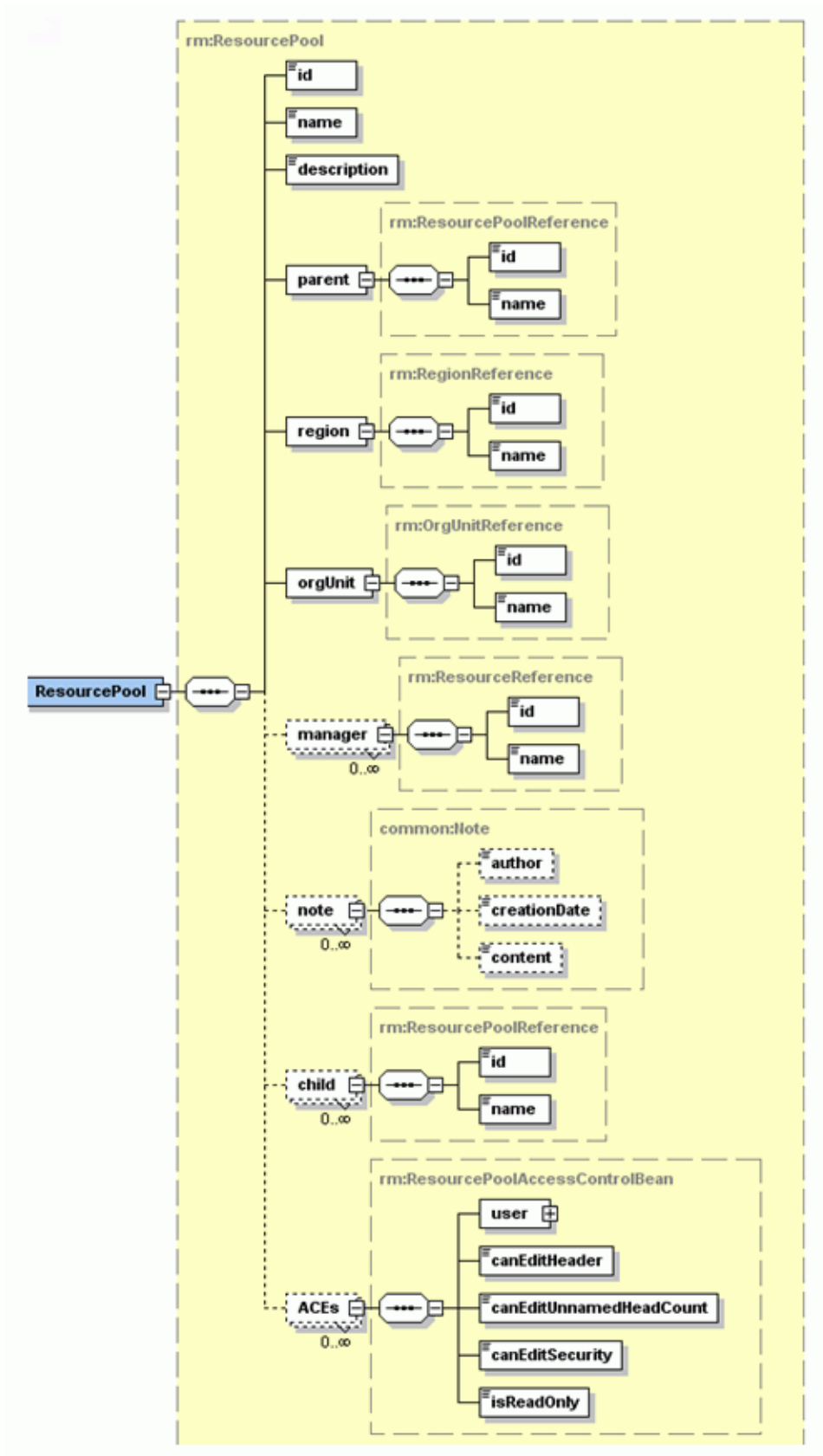
Note:

ResourcePool is used as the INPUT in the following operations:

- Operation "[createResourcePools](#)" on page 237
- Operation "[updateResourcePools](#)" on page 249

ResourcePool is used as the OUTPUT in the following operations:

- Operation "[searchResourcePools](#)" on page 243
- Operation "[getResourcePools](#)" on page 247



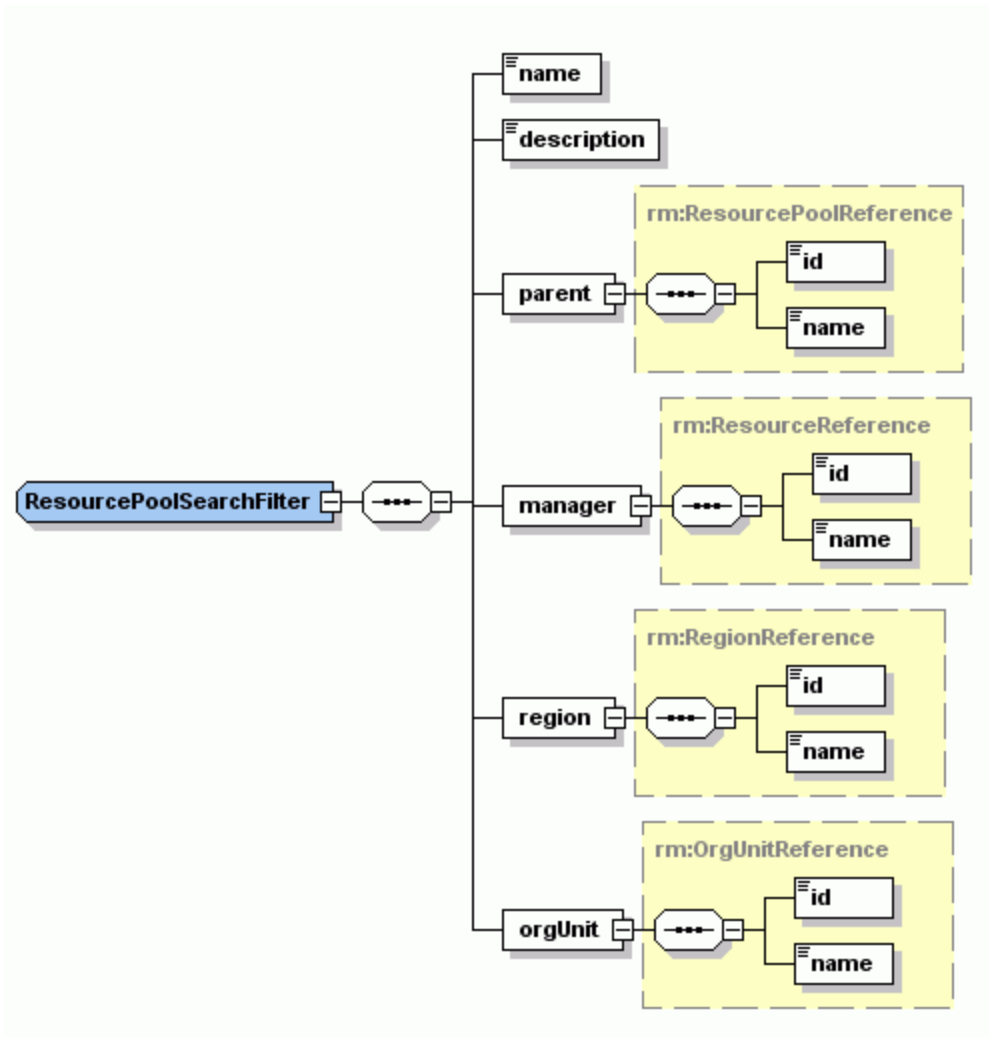
Property	Type	Description	Required	Default
id	Long	ID of the resource pool. A value which is typically greater than 30,000.	No	N/A
name	String	Name of the resource pool. Name cannot be more than 260 characters.	Yes	N/A
description	String	Description of the resource pool. Description cannot be more than 650 characters.	No	N/A
parent	ResourcePoolReference	This is a ResourcePoolReference value that contains the ID and name of the parent Resource Pool.	No	N/A
region	RegionReference	This is a RegionReference value that contains the ID and name of the pool's region.	Yes	N/A
orgUnit	OrgUnitReference	This is a OrgUnitReference value that contains the ID and name of the org unit the resource is used for.	No	N/A
manager	List	This is a list of ResourceReference values that contains the ID and names of the resource pool managers.	No	N/A
note	List	This is a list of NoteBean values that contains Author, Content, and CreationDate.	No	N/A
child	List	This is a list of ResourcePoolReference values that contains the IDs and names of the children resource pools.	No	N/A
ACEs	List	This is a list of ResourcePoolAccessControlBean values.	No	N/A

ResourcePoolSearchFilter

The search filter criteria bean that web services uses to search for resource pools in the system.

Note: ResourcePoolSearchFilter is used as the INPUT in the following operations:

- Operation "[searchResourcePools](#)" on page 243



Property	Type	Description	Required	Default
name	String	Name of the resource pool.	No	N/A
description	String	Description of the resource pool.	No	N/A
parent	ResourcePoolReference	This is a ResourcePoolReference value that contains the ID and name of the parent Resource Pool.	No	N/A

Property	Type	Description	Required	Default
region	RegionReference	This is a RegionReference value that contains the ID and name of the pool's region.	Yes	N/A
orgUnit	OrgUnitReference	This is a OrgUnitReference value that contains the ID and name of the org unit the resource belongs to.	No	N/A
manager	List	This is a list of ResourceReference values that contains the ID and name of the resource pool manager.	No	N/A

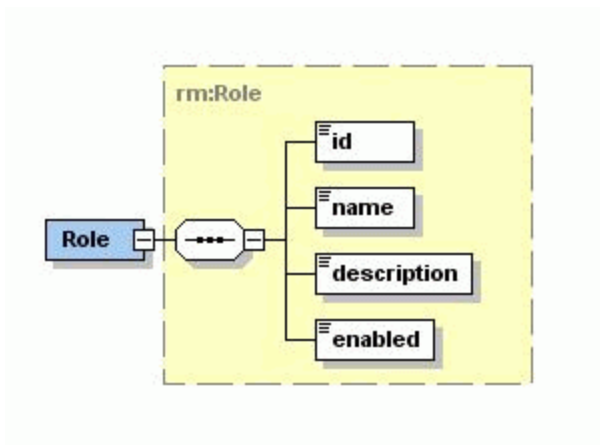
Role

Role is a value object that specifies the role information. It is also used to create roles.

Note:

Role is used as the INPUT in the following operations:

- Operation "[createRoles](#)" on page 263



Property	Type	Description	Required	Default
id	Long	ID of the role.	No	N/A
name	String	Name of the role.	Yes	N/A

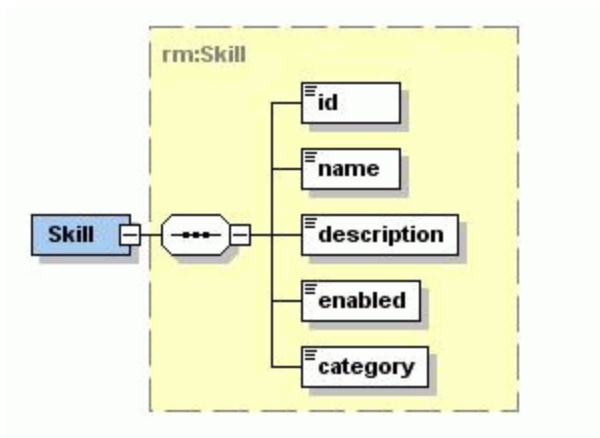
Property	Type	Description	Required	Default
description	String	Description of the role.	No	N/A
enabled	boolean	Boolean value - true if the role is enabled.	Yes	N/A

Skill

Skill is a value object that specifies the skill information. It can also be used to create skills.

Note: Skill is used as the INPUT in the following operations:

- Operation "[createSkills](#)" on page 266



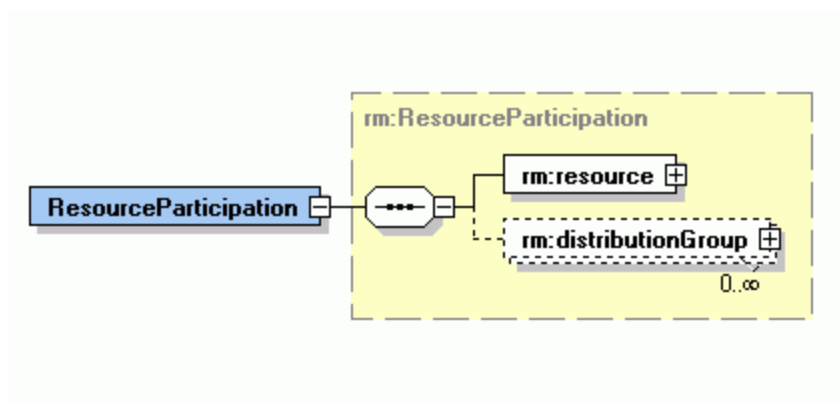
Property	Type	Description	Required	Default
id	Long	ID of the skill.	No	N/A
name	String	Name of the skill.	Yes	N/A
description	String	Description of the skill.	No	N/A
enabled	boolean	Boolean value - true if the skill is enabled.	Yes	N/A
category	String	Category of the skill.	No	N/A

ResourceParticipation

ResourceParticipation is a value object that specifies the participation of a resource in one or more resource pools. It can also be used to get resource participations.

Note:

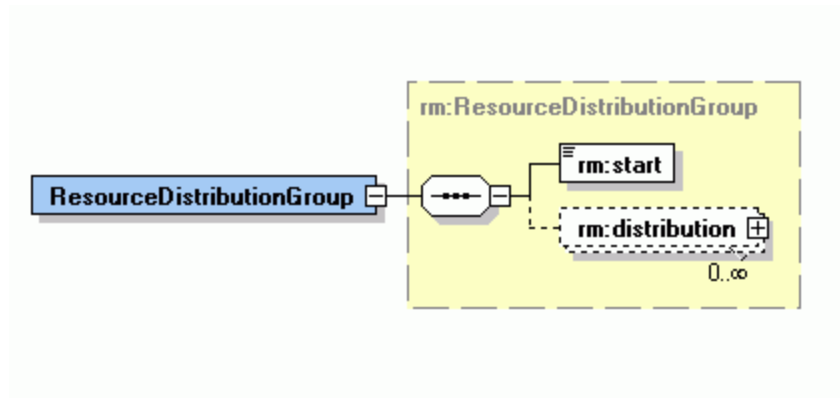
- ResourceParticipation is used as the INPUT in the following operations:
 - Operation ["setResourceParticipation" on page 258](#)
- ResourceParticipation is used as the OUTPUT in the following operation:
 - Operation ["getResourceParticipation" on page 255](#)



Property	Type	Description	Required	Default
resource	"RoleReference" on page 225	This is a ResourceReference value that contains the ID and name of the resource.	Yes	N/A
distributionGroup	"ResourceDistributionGroup" on the next page	This is a ResourceDistributionGroup value that contains the startDate and the distributions of the resource pools.	No	N/A

ResourceDistributionGroup

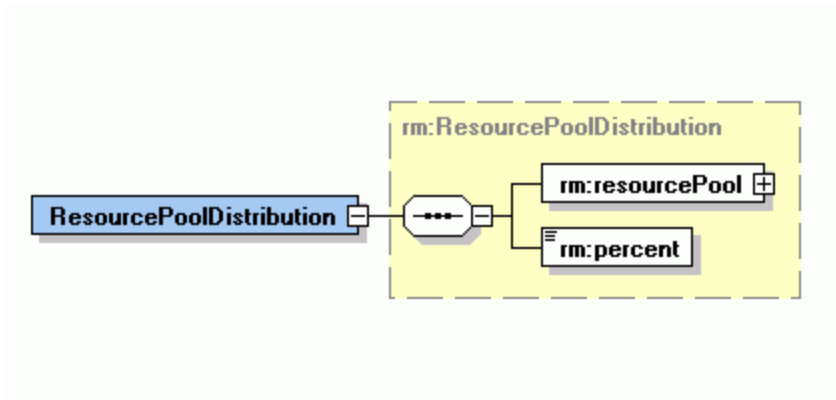
ResourceDistributionGroup is a value object that specifies the distribution of a resource among one or more resource pools starting on a given date.



Property	Type	Description	Required	Default
start	Date	This value specifies the start date for the given resource.	No	N/A
distribution	"ResourceDistributionGroup" above	This is a ResourcePoolDistribution value that contains the resource pool information and the allocation percentage.	No	N/A

ResourcePoolDistribution

ResourcePoolDistribution is a value object that specifies the allocation of a resource to a specific resource pool.



Property	Type	Description	Required	Default
resourcePool	"ResourcePoolReference" on page 226	This is a ResourcePoolReference value that contains the ID and name of the specific Resource Pool.	No	N/A
percent	Double	This value specifies how much percentage a resource is allocated to the resource pool.	No	N/A

Operations

The following operations are included in Resource Management Web services:

- ["createResourcePools" on the next page](#)
- ["searchResourcePools" on page 243](#)
- ["getResourcePools" on page 247](#)
- ["updateResourcePools" on page 249](#)
- ["getResourceParticipation" on page 255](#)
- ["setResourceParticipation" on page 258](#)
- ["createRoles" on page 263](#)
- ["createSkills" on page 266](#)

createResourcePools

Purpose

This operation creates one or more new resource pool objects in PPM Center.

Function and Parameters

This operation creates one or more (maximum 1,000) new resource pool objects, identified by resource pool ID in PPM Center.

The user performing this operation must have the following access grants:

- Create Resource Pools
- Edit Resource Pool or Edit All Resource Pools

Required Fields:

- Name
- Region

Optional Fields:

- ID
- Description
- Parent
- Org unit
- Manager
- Note
- Child
- Access Control

If these optional fields are not set, no defaults are used.

Limitations

This operation has the following limitations:

- User data cannot be created.
- This operation assumes basic data exists in the database. The following data must exist in any production PPM Center instance.
 - Resources
 - Users
 - Parent resource pool
 - Org Unit
 - Region

All values provided in the parameters must comply with what is expected in PPM Center. The possible values are provided above when each parameter is described.

- If a data problem occurs with any data provided for the resource pools, the entire data set is rejected. No resource pools are created.

Related Information

["updateResourcePools" on page 249](#) - update fields of an existing resource pool.

Input

An array of ["ResourcePool" on page 228s](#).

Return

An array of ["ResourcePoolReference" on page 226s](#).

Java Interface

```
CreateResourcePoolsResponseDocument createResourcePools(  
CreateResourcePoolsDocument in)
```

Parameters	Description
CreateResourcePoolsDocument	<p>Wrapper of the resource pool array object (WSResourcePoolBean[]).</p> <p>See the following example for the construction.</p> <p>The bean includes the following fields:</p> <ul style="list-style-type: none">• Long id;• String name;• String description;• ResourcePoolRefernce parent;• RegionReference region;• OrgUnitReference orgUnit;• List manager;• List Notes;• ResourcePoolRefernce[] child;• ResourcePoolAccessControlBean[] aces;
CreateResourcePoolsResponseDocument	<p>Wrapper for the resource pool ref array (ResourcePoolReference[]).</p> <p>See the following example for retrieving the bean and its fields.</p>

Java Examples

Example: create a new resource pool.

```
/**  
 * Test creating a resource pool with values for all fields, reading it back out  
 and searching for it.  
 */  
public long testCreateResourcePool() throws Exception {
```

```
        // create the resource pool object
ResourcePool resourcePool1 =
createTestResourcePoolObject("TestResourcePool1");
ResourcePool resourcePool2 =
createTestResourcePoolObject("TestResourcePool2");

        // call the service
ResourcePoolReference[] resourcePoolReference =
createResourcePools(new ResourcePool[] { resourcePool1,
resourcePool2 });

        // return first pool ID
return resourcePoolReference[0].getId();

    }
/**
 * This is a wrapper method around the createResourcePools
web service. It handles the details of creating
 * the document, invoking the web service and unwrapping the
response.
 *
 * @param resourcePools Zero or more resource pools to
create in PPM.
 * @return A resource pool reference for each object that
was successfully created.
 * @throws Exception
 */
ResourcePoolReference[] createResourcePools(ResourcePool[]
resourcePools) throws Exception {
    ResourceServiceStub service = new
ResourceServiceStub(ctx, WSURL);
    CreateResourcePoolsDocument createResourcePoolsDoc =
CreateResourcePoolsDocument.Factory.newInstance();
createResourcePoolsDoc.addNewCreateResourcePools().setResourceP
oolArray(resourcePools);
    CreateResourcePoolsResponseDocument responseDocCreate =
service.createResourcePools(createResourcePoolsDoc);

CreateResourcePoolsResponse responseCreate =
responseDocCreate.getCreateResourcePoolsResponse();
debugPrint(responseCreate, "create response");
return responseCreate.getResourcePoolRefArray();
}

    public static ResourcePool
createTestResourcePoolObject(String name) {

        ResourcePool resourcePool =
ResourcePool.Factory.newInstance();
```



```
// resource pool name
resourcePool.setName(name);

// description
resourcePool.setDescription("A resource pool created
programmatically through web services ");

// region
resourcePool.addNewRegion().setName("America");

// pool managers
resourcePool.addNewManager().setName("admin");

// set parent pool
resourcePool.addNewParent().setName("parent pool
name");

// add children pool
resourcePool.addNewChild().setName("child pool name");

// org unit
resourcePool.addNewOrgUnit().setName(" org unit name");

// add notes
Note note = resourcePool.addNewNote();
note.setContent("Note content");
note.setAuthor("admin");

// set access control list
for (int i = 0; i < usersInACL.length; i++) {
    ResourcePoolAccessControlBean acb =
resourcePool.addNewACEs();
    ResourceReference aceUser = acb.addNewUser();
    aceUser.setName(usersInACL[i]);
    acb.setCanEditHeader(true);
    acb.setCanEditSecurity(true);
    acb.setCanEditUnnamedHeadCount(false);
}
return resourcePool;
}
}
```

Errors and Exceptions

When an error occurs on this operation, you will receive a description of the root cause in the log or in the response message.

The server log file content is similar to the following:

```
Exception in thread "main" org.apache.axis2.AxisFault:
<exception:exceptionDetails xmlns:exception="http://
www.mercury.com/ppm/ws/exception">
<exception:detail>[root cause description] </exception:detail>
</exception:exceptionDetails>
```

Possible root cause descriptions:

:

Message Code	Message	Cause(s)	Possible Corrective Action
error.ws.maxResourcePools	The maximum number of resource pools to process in a single web service transaction may not exceed 1000.	The number of resource pools exceeds 1,000.	Break up the list of resource pools into smaller chunks and process over multiple transactions.
error.ws.create.duplicateName	A resource pool with the name "{0}" already exists. Enter a different name.	A resource pool with the name already exists.	Enter a different name.
error.ws.create.cycle	Cannot create resource pool as it would introduce a cyclical parent-child relationship between the resource pools {0} and {1}.	Two resource pools have each other as parents.	Make sure that the resource pools do not have each other as parents.
error.ws.regionRequired	Resource pool region is required. Specify a region for each resource pool.	The region for a resource pool is not specified.	Specify the region for the resource pool.
error.ws.nameRequired	Resource pool name cannot be blank. Provide a name for each resource pool.	The resource pool name is not set for a pool.	Set the resource pool name.
error.ws.descriptionTooLong	Description cannot be longer than 650 characters.	Description of a resource pool is more than 650 characters.	Reduce length of description.
error.ws.nameTooLong	Name cannot be longer	Name of a	Reduce length of name.

Message Code	Message	Cause(s)	Possible Corrective Action
	than 260 characters.	resource pool is great than 260.	
N/A	Entity Validation errors if region, manager, resource pool does not exist.	Resource pool region, manager, or parent resource pool does not exist.	Ensure that Resource pool region, manager, and parent resource pool exist for all pools to be created before this web service call.

searchResourcePools

Purpose

This operation searches for resource pool objects in PPM Center based on certain criteria.

Function and Parameters

This operation returns all resource pool objects matching the search criteria from PPM Center.

The user performing this operation must have one of the following access grants:

- View Resource Pools
- View All Resource Pools
- Edit Resource Pool
- Edit All Resource Pools

Required Fields: None

Optional Fields:

- Name
- Description
- Parent
- Org unit
- Manager
- Region

If none of the optional fields are set, all resource pools are returned. If more than one criteria are specified, they are combined with each other by the AND logic.

Limitations

Performance might be an issue if more than a few thousand resource pools are read at once.

Input

ResourcePoolSearchFilter

Return

["ResourcePool"](#) on page 228 matching the criteria.

Java Interface

```
CreateResourcePoolsResponseDocument createResourcePools(  
CreateResourcePoolsDocument in)
```

Parameters	Description
SearchResourcePoolsDocument	Wrapper of the resource pool search filter. (ResourcePoolSearchFilter)
SearchResourcePoolsResponseDocument	Wrapper for the resource pool array. (WSResourcePoolBean[])

Java Examples

Example: search resource pools.

```
/**
 * Create a search filter with all fields set to match the nonnull
 * fields of a *specific resource pool, thereby creating a
 * search filter that should match only that *one resource pool.
 * @param target The resource pool that the filter will be
 * made for.
 * @return the search filter.
 */
ResourcePoolSearchFilter createSearchFilter(ResourcePool
target) {
    ResourcePoolSearchFilter filter =
ResourcePoolSearchFilter.Factory.newInstance();
    filter.setName(target.getName()); // required field
    filter.setDescription(target.getDescription());
    filter.addNewRegion().setName(target.getRegion().getName());
// required field
    if (target.sizeOfManagerArray() > 0) {
        // use the first of the resource pool's managers in the
search query
filter.addNewManager().setName(target.getManagerArray(0).getNam
e());
    }
    if (target.getOrgUnit() != null) {
filter.addNewOrgUnit().setName(target.getOrgUnit().getName());
    }
    if (target.getParent() != null) {
filter.addNewParent().setName(target.getParent().getName());
    }
    return filter;
}
/**
 * This is a wrapper method around the searchResourcePools web
 * service, which reads *resource pools based on filter criteria.
 * This method handles the details of creating *the document,
 * invoking the service and unwrapping the response.
 *
 * @param filter A bunch of filter fields to narrow down the
 * search.
 * @return Zero or more resource pools that match the
 * search criteria.
```

```
* @throws Exception
*/
ResourcePool[] searchResourcePools(ResourcePoolSearchFilter
filter) throws Exception {
    ResourceServiceStub service = new ResourceServiceStub(ctx,
WSURL);
    SearchResourcePoolsDocument searchResourcePoolsDoc =
SearchResourcePoolsDocument.Factory.newInstance();

searchResourcePoolsDoc.addNewSearchResourcePools().setFilter(fi
lter);

    SearchResourcePoolsResponse searchResponse =
service.searchResourcePools(searchResourcePoolsDoc).getSearchRe
sourcePoolsResponse();

    return searchResponse.getResourcePoolArray();
}
```

Errors and Exceptions

When an error occurs on this operation, you will see a description of the root cause in the log or in the response message.

The server log file content is similar to the following:

```
Exception in thread "main" org.apache.axis2.AxisFault:
<exception:exceptionDetails xmlns:exception="http://
www.mercury.com/ppm/ws/exception">
<exception:detail>[root cause description] </exception:detail>
</exception:exceptionDetails>
```

Possible root cause descriptions

:

Message Code	Message	Cause(s)	Possible Corrective Action
exception.authorization	You do not have the privilege to take this action. Consult your PPM Administrator.	You do not have the required access grants for this operation.	Verify that your account has the required access grants.

getResourcePools

Purpose

This operation fetches resource pool objects in PPM Center based on the resource pool IDs.

Function and Parameters

This operation returns one or more resource pool objects according to a specified list of resource pool IDs from PPM Center. If no IDs are specified or if there are no resource pools with those IDs, no records are returned.

The user performing this operation must have one of the following access grants:

- View Resource Pools
- View All Resource Pools
- Edit Resource Pool
- Edit All Resource Pools

Required Fields:

- Resource pool IDs

Optional Fields: None

Limitations

Performance might be an issue if more than a few thousand resource pools are read at once.

Input

Resource pool IDs

Return

An array of ["ResourcePool"](#) on page 228.

Java Interface

```
CreateResourcePoolsResponseDocument createResourcePools(  
CreateResourcePoolsDocument in)
```

Parameters	Description
GetResourcePoolsDocument	Wrapper array of resource pool IDs. To construct this parameter: long [] ids;
GetResourcePoolsResponseDocument	Wrapper for the resource pool array. (WSResourcePoolBean[])

Java Examples

Example: read resource pools.

```
/**  
 * This is a wrapper method around the getResourcePools web  
 service, which reads resource pools by ID. This  
 * method handles the details of creating the document, invoking  
 the web service and unwrapping the response.  
 *  
 * @param ids The primary keys of the resource pools to  
 retrieve.  
 * @return The resource pools that match the given primary  
 keys; primary keys that don't correspond to entities in the  
 database are silently ignored.  
 * @throws Exception  
 */  
ResourcePool[] getResourcePools(long[] ids) throws Exception {  
    ResourceServiceStub service = new ResourceServiceStub(ctx,  
WSURL);  
  
    GetResourcePoolsDocument getResourcePoolsDoc =  
    GetResourcePoolsDocument.Factory.newInstance();
```



```
GetResourcePools getResourcePools =  
getResourcePoolsDoc.addNewGetResourcePools();  
  
getResourcePools.setResourcePoolIdArray(ids);  
  
GetResourcePoolsResponseDocument responseGet =  
service.getResourcePools(getResourcePoolsDoc);  
    debugPrint(responseGet, "get response");  
    return  
responseGet.getGetResourcePoolsResponse().getResourcePoolArray(  
);  
}
```

Errors and Exceptions

When an error occurs on this operation, you will see a description of the root cause in the log or in the response message:

The server log file content is similar to the following:

```
Exception in thread "main" org.apache.axis2.AxisFault:  
<exception:exceptionDetails xmlns:exception="http://  
www.mercury.com/ppm/ws/exception">  
<exception:detail>[root cause description] </exception:detail>  
</exception:exceptionDetails>
```

Possible root cause descriptions:

Message Code	Message	Cause(s)	Possible Corrective Action
exception.authorization	You do not have the privilege to take this action. Consult your PPM Administrator.	You do not have the required access grants for this operation.	Verify that your account has the required access grants.

updateResourcePools

Purpose

This operation updates existing resource pool objects in PPM Center.

Function and Parameters

This operation returns one or more (maximum 1,000) updated resource pool objects from PPM Center according to a specified list of resource pool objects. The user performing this operation must have one of the following access grants:

- Edit Resource Pool
- Edit All Resource Pools

Required Fields:

- ID

Optional Fields:

- Name
- Region
- Description
- Parent
- Org unit
- Manager
- Note
- Child
- ACEs

Limitations

This operation has the following limitations:

- User data cannot be created.
- This operation assumes basic data exists in the database. The following data must exist in any production PPM Center instance:

- Resources
- Users
- Parent resource pool
- Org Unit
- Region

All values provided in the parameters must comply with what is expected in PPM Center. The possible values are provided above when each parameter is described.

- If a data problem occurs with any data provided for the resource pools, the entire data set is rejected. No resource pools are created.

Related Information

["createResourcePools" on page 237](#) - to create new resource pools.

Input

["ResourcePool" on page 228](#)

Return

None

Java Interface

```
UpdateResourcePoolsResponseDocument updateResourcePools(UpdateResourcePoolsDocument in)
```

Parameters	Description
UpdateResourcePoolsDocument	Wrapper array of resource pool objects. (WSResourcePoolBean [])

Java Examples

Example: update a resource pool

```
/**
 * Test updating a resource pool.
 */
public void testUpdateResourcePool(long poolId) throws
Exception {

    // query the resource pool
    ResourcePool[] resourcePools = getResourcePools(new long[]
{poolId});
    ResourcePool pool = resourcePools[0];

    // update the pool object
    pool.setDescription("New version of the pool");

    // remove existing manager
    pool.setManagerArray(new ResourceReference[0]);

    // add new manager
    pool.addNewManager().setName("user1");

    // remove note
    pool.setNoteArray(new Note[] {});

    // call the service
    updateResourcePools(new ResourcePool[] { pool} );

}

/**
 * This is a wrapper method around the updateResourcePools web
service, which modifies existing resource
 * pools. This method handles the details of creating the
document, invoking the web service and unwrapping
 * the response.
 * @param resourcePools
 * The changes that should be made to existing resource pools.
Resource pools are
 * identified by primary key, so the ID of each resource pool
must be non-null.
 * @throws Exception
 */
void updateResourcePools(ResourcePool[] resourcePools) throws
Exception {
    ResourceServiceStub service = new ResourceServiceStub(ctx,
WSURL);
```

```

        UpdateResourcePoolsDocument updateResourcePoolsDoc =
UpdateResourcePoolsDocument.Factory.newInstance();

updateResourcePoolsDoc.addNewUpdateResourcePools().setResourceP
oolArray(resourcePools);
        service.updateResourcePools(updateResourcePoolsDoc);
    }
    
```

Errors and Exceptions

When an error occurs on this operation, you will see a description of the root cause in the log or in the response message:

```

Exception in thread "main" org.apache.axis2.AxisFault:
<exception:exceptionDetails xmlns:exception="http://
www.mercury.com/ppm/ws/exception">
<exception:detail>[root cause description] </exception:detail>
</exception:exceptionDetails>
    
```

Possible root cause descriptions:

:

Message Code	Message	Cause(s)	Possible Corrective Action
error.ws.update.idrequired	When updating a resource pool you must provide the ID of the resource pool you wish to modify. The resource pool name is not sufficient.	The pending update Resource pool ID is null.	Enter the resource pool ID.
error.ws.maxResourcePools	The maximum number of resource pools to process in a single web service transaction may not exceed 1000.	The number of resource pools exceeds 1,000.	Break up the list of resource pools into smaller chunks and process over multiple transactions.
error.ws.create.duplicateName	A resource pool with the name "{0}" already exists. Enter a different name.	A resource pool with the name already exists.	Enter a different name.
error.ws.create.cycle	Cannot create resource pool as it would introduce	Two resource	Make sure that the resource pools do not

Message Code	Message	Cause(s)	Possible Corrective Action
	a cyclical parent-child relationship between the resource pools {0} and {1}.	pools have each other as parents.	have each other as parents.
error.ws.regionRequired	Resource pool region is required. Specify a region for each resource pool.	The region for a resource pool is not specified.	Specify the region.
error.ws.nameRequired	Resource pool name cannot be blank. Provide a name for each resource pool.	The resource pool name is not set for a pool.	Set the resource pool name.
error.ws.descriptionTooLong	Description cannot be longer than 650 characters.	The description of a resource pool contains more than 650 characters	Reduce length of description.
error.ws.nameTooLong	Name cannot be longer than 260 characters.	The name of a resource pool is greater than 260.	Reduce length of name.
N/A	Entity Validation errors if region, manager, resource pool does not exist.	The resource pool region, manager, or parent resource pool does not exist.	Ensure that Resource pool region, manager, and parent resource pool exist for all pools to be created before this web service call.
exception.cannotLoadEntity	Cannot load the resource pool with the specified id: {1}. Resource Pool may be deleted by another user.	The resource pool does not exist.	Ensure the pending update resource pool exists.

getResourceParticipation

Purpose

This operation fetches participations of resources in various resource pools.

Function and Parameters

This operation returns an array of resource distribution groups from PPM Center according to a specified list of (maximum 1,000) resource IDs. The user performing this operation must have the following access grant:

- View Resource

Required Fields:

- ["ResourceReference" on page 223](#)

Optional Fields: None

Limitations:

- Maximum 1,000 ResourceReferences can be passed in a single invocation.

Related Information

["setResourceParticipation" on page 258](#) - to set resource participations

Input

An array of ["ResourceReference" on page 223](#).

Return

An array of ["ResourceParticipation" on page 234](#).

Java Interface

```
GetResourceParticipationResponseDocument getResourceParticipation  
(GetResourceParticipationDocument in)
```

Parameters	Description
GetResourceParticipationDocument	Wrapper array of resource references. (ResourceReference[])
GetResourceParticipationResponseDocument	Wrapper for the resource participation array. (ResourceParticipationBean[])

Java Examples

Example: get resource participation.

```
/**  
 * This is a wrapper method around the  
 getResourceParticipation web service, which reads the  
 participation of  
 * resources in resource pools. This method handles the  
 details of creating the document, invoking the web service  
 * and unwrapping the response.  
 *  
 * @param participation Zero or more  
 ResourceParticipation objects, which encapsulate the  
 participation details  
 * for a resource.  
 * @throws Exception  
 */  
  
public ResourceParticipation[]  
getResourceParticipation(ResourceReference[] resources) throws  
Exception {  
    ResourceServiceStub service = new  
ResourceServiceStub(ctx, RESOURCE_SERVICE);
```



```

GetResourceParticipationDocument getParticipationDoc =
GetResourceParticipationDocument.Factory.newInstance();

getParticipationDoc.addNewGetResourceParticipation().setResourceArray(resources);

return
service.getResourceParticipation(getParticipationDoc).getGetResourceParticipationResponse().getResourceParticipationArray();
    }
/**
 * Return a resource reference object with the given name
and ID.
 */
ResourceReference resourceRef(Long id, String name) {
    ResourceReference ref =
ResourceReference.Factory.newInstance();
    if (id != null) ref.setId(id.longValue());
    ref.setName(name);
    return ref;
}
    
```

Errors and Exceptions

When an error occurs on this operation, you will see a description of the root cause in the log or in the response message:

The server log file content is similar to the following:

```

Exception in thread "main" org.apache.axis2.AxisFault:
<exception:exceptionDetails xmlns:exception="http://
www.mercury.com/ppm/ws/exception">
<exception:detail>[root cause description] </exception:detail>
</exception:exceptionDetails>
    
```

Possible root cause descriptions:

Message Code	Message	Cause(s)	Possible Corrective Action
exception.authorization	You do not have the privilege to take this action. Consult your PPM Administrator.	You do not have the required access grants for this operation.	See above for list of access grants required for this operation.
error.ws.maxResources	The maximum number of	The number of	Break up the list of

Message Code	Message	Cause(s)	Possible Corrective Action
	resource to process in a single web service transaction may not exceed 1000.	resource exceeds 1,000.	resource into smaller chunks and process over multiple transactions.

setResourceParticipation

Purpose

This operation sets the participation of resources in various resource pools.

Function and Parameters

This operation takes an array of (maximum 1,000) resource participation objects to set the participation of resources in different resource pools. The user performing this operation must have one of the following access grants:

- Edit Resource Pools
- Edit All Resource Pools

The start date of each resource distribution group in the same participation must be unique.

The sum of percentages in each resource distribution group must be in the range of 0 to 100.

Resource distribution group start date cannot be outside the range of resource start and end date in PPM Center.

Each resource distribution group may contain at most one distribution per resource pool.

Required Fields:

- ["ResourceParticipation" on page 234](#)

Optional Fields: None

Limitations

Up to 1,000 resource participation objects can pass in a single invocation.

Related Information

["getResourceParticipation" on page 255](#) - to get resource participations.

Input

["ResourceParticipation" on page 234](#)

Return

None

Java Interface

```
SetResourceParticipationResponseDocument setResourceParticipation  
(SetResourceParticipationDocument in)
```

Parameters	Description
SetResourceParticipationDocument	Wrapper array of resource participations. (ResourceParticipationBean[])

Java Examples

Example: set resource participation

```
/**  
 * Test creating a resource participation for a resource and  
 then modifying it.  
 */
```

```
public void testSetResourceParticipation() throws Exception
{
    String resource = "user1";

    // create participation object
    ResourceParticipation p =
ResourceParticipation.Factory.newInstance();
    p.addNewResource().setName(resource);

    ResourceDistributionGroup group1 =
ResourceDistributionGroup.Factory.newInstance();
    group1.setStart(calendar("Jan 1, 2005"));
    group1.setDistributionArray(new
ResourcePoolDistribution[] {
        createDistribution("TestResourcePool1", 25),
        createDistribution("TestResourcePool2", 50)
    });

    ResourceDistributionGroup group2 =
ResourceDistributionGroup.Factory.newInstance();
    group2.setStart(calendar("Jan 1, 2006"));
    group2.setDistributionArray(new
ResourcePoolDistribution[] {
        createDistribution("TestResourcePool1", 5)
    });

    ResourceDistributionGroup group3 =
ResourceDistributionGroup.Factory.newInstance();
    group3.setStart(calendar("Jan 1, 2007"));
    group3.setDistributionArray(new
ResourcePoolDistribution[] {
        createDistribution("TestResourcePool1", 90),
        createDistribution("TestResourcePool2", 10)
    });

    p.setDistributionGroupArray(new
ResourceDistributionGroup[] { group1, group2, group3 });

    // call service
    setResourceParticipation(new ResourceParticipation[] { p
});
}
/**
 * This is a wrapper method around the
setResourceParticipation web service, which manages the
participation of
 * resources in resource pools. This method handles the
details of creating the document, invoking the web service
 * and unwrapping the response.
 *
 * @param participation Zero or more ResourceParticipation
```

```

objects, which encapsulate the participation details
    * for a resource.
    * @throws Exception
    */
    public void
    setResourceParticipation(ResourceParticipation[] participation)
    throws Exception {
        ResourceServiceStub service = new
    ResourceServiceStub(ctx, WSURL);
        SetResourceParticipationDocument setParticipationDoc =
    SetResourceParticipationDocument.Factory.newInstance();
    setParticipationDoc.addNewSetResourceParticipation().setResourc
    eParticipationArray(participation);
        service.setResourceParticipation(setParticipationDoc);
    }
    
```

Errors and Exceptions

When an error occurs on this operation, you will see a description of the root cause in the log or in the response message.

The server log file content is similar to the following:

```

Exception in thread "main" org.apache.axis2.AxisFault:
<exception:exceptionDetails xmlns:exception="http://
www.mercury.com/ppm/ws/exception">
<exception:detail>[root cause description] </exception:detail>
</exception:exceptionDetails>
    
```

Possible root cause descriptions:

Message Code	Message	Cause(s)	Possible Corrective Action
error.ws.participation.badPercent	The resource distribution group with start date {0} for resource {1} has an invalid total percentage. The sum of percentages in each group must be in the range 0 to 100	The sum of percentages in the resource distribution group exceeds 100.	Ensure the sum of percentages is between 0 and 100.

Message Code	Message	Cause(s)	Possible Corrective Action
error.ws.participation.duplicateStartDate	Duplicate start date: {0}. The start date of each resource distribution group in the same participation should be unique.	The start date of each resource distribution group in the same participation is not unique.	The start date of each resource distribution group in the same participation should be unique.
error.ws.participation.duplicateDistribution	The resource distribution group with start date {0} for the resource {1} has multiple distributions for the same resource pool. Each group may contain at most one distribution per resource pool.	The resource distribution group with start date {0} for the resource {1} has multiple distributions for the same resource pool.	Each group may contain at most one distribution per resource pool.
error.ws.participation.resourceUnavailable	The PPM start or end date of the resource {0} makes the resource unable to participate in a distribution group that starts on {1}.	Distribution group start date is outside the range of the resource start and end date in PPM Center.	Adjust either the start and end date for the resource in PPM Center or the distribution group start date.
exception.authorization	You do not have the privilege to take this action. Consult your PPM Administrator.	You do not have the required access grants for this operation.	Verify that your account has the required access grants.

createRoles

Purpose

This operation creates multiple new roles in PPM Center.

Function and Parameters

This operation returns zero or more role reference objects from PPM Center according to a list of (maximum 1,000) role beans to create roles. The user performing this operation must have the following access grant:

- Edit All Roles

Required Fields:

- Name
- Enabled

Optional Fields:

- Description
- ID

Limitations

This operation has the following limitations:

- Maximum 1,000 roles can be created in a single invocation.
- Creation of new roles using this operation occurs in a single transaction and as an atomic operation. If the creation of one role fails, the whole operation is rolled back.

Input

An array of ["Role"](#) on page 232.

Return

An array of ["RoleReference"](#) on page 225.

Java Interface

```
CreateRolesResponseDocument createRoles(CreateRolesDocument in)
```

Parameters	Description
CreateRolesDocument	Wrapper array of role bean. (RoleBean[])
CreateRolesResponseDocument	Wrapper for the roles reference array. (RoleReference[])

Java Examples

Example: create new roles.

```
public static void main(String[] args) throws Exception {
    // check parameter
    if (args.length < 1) {
        System.out.println("Usage: java
ResourceServiceClient <service URL> [<true/false>");
        System.exit(1);
    }

    System.out.println("Starting Resource Service
tests...");
    ResourceServiceClient rm = new ResourceServiceClient();
    rm.WSURL = args[0];
    if (args.length > 1 && args[1].equalsIgnoreCase("true"))
    {
        rm.DEBUG = true;
    }
}
```



```
        System.out.println("Test create role ...");
        Role role = Role.Factory.newInstance();
        role.setName("Test Role " +
System.currentTimeMillis());
        role.setDescription("New Role");
        role.setEnabled(true);
        RoleReference[] roleReferences = createRoles(new
Role[]{role});

        System.out.println("Resource Service tests complete.");
    }
    RoleReference[] createRoles(Role[] roles) throws Exception {
        ResourceServiceStub service = new
ResourceServiceStub(ctx, RESOURCE_SERVICE);
        CreateRolesDocument createRolesDoc =
CreateRolesDocument.Factory.newInstance();
        createRolesDoc.addNewCreateRoles().setRoleArray(roles);
        CreateRolesResponseDocument responseDocCreate =
service.createRoles(createRolesDoc);
        CreateRolesResponse responseCreate =
responseDocCreate.getCreateRolesResponse();
        debugPrint(responseCreate, "create response");
        return responseCreate.getRoleRefArray();
    }
}
```

Errors and Exceptions

When an error occurs on this operation, you will see a description of the root cause in the log or in the response message.

The server log file content is similar to the following:

```
Exception in thread "main" org.apache.axis2.AxisFault:
<exception:exceptionDetails xmlns:exception="http://
www.mercury.com/ppm/ws/exception">
<exception:detail>[root cause description] </exception:detail>
</exception:exceptionDetails>
```

Possible root cause descriptions:

Message Code	Message	Cause(s)	Possible Corrective Action
error.ws.maxRoles	The maximum number of roles to process in a single web service transaction	The role list has more than 1,000 roles.	Break up the list into smaller chunks and process over multiple

Message Code	Message	Cause(s)	Possible Corrective Action
	may not exceed 1000.		transactions.
error.duplicateName	A role with the name "{0}" already exists. Enter a different name.	The role already exists.	Enter a different name.
error.ws.nameRequiredRole	Name cannot be blank. Provide a name for each role.	At least one of the roles does not have a name.	Ensure all roles have a name.
error.ws.descriptionTooLong	Description cannot be longer than 650 characters.	The description of a role is more than 650 characters.	Reduce length of description.
error.ws.nameTooLong	Name cannot be longer than 260 characters.	The name of a role is great than 260.	Reduce length of name.

createSkills

Purpose

This operation creates multiple new skills in PPM Center.

Function and Parameters

This operation returns zero or more skill reference objects from PPM Center according to a specified list of (maximum 1,000) skill beans to create skills. The user performing this operation must have the following access grant:

- Edit All Skills

Required Fields:

- Name
- Enabled

Optional Fields:

- Description
- ID
- category

Limitations

This operation has the following limitations:

- Maximum 1,000 skills can be created in a single invocation.
- Creation of skills using this operation occur in one single transaction and as an atomic operation. If the creation of one skill fails, the whole operation is rolled back.

Input

An array of ["Skill" on page 233](#).

Return

An array of ["SkillReference" on page 226](#).

Java Interface

```
CreateSkillsResponseDocument createSkills(CreateSkillsDocument in)
```

Parameters	Description
CreateSkillsDocument	Wrapper array of skill bean. (SkillBean[])
CreateSkillsResponseDocument	Wrapper for the skill reference array. (SkillReference [])

Java Examples

Example: create new skills.

```
public static void main(String[] args) throws Exception {
    // check parameter
    if (args.length < 1) {
        System.out.println("Usage: java
ResourceServiceClient <service URL> [<true/false>");
        System.exit(1);
    }

    System.out.println("Starting Resource Service
tests...");
    ResourceServiceClient rm = new ResourceServiceClient();
    rm.WSURL = args[0];
    if (args.length > 1 && args[1].equalsIgnoreCase("true"))
    {
        rm.DEBUG = true;
    }

    System.out.println("Test create skill ...");

    Skill skill = Skill.Factory.newInstance();
    skill.setName("Test Skill " +
System.currentTimeMillis());
    skill.setDescription("Test Skill");
    skill.setEnabled(true);
    SkillReference[] skillReferences = createSkills(new
Skill[] {skill});

    System.out.println("Resource Service tests complete.");
}

SkillReference[] createSkills(Skill[] skills) throws Exception
{
    ResourceServiceStub service = new ResourceServiceStub(ctx,
RESOURCE_SERVICE);
    CreateSkillsDocument createSkillsDoc =
CreateSkillsDocument.Factory.newInstance();
    createSkillsDoc.addNewCreateSkills().setSkillArray(skills);
    CreateSkillsResponseDocument responseDocCreate =
service.createSkills(createSkillsDoc);
    CreateSkillsResponse responseCreate =
responseDocCreate.getCreateSkillsResponse();
    debugPrint(responseCreate, "create response");
    return responseCreate.getSkillRefArray();
}
```

Errors and Exceptions

When an error occurs with this operation, you will see a description of the root cause in the log or in the response message.

The server log file content is similar to the following:

```
Exception in thread "main" org.apache.axis2.AxisFault:
<exception:exceptionDetails xmlns:exception="http://
www.mercury.com/ppm/ws/exception">
<exception:detail>[root cause description] </exception:detail>
</exception:exceptionDetails>
```

Possible root cause descriptions:

Message Code	Message	Cause(s)	Possible Corrective Action
error.ws.maxSkills	The maximum number of skills to process in a single web service transaction may not exceed 1000.	The skill list has more than 1,000 skills.	Break up the list into smaller chunks and process over multiple transactions.
error.duplicateName	A skill with the name "{0}" already exists. Enter a different name.	The skill of the name is duplicated with an existing one.	Enter a different name.
error.ws.nameRequiredSkill	Name cannot be blank. Provide a name for each skill.	At least one of the skills does not have a name.	Ensure all skills have a name.
error.ws.descriptionTooLong	Description cannot be longer than 650 characters.	The description of a skill is more than 650 characters.	Reduce length of description.
error.ws.nameTooLong	Name cannot be longer than 260 characters.	The name of a skill is great than 260.	Reduce length of name

Chapter 8: Resource Management (Staffing Profile) Web Services

- ["Overview" below](#)
- ["References" below](#)
- ["Data Types" on the next page](#)
- ["Operations" on page 289](#)

Overview

The Web services for staffing profile provides operations to create and edit staffing profile data from external applications. Using the Staffing Profile Web service, you are able to do the following:

- Create new staffing profiles in PPM Center
- Update staffing profile header information
- Read staffing profile header information
- Read staffing profile Access Control List (ACL)
- Update staffing profile ACL
- Add positions and allocations to staffing profiles
- Read positions and allocations information
- Update positions and allocations information
- Delete positions and allocations information

Note: For more information about staffing profile terms and concepts, see the *Resource Management User's Guide*.

References

Data types definition:

`webservice_toolkit\java\conf\xsd\StaffingProfile.xsd`

Operations definition:

`webservice_toolkit\java\conf\wsdl\StaffingProfileService.wsdl`

Java sample code:

`webservice_toolkit\java\client\src\examples\rm\StaffingProfileServiceClient.java`

Data Types

HP Resource Management Web Services for staffing profile includes the following data types:

- ["StaffingProfileHeaderType" on the next page](#)
- ["ParentEntityType" on page 273](#)
- ["ParentType" on page 273](#)
- ["ParentIdentifier" on page 274](#)
- ["StaffingProfileACLType" on page 274](#)
- ["Entry" on page 274](#)
- ["positionIdentifierType" on page 275](#)
- ["positionLineType" on page 275](#)
- ["effortContourType" on page 277](#)
- ["effortCellType" on page 277](#)
- ["effortSimpleType" on page 278](#)
- ["assignmentType" on page 278](#)
- ["promisedAssignmentType" on page 278](#)
- ["skillProficiencyType" on page 279](#)
- ["userDataType" on page 279](#)
- ["PositionUserDataInfo" on page 281](#)
- ["resultObjectType" on page 288](#)

StaffingProfileHeaderType

This data type specifies a staffing profile header.

Property	Type	Description	Required	Default
Id	Integer	Staffing profile ID	No	N/A
Name	String	Staffing profile name	No	N/A
Description	String	Staffing profile description	No	N/A
ParentEntityBean	"ParentEntityType" on the next page	Parent entity information	No	N/A
managerUserName	String[]	Name list of staffing profile managers	No	N/A
Status	String	Staffing profile status Valid status are: <ul style="list-style-type: none"> • In Planning • Active • Completed • Cancelled • Lock down 	No	N/A
PlannedStartPeriodFullName	String	Staffing profile start period	No	N/A
PlannedFinishPeriodFullName	String	Staffing profile end period	No	N/A
regionName	String	Staffing profile Region	No	N/A
workLoadCategory	String	Staffing profile workload category	No	N/A
DefaultResourcePool	String	Staffing profile default resource pool	No	N/A
showAsWorkLoad	Boolean	Set to true if the	No	Yes

Property	Type	Description	Required	Default
		staffing profile shows as workload		
userData	"userDataType" on page 279	User defined data	No	N/A

ParentEntityType

This data type specifies a staffing profile parent entity. See "[ParentType](#)" below and "[ParentIdentifier](#)" on the next page.

Property	Type	Description	Required	Default
parentType	"ParentType" below	Parent type	No	N/A
parentIdentifier	"ParentIdentifier" on the next page	Parent identifier	No	N/A

ParentType

This data type defines the parent type.

Property	Type	Description	Required	Default
parentType	Enumeration	Parent type of staffing profile. Enumeration of the following types: <ul style="list-style-type: none"> • Proposal • Project • Asset • Organization Unit • Freestanding 	No	N/A

ParentIdentifier

This data type defines the parent identifier.

Property	Type	Description	Required	Default
Id	Integer	Parent entity ID	No	N/A
Name	String	Parent entity name	No	N/A

StaffingProfileACLType

This data type specifies the access control list (ACL) of a staffing profile. See ["Entry" below](#).

Property	Type	Description	Required	Default
Entry	"Entry" below	List of users with the access grant	No	N/A

Entry

This data type describes the access grants of users on the ACL of a staffing profile.

Property	Type	Description	Required	Default
Username	String	User name	Yes	N/A
editHeader	Boolean	Set to true if user can edit header	Yes	No
editPositions	Boolean	Set to true if user can edit position	Yes	No
editAssignmentActuals	Boolean	Set to true if user can edit assignments	Yes	No
editSecurity	Boolean	Set to true if user can edit security	Yes	No

positionIdentifierType

This data type defines a position identifier.

Property	Type	Description	Required	Default
id	Integer	Position ID	Yes	N/A
periodType	Enumeration	Period type Valid types are: <ul style="list-style-type: none"> • week • month • quarter • year 	Yes	N/A
effortUnit	Enumeration	Effort unit Valid items are: <ul style="list-style-type: none"> • FTE • hours • person days 	Yes	N/A

positionLineType

This data type defines a position line.

Property	Type	Description	Required	Default
id	Integer	Position ID	No	N/A
label	String	Position name	Yes	N/A
role	String	Role name	Yes	N/A
resourceType	String	Resource type	No	N/A

Property	Type	Description	Required	Default
resourcePool	String	Resource pool name	No	N/A
staffingProfileId	Integer	Staffing profile ID	Yes	N/A
comment	String	Position comment	No	N/A
responseNeededBy	Date	Response needed by date	No	N/A
status	String	Position status	No	N/A
laborCategory	String	Labor category	No	N/A
expenseType	String	Expense type that you choose Valid types are: <ul style="list-style-type: none"> • Capex • Opex • Split between capital and operating cost 	No	N/A
capitalRatio	Integer	Ratio between capital cost and operating cost This property is ignored when expenseType is set to one of the following: <ul style="list-style-type: none"> • Capex • Opex 	No	N/A
forecast	"effortContourType" on the next page	Forecast information	No	N/A
forecastSimple	"effortSimpleType" on page 278	Forecast information	No	N/A
allocations	"assignmentType" on page 278	Allocation information	No	N/A
promisedAllocations	"promisedAssignmentType" on page 278	Promised (unnamed) allocation information		
skillSet	"skillProficiencyType" on page 279	Skill set information	No	N/A
userData	"userDataType" on page 279	User defined data	No	N/A

effortContourType

This data type specifies the effort contour for a position line.

Property	Type	Description	Required	Default
timeBuckets	" effortCellType " below	Time bucket information	Yes	N/A
periodType	Enumeration	Period type Valid types are: <ul style="list-style-type: none"> • week • month • quarter • year 	Yes	N/A
effortUnit	Enumeration	Effort unit Valid values are: <ul style="list-style-type: none"> • FTE • hours • person days 	Yes	N/A

effortCellType

This data type specifies the effort cell for a position.

Property	Type	Description	Required	Default
value	Decimal	Effort assigned to position	Yes	N/A
startDate	Date	Start date of position	Yes	N/A

effortSimpleType

This data type specifies the effort for a position.

Property	Type	Description	Required	Default
startDate	Date	Start date of the effort	Yes	N/A
finishDate	Date	Finish date of the effort	Yes	N/A
effortUnit	Enumeration	Effort unit Valid value is: <ul style="list-style-type: none"> FTE 	Yes	FTE

assignmentType

This data type describes an assignment for a position.

Property	Type	Description	Required	Default
id	Integer	Assignment ID	No	N/A
resourceUsername	String	User name of resource	Yes	N/A
status	String	Status of resource assignment	Yes	N/A
allocation	"effortContourType" on the previous page	Allocation information	No	N/A

promisedAssignmentType

This data type describes a promised (unnamed) assignment for a position.

Property	Type	Description	Required	Default
id	Integer	Assignment ID	No	N/A
resourceUsername	String	User name of resource	Yes	N/A
status	String	Status of resource assignment	Yes	N/A
allocation	"effortContourType" on page 277	Allocation information	No	N/A

skillProficiencyType

This data type specifies a skill together with the proficiency.

Property	Type	Description	Required	Default
skill	String	Skill name	Yes	N/A
proficiency	String	Skill level	Yes	N/A

userDataType

This data type is used for user defined data.

Property	Type	Description	Required	Default
userData1	String	User defined data	No	N/A
userData2	String	User defined data	No	N/A
userData3	String	User defined data	No	N/A
userData4	String	User defined data	No	N/A
userData5	String	User defined data	No	N/A
userData6	String	User defined data	No	N/A
userData7	String	User defined data	No	N/A

Property	Type	Description	Required	Default
userData8	String	User defined data	No	N/A
userData9	String	User defined data	No	N/A
userData10	String	User defined data	No	N/A
userData11	String	User defined data	No	N/A
userData12	String	User defined data	No	N/A
userData13	String	User defined data	No	N/A
userData14	String	User defined data	No	N/A
userData15	String	User defined data	No	N/A
userData16	String	User defined data	No	N/A
userData17	String	User defined data	No	N/A
userData18	String	User defined data	No	N/A
userData19	String	User defined data	No	N/A
userData20	String	User defined data	No	N/A
visUserData1	String	User defined data	No	N/A
visUserData2	String	User defined data	No	N/A
visUserData3	String	User defined data	No	N/A
visUserData4	String	User defined data	No	N/A
visUserData5	String	User defined data	No	N/A
visUserData6	String	User defined data	No	N/A
visUserData7	String	User defined data	No	N/A
visUserData8	String	User defined data	No	N/A
visUserData9	String	User defined data	No	N/A
visUserData10	String	User defined data	No	N/A
visUserData11	String	User defined data	No	N/A
visUserData12	String	User defined data	No	N/A
visUserData13	String	User defined data	No	N/A
visUserData14	String	User defined data	No	N/A

Property	Type	Description	Required	Default
visUserData15	String	User defined data	No	N/A
visUserData16	String	User defined data	No	N/A
visUserData17	String	User defined data	No	N/A
visUserData18	String	User defined data	No	N/A
visUserData19	String	User defined data	No	N/A
visUserData20	String	User defined data	No	N/A

PositionUserDataInfo

This data type describes the user data fields for a position.

Property	Type	Description	Required	Default
userData1	String	Code to identify the user data field.	No	N/A
userData2	String	Code to identify the user data field.	No	N/A
userData3	String	Code to identify the user data field.	No	N/A
userData4	String	Code to identify the user data field.	No	N/A
userData5	String	Code to identify the user data field.	No	N/A
userData6	String	Code to identify the user data field.	No	N/A
userData7	String	Code to identify the user data field.	No	N/A
userData8	String	Code to identify the user data field.	No	N/A
userData9	String	Code to identify the user data field.	No	N/A
userData10	String	Code to identify the user data field.	No	N/A
userData11	String	Code to identify the user data field.	No	N/A
userData12	String	Code to identify the user data field.	No	N/A
userData13	String	Code to identify the user data field.	No	N/A
userData14	String	Code to identify the user data field.	No	N/A
userData15	String	Code to identify the user data field.	No	N/A

Property	Type	Description	Required	Default
userData16	String	Code to identify the user data field.	No	N/A
userData17	String	Code to identify the user data field.	No	N/A
userData18	String	Code to identify the user data field.	No	N/A
userData19	String	Code to identify the user data field.	No	N/A
userData20	String	Code to identify the user data field.	No	N/A
userData21	String	Code to identify the user data field.	No	N/A
userData22	String	Code to identify the user data field.	No	N/A
userData23	String	Code to identify the user data field.	No	N/A
userData24	String	Code to identify the user data field.	No	N/A
userData25	String	Code to identify the user data field.	No	N/A
userData26	String	Code to identify the user data field.	No	N/A
userData27	String	Code to identify the user data field.	No	N/A
userData28	String	Code to identify the user data field.	No	N/A
userData29	String	Code to identify the user data field.	No	N/A
userData30	String	Code to identify the user data field.	No	N/A
userData31	String	Code to identify the user data field.	No	N/A
userData32	String	Code to identify the user data field.	No	N/A
userData33	String	Code to identify the user data field.	No	N/A
userData34	String	Code to identify the user data field.	No	N/A
userData35	String	Code to identify the user data field.	No	N/A
userData36	String	Code to identify the user data field.	No	N/A
userData37	String	Code to identify the user data field.	No	N/A
userData38	String	Code to identify the user data field.	No	N/A
userData39	String	Code to identify the user data field.	No	N/A
userData40	String	Code to identify the user data field.	No	N/A
userData41	String	Code to identify the user data field.	No	N/A
userData42	String	Code to identify the user data field.	No	N/A

Property	Type	Description	Required	Default
userData43	String	Code to identify the user data field.	No	N/A
userData44	String	Code to identify the user data field.	No	N/A
userData45	String	Code to identify the user data field.	No	N/A
userData46	String	Code to identify the user data field.	No	N/A
userData47	String	Code to identify the user data field.	No	N/A
userData48	String	Code to identify the user data field.	No	N/A
userData49	String	Code to identify the user data field.	No	N/A
userData50	String	Code to identify the user data field.	No	N/A
userData51	String	Code to identify the user data field.	No	N/A
userData52	String	Code to identify the user data field.	No	N/A
userData53	String	Code to identify the user data field.	No	N/A
userData54	String	Code to identify the user data field.	No	N/A
userData55	String	Code to identify the user data field.	No	N/A
userData56	String	Code to identify the user data field.	No	N/A
userData57	String	Code to identify the user data field.	No	N/A
userData58	String	Code to identify the user data field.	No	N/A
userData59	String	Code to identify the user data field.	No	N/A
userData60	String	Code to identify the user data field.	No	N/A
userData61	String	Code to identify the user data field.	No	N/A
userData62	String	Code to identify the user data field.	No	N/A
userData63	String	Code to identify the user data field.	No	N/A
userData64	String	Code to identify the user data field.	No	N/A
userData65	String	Code to identify the user data field.	No	N/A
userData66	String	Code to identify the user data field.	No	N/A
userData67	String	Code to identify the user data field.	No	N/A
userData68	String	Code to identify the user data field.	No	N/A
userData69	String	Code to identify the user data field.	No	N/A

Property	Type	Description	Required	Default
userData70	String	Code to identify the user data field.	No	N/A
userData71	String	Code to identify the user data field.	No	N/A
userData72	String	Code to identify the user data field.	No	N/A
userData73	String	Code to identify the user data field.	No	N/A
userData74	String	Code to identify the user data field.	No	N/A
userData75	String	Code to identify the user data field.	No	N/A
userData76	String	Code to identify the user data field.	No	N/A
userData77	String	Code to identify the user data field.	No	N/A
userData78	String	Code to identify the user data field.	No	N/A
userData79	String	Code to identify the user data field.	No	N/A
userData80	String	Code to identify the user data field.	No	N/A
userData81	String	Code to identify the user data field.	No	N/A
userData82	String	Code to identify the user data field.	No	N/A
userData83	String	Code to identify the user data field.	No	N/A
userData84	String	Code to identify the user data field.	No	N/A
userData85	String	Code to identify the user data field.	No	N/A
userData86	String	Code to identify the user data field.	No	N/A
userData87	String	Code to identify the user data field.	No	N/A
userData88	String	Code to identify the user data field.	No	N/A
userData89	String	Code to identify the user data field.	No	N/A
userData90	String	Code to identify the user data field.	No	N/A
userData91	String	Code to identify the user data field.	No	N/A
userData92	String	Code to identify the user data field.	No	N/A
userData93	String	Code to identify the user data field.	No	N/A
userData94	String	Code to identify the user data field.	No	N/A
userData95	String	Code to identify the user data field.	No	N/A
userData96	String	Code to identify the user data field.	No	N/A

Property	Type	Description	Required	Default
userData97	String	Code to identify the user data field.	No	N/A
userData98	String	Code to identify the user data field.	No	N/A
userData99	String	Code to identify the user data field.	No	N/A
userData100	String	Code to identify the user data field.	No	N/A
visUserData1	String	User data field.	No	N/A
visUserData2	String	User data field.	No	N/A
visUserData3	String	User data field.	No	N/A
visUserData4	String	User data field.	No	N/A
visUserData5	String	User data field.	No	N/A
visUserData6	String	User data field.	No	N/A
visUserData7	String	User data field.	No	N/A
visUserData8	String	User data field.	No	N/A
visUserData9	String	User data field.	No	N/A
visUserData10	String	User data field.	No	N/A
visUserData11	String	User data field.	No	N/A
visUserData12	String	User data field.	No	N/A
visUserData13	String	User data field.	No	N/A
visUserData14	String	User data field.	No	N/A
visUserData15	String	User data field.	No	N/A
visUserData16	String	User data field.	No	N/A
visUserData17	String	User data field.	No	N/A
visUserData18	String	User data field.	No	N/A
visUserData19	String	User data field.	No	N/A
visUserData20	String	User data field.	No	N/A
visUserData21	String	User data field.	No	N/A
visUserData22	String	User data field.	No	N/A
visUserData23	String	User data field.	No	N/A

Property	Type	Description	Required	Default
visUserData24	String	User data field.	No	N/A
visUserData25	String	User data field.	No	N/A
visUserData26	String	User data field.	No	N/A
visUserData27	String	User data field.	No	N/A
visUserData28	String	User data field.	No	N/A
visUserData29	String	User data field.	No	N/A
visUserData30	String	User data field.	No	N/A
visUserData31	String	User data field.	No	N/A
visUserData32	String	User data field.	No	N/A
visUserData33	String	User data field.	No	N/A
visUserData34	String	User data field.	No	N/A
visUserData35	String	User data field.	No	N/A
visUserData36	String	User data field.	No	N/A
visUserData37	String	User data field.	No	N/A
visUserData38	String	User data field.	No	N/A
visUserData39	String	User data field.	No	N/A
visUserData40	String	User data field.	No	N/A
visUserData41	String	User data field.	No	N/A
visUserData42	String	User data field.	No	N/A
visUserData43	String	User data field.	No	N/A
visUserData44	String	User data field.	No	N/A
visUserData45	String	User data field.	No	N/A
visUserData46	String	User data field.	No	N/A
visUserData47	String	User data field.	No	N/A
visUserData48	String	User data field.	No	N/A
visUserData49	String	User data field.	No	N/A
visUserData50	String	User data field.	No	N/A

Property	Type	Description	Required	Default
visUserData51	String	User data field.	No	N/A
visUserData52	String	User data field.	No	N/A
visUserData53	String	User data field.	No	N/A
visUserData54	String	User data field.	No	N/A
visUserData55	String	User data field.	No	N/A
visUserData56	String	User data field.	No	N/A
visUserData57	String	User data field.	No	N/A
visUserData58	String	User data field.	No	N/A
visUserData59	String	User data field.	No	N/A
visUserData60	String	User data field.	No	N/A
visUserData61	String	User data field.	No	N/A
visUserData62	String	User data field.	No	N/A
visUserData63	String	User data field.	No	N/A
visUserData64	String	User data field.	No	N/A
visUserData65	String	User data field.	No	N/A
visUserData66	String	User data field.	No	N/A
visUserData67	String	User data field.	No	N/A
visUserData68	String	User data field.	No	N/A
visUserData69	String	User data field.	No	N/A
visUserData70	String	User data field.	No	N/A
visUserData71	String	User data field.	No	N/A
visUserData72	String	User data field.	No	N/A
visUserData73	String	User data field.	No	N/A
visUserData74	String	User data field.	No	N/A
visUserData75	String	User data field.	No	N/A
visUserData76	String	User data field.	No	N/A
visUserData77	String	User data field.	No	N/A

Property	Type	Description	Required	Default
visUserData78	String	User data field.	No	N/A
visUserData79	String	User data field.	No	N/A
visUserData80	String	User data field.	No	N/A
visUserData81	String	User data field.	No	N/A
visUserData82	String	User data field.	No	N/A
visUserData83	String	User data field.	No	N/A
visUserData84	String	User data field.	No	N/A
visUserData85	String	User data field.	No	N/A
visUserData86	String	User data field.	No	N/A
visUserData87	String	User data field.	No	N/A
visUserData88	String	User data field.	No	N/A
visUserData89	String	User data field.	No	N/A
visUserData90	String	User data field.	No	N/A
visUserData91	String	User data field.	No	N/A
visUserData92	String	User data field.	No	N/A
visUserData93	String	User data field.	No	N/A
visUserData94	String	User data field.	No	N/A
visUserData95	String	User data field.	No	N/A
visUserData96	String	User data field.	No	N/A
visUserData97	String	User data field.	No	N/A
visUserData98	String	User data field.	No	N/A
visUserData99	String	User data field.	No	N/A
visUserData100	String	User data field.	No	N/A

responseObjectType

This data type describes the result for the following operations:

- ["addPositionLines" on page 315](#)
- ["updatePositionLines" on page 319](#)
- ["deletePositionLines" on page 322](#)

Property	Type	Description	Required	Default
id	Integer	Position ID	Yes	N/A
warning	String	Warning message	Yes	N/A

Operations

The following operations are included in Resource Management Web services for staffing profile:

- ["createStaffingProfileHeader" on the next page](#)
- ["readStaffingProfileHeader" on page 297](#)
- ["updateStaffingProfileHeader" on page 300](#)
- ["readStaffingProfileACL" on page 307](#)
- ["updateStaffingProfileACL" on page 310](#)
- ["readPositionLines" on page 313](#)
- ["addPositionLines" on page 315](#)
- ["updatePositionLines" on page 319](#)
- ["deletePositionLines" on page 322](#)
- ["getPositionIdsforStaffingProfile" on page 325](#)
- ["clearStaffingProfileForecastAndAssignmet " on page 327](#)

createStaffingProfileHeader

Purpose

This operation creates a staffing profile header in PPM Center.

Function

This operation enables external applications to create a staffing profile header in PPM Center. The operation returns the ID of the staffing profile created in PPM Center.

To perform this operation, the user must meet one of the following access grant requirements:

- Have both the create staffing profile access grant and the edit staffing profile access grant
- Have the edit all staffing profiles access grant

Apart from the access grant requirement, the user must also meet one of the following:

- Be the manager of the staffing profile
- In the staffing profile ACL list
- Be the manager of the project associated with the staffing profile

Related Information

The following data types:

- ["StaffingProfileHeaderType" on page 272](#)
- ["ParentEntityType" on page 273](#)
- ["ParentType" on page 273](#)
- ["ParentIdentifier" on page 274](#)

Input

A ["StaffingProfileHeaderType" on page 272](#) object

Return

The staffing profile ID will be returned if the staffing profile is created successfully.

Java Interface

```
public long  
createStaffingProfileHeader(com.mercury.itg.ws.rm.vo.StaffingProfileHeaderWSBean  
inputSP)
```

Parameters	Description
inputSP	Wrapper of the StaffingProfileHeaderType object

Java Examples

Example: creates a staffing profile header

```
private Long createStaffingProfileHeader(String serviceURL)  
throws Exception{  
    Long id = null;  
    StaffingProfileHeaderType spHeaderType =  
StaffingProfileHeaderType.Factory.newInstance();  
    generateTestSPHeaderWSBean(spHeaderType);  
    StaffingProfileServiceStub stub = new  
StaffingProfileServiceStub(ctx, serviceURL);  
    CreateStaffingProfileHeaderDocument inDoc =  
CreateStaffingProfileHeaderDocument.Factory.newInstance();  
    CreateStaffingProfileHeaderDocument.CreateStaffingProfileHeader  
createSPHeader = inDoc.addNewCreateStaffingProfileHeader();  
    createSPHeader.setStaffingProfileHeaderWSBean(spHeaderType);  
    // Invoke web service  
    CreateStaffingProfileHeaderResponseDocument outDoc =
```

```

stub.createStaffingProfileHeader(inDoc);
    id =
outDoc.getCreateStaffingProfileHeaderResponse().getStaffingProfileId();

    return id;
}

```

Errors and Exceptions

Possible root cause descriptions:

Message Code	Message	Cause(s)	Possible Corrective Action
exception.authorization	You do not have the privilege to take this action. Please consult your PPM Administrator	You do not have the required access grants for this operation	Verify that your account has the required access grants
rm.ws.exception.staffingProfileNoName	Staffing profile name can not be null	Staffing profile name is not specified	Specify a staffing profile
rm.ws.exception.startPeriodOutOfRange	Start period is out of range	The start period is out of the period range stored in the database	Make sure the start period exists in the PPM_FISCAL_PERIODS view
rm.ws.exception.finishPeriodOutOfRange	Finish period is out of range	The finish period is out of the period range	Make sure the finish period exists in the PPM_FISCAL_

Message Code	Message	Cause(s)	Possible Corrective Action
		stored in the database	PERIODS view
rm.ws.exception.finishBeforeStart	Finish date is before start date	The finish date is before the start date	Make sure the finish data is after the start date
rm.ws.exception.Region	Region is not found	The region is not found	Make sure the region exists before calling the Web service operation
rm.ws.exception.duplicateStaffingProfileName	The name of the staffing profile you are trying to create already exists. Please try a different name	The name of the staffing profile already exists.	Enter a different name
rm.ws.exception.missingStaffingProfileManager	Missing staffing profile manager	The user doesn't enter the manager of staffing profile	Specify the staffing profile manager
rm.ws.exception.staffingProfileManager	Staffing profile manager is not found	The staffing profile manager is not found	Check if the staffing profile manager exists
rm.ws.exception.resourcePool	Resource pool is not found	The resource pool is not found	Check if the resource pool exists

Message Code	Message	Cause(s)	Possible Corrective Action
rm.ws.exception.missingStaffingProfileType	Missing staffing profile type	The staffing profile type is not specified	Specify a staffing profile type
rm.ws.exception.notValidParentType	Not a valid staffing profile parent type	The staffing profile parent type is invalid	Make sure to enter a valid staffing profile parent type. The possible values are: <ul style="list-style-type: none"> • Freestanding • Organization Unit. • Asset • Proposal • Project
rm.ws.exception.notValidParentIdentifier	Missing parent entity name or id	The name or ID of parent entity is not specified	Specify a name or ID for parent entity.
rm.ws.exception.staffingProfileWorkLoadCategory	Staffing profile work load category is not found	Staffing profile work load category is not found	Enter a valid staffing profile work load category
rm.ws.exception.missingWorkLoadCategory	Missing staffing profile work load category	The staffing profile is NOT linked to a project and its work load category is null	Specify a staffing profile work load category

Message Code	Message	Cause(s)	Possible Corrective Action
rm.ws.exception.project	Project is not found	The project name or ID is not found	Enter a valid project name or ID
rm.ws.exception.staffingprofileNotValidProject	Not a valid linked project	The linked project is invalid	Enter a valid linked project
rm.ws.exception.staffingprofileStartNotEqualProjectStart	Staffing profile start period does not equal to linked project start period {0}	The staffing profile start period does not equal to its linked project start period	Make sure the staffing profile start period is equal to the linked project start period
rm.ws.exception.missingParentProposal	Linked proposal is not found	The linked proposal is not found	Enter a valid linked proposal
rm.ws.exception.staffingprofileNotValidProposal	Not a valid linked proposal	The linked proposal is invalid	Enter a valid linked proposal
rm.ws.exception.staffingprofileStartNotEqualProposalStart	Staffing profile start date does not equal to linked proposal start date	The staffing profile start date does not equal to the linked proposal start date	Make sure the staffing profile start date is equal to the linked proposal start date
rm.ws.exception.missingParentOrgUnit	Linked organization unit is not found	The linked organization unit is not found	Enter an existing organization unit
rm.ws.exception.staffingprofileNotValidOrgUnit	Not a valid linked organization unit	The linked organization unit is invalid	Enter a valid linked organization unit

Message Code	Message	Cause(s)	Possible Corrective Action
rm.ws.exception.missingParentAsset	Linked asset is not found	The linked asset is not found	Enter an existing linked asset
rm.ws.exception.staffingprofileNotValidAsset	Not a valid linked asset	The linked asset is invalid	Enter a valid linked asset
rm.ws.exception.unknownType	Unknown staffing profile parent type	The staffing profile parent type is unknown	Make sure to enter a valid staffing profile parent type. The possible values are <ul style="list-style-type: none"> • Freestanding • Organization Unit. • Asset • Proposal • Project
rm.ws.exception.missingStatus	Missing status	The staffing profile's status is not specified	Specify a staffing profile status
rm.ws.exception.notValidStatus	Not a valid status	The staffing profile status is invalid	Enter a valid staffing profile status
rm.ws.exception.invalidCreateBaselineStatus	Not allowed to create a baseline staffing profile	You try to create a baseline staffing profile	Specify a valid staffing profile status except the following: <ul style="list-style-type: none"> • Baseline • Approved

Message Code	Message	Cause(s)	Possible Corrective Action
rm.ws.exception.invalidCreateApprovedStatus	Not allowed to create a approved staffing profile	You try to create a approved staffing profile	Specify a valid staffing profile status except the following: <ul style="list-style-type: none"> • Baseline • Approved
rm.ws.exception.notValidBudget	Not a valid budget to synchronize with staffing profile	The budget to synchronize with staffing profile is invalid	Specify a valid budget to synchronize with staffing profile
rm.ws.exception.notSpecifyBudget	You need specify a budget name when set the staffing profile synchronize with budget	The budget name is not specified when set the staffing profile synchronize with budget	Specify a budget name

readStaffingProfileHeader

Purpose

This operation reads a staffing profile header in PPM Center.

Function

This operation enables external applications to read a staffing profile header in PPM Center identified by staffing profile ID. The operation returns the staffing profile object

To perform this operation, the user must meet the following conditions:

- Have the view staffing profile access grant
- In the Staffing profile ACL list

Related Information

The following data types:

- ["StaffingProfileHeaderType" on page 272](#)
- ["ParentEntityType" on page 273](#)
- ["ParentType" on page 273](#)
- ["ParentIdentifier" on page 274](#)

Input

The staffing profile ID

Return

A ["StaffingProfileHeaderType" on page 272](#) object will be returned if the staffing profile is created successfully.

Java Interface

```
Public com.mercury.itg.ws.rm.vo.StaffingProfileHeaderWSBean  
readStaffingProfileHeader(long staffingProfileId)
```

Parameters	Description
staffingProfileId	Staffing profile ID

Java Examples

Example: reads a staffing profile header

```
private StaffingProfileHeaderType
readStaffingProfileHeader(String serviceURL, Long id) throws
Exception{

    StaffingProfileHeaderType spHeaderType = null;
    StaffingProfileServiceStub stub = new
StaffingProfileServiceStub(ctx, serviceURL);
    ReadStaffingProfileHeaderDocument inDoc =
ReadStaffingProfileHeaderDocument.Factory.newInstance();
    ReadStaffingProfileHeaderDocument.ReadStaffingProfileHeader
readSPHeader = inDoc.addNewReadStaffingProfileHeader();
    readSPHeader.setStaffingProfileId(id);
    // Invoke web service
    ReadStaffingProfileHeaderResponseDocument outDoc =
stub.readStaffingProfileHeader(inDoc);
    spHeaderType =
outDoc.getReadStaffingProfileHeaderResponse().getReturn();

    return spHeaderType;
}
```

Errors and Exceptions

Possible root cause descriptions:

Message Code	Message	Cause(s)	Possible Corrective Action
rm.ws.exception.StaffingProfileNotExist	Staffing profile does not exist	The staffing profile cannot be found by using the staffing profile ID	Verify that the staffing profile ID exists
exception.authorization	You do not have the privilege to take this action. Please consult your PPM	You do not have the required access grants for this	Verify that your account has the required

Message Code	Message	Cause(s)	Possible Corrective Action
	Administrator	operation	access grants

updateStaffingProfileHeader

Purpose

This operation updates a staffing profile header in PPM Center.

Function

This operation updates external applications to create a staffing profile header in PPM Center. The operation returns the ID of the staffing profile created in PPM Center.

To perform this operation, the user must meet the following access grant requirement:

- Have the edit all staffing profiles access grant or the edit staffing profile access grant

Apart from the access grant requirement, the user must also meet one of the following:

- Be the manager of the staffing profile
- In the staffing profile ACL list
- Be the manager of the project associated with the staffing profile

Related Information

The following data types:

- ["StaffingProfileHeaderType" on page 272](#)
- ["ParentEntityType" on page 273](#)

- ["ParentType" on page 273](#)
- ["ParentIdentifier" on page 274](#)

Input

A ["StaffingProfileHeaderType" on page 272](#) object

Return

The staffing profile ID will be returned if the staffing profile is created successfully.

Java Interface

```
public long updateStaffingProfileHeader  
(com.mercury.itg.ws.rm.vo.StaffingProfileHeaderWSBean inputSP)
```

Parameters	Description
inputSP	Wrapper of the StaffingProfileHeaderType object

Java Examples

Example: updates a staffing profile header

```
private Long updateStaffingProfileHeader(String serviceURL,  
Long id) throws Exception{  
    StaffingProfileHeaderType spHeaderType =  
StaffingProfileHeaderType.Factory.newInstance();  
    updateTestSPHeaderWSBean(spHeaderType);  
    StaffingProfileServiceStub stub = new  
StaffingProfileServiceStub(ctx, serviceURL);  
    UpdateStaffingProfileHeaderDocument inDoc =  
UpdateStaffingProfileHeaderDocument.Factory.newInstance();  
UpdateStaffingProfileHeaderDocument.UpdateStaffingProfileHeader  
updateSPHeader = inDoc.addNewUpdateStaffingProfileHeader();  
updateSPHeader.setStaffingProfileHeaderWSBean(spHeaderType);
```

```

    // Invoke web service
    UpdateStaffingProfileHeaderResponseDocument outDoc =
stub.updateStaffingProfileHeader(inDoc);
    id =
outDoc.getUpdateStaffingProfileHeaderResponse().getStaffingProfileId();

    return id;
}

```

Errors and Exceptions

Possible root cause descriptions:

Message Code	Message	Cause(s)	Possible Corrective Action
exception.authorization	You do not have the privilege to take this action. Please consult your PPM Administrator	You do not have the required access grants for this operation	Verify that your account has the required access grants
rm.ws.exception.staffingProfileNoName	Staffing profile name can not be null	Staffing profile name is not specified	Specify a staffing profile
rm.ws.exception.startPeriodOutOfRange	Start period is out of range	The start period is out of the period range stored in the database	Make sure the start period exists in the PPM_FISCAL_PERIODS view
rm.ws.exception.finishPeriodOutOfRange	Finish period is out of range	The finish period is	Make sure the finish period exists in the

Message Code	Message	Cause(s)	Possible Corrective Action
		out of the period range stored in the database	PPM_FISCAL_PERIODS view
rm.ws.exception.finishBeforeStart	Finish date is before start date	The finish date is before the start date	Make sure the finish data is after the start date
rm.ws.exception.Region	Region is not found	The region is not found	Make sure the region exists before calling the Web service operation
rm.ws.exception.duplicateStaffingProfileName	The name of the staffing profile you are trying to create already exists. Please try a different name	The name of the staffing profile already exists.	Enter a different name
rm.ws.exception.missingStaffingProfileManager	Missing staffing profile manager	The user doesn't enter the manager of staffing profile	Specify the staffing profile manager
rm.ws.exception.staffingProfileManager	Staffing profile manager is not found	The staffing profile manager is not found	Check if the staffing profile manager exists
rm.ws.exception.resourcePool	Resource pool is not	The resource	Check if the resource pool

Message Code	Message	Cause(s)	Possible Corrective Action
	found	pool is not found	exists
rm.ws.exception.missingStaffingProfileType	Missing staffing profile type	The staffing profile type is not specified	Specify a staffing profile type
rm.ws.exception.notValidParentType	Not a valid staffing profile parent type	The staffing profile parent type is invalid	Make sure to enter a valid staffing profile parent type. The possible values are: <ul style="list-style-type: none"> • Freestanding • Organization Unit. • Asset • Proposal • Project
rm.ws.exception.notValidParentIdentifier	Missing parent entity name or id	The name or ID of parent entity is not specified	Specify a name or ID for parent entity.
rm.ws.exception.staffingProfileWorkLoadCategory	Staffing profile work load category is not found	Staffing profile work load category is not found	Enter a valid staffing profile work load category
rm.ws.exception.missingWorkLoadCategory	Missing staffing profile work load category	The staffing profile is NOT linked to a project and its work	Specify a staffing profile work load category

Message Code	Message	Cause(s)	Possible Corrective Action
		load category is null	
rm.ws.exception.project	Project is not found	The project name or ID is not found	Enter a valid project name or ID
rm.ws.exception.staffingprofileNotValidProject	Not a valid linked project	The linked project is invalid	Enter a valid linked project
rm.ws.exception.staffingprofileStartNotEqualProjectStart	Staffing profile start period does not equal to linked project start period {0}	The staffing profile start period does not equal to its linked project start period	Make sure the staffing profile start period is equal to the linked project start period
rm.ws.exception.missingParentProposal	Linked proposal is not found	The linked proposal is not found	Enter a valid linked proposal
rm.ws.exception.staffingprofileNotValidProposal	Not a valid linked proposal	The linked proposal is invalid	Enter a valid linked proposal
rm.ws.exception.staffingprofileStartNotEqualProposalStart	Staffing profile start date does not equal to linked proposal start date	The staffing profile start date does not equal to the linked proposal start date	Make sure the staffing profile start date is equal to the linked proposal start date
rm.ws.exception.missingParentOrgUnit	Linked organization unit is not found	The linked organization unit is not found	Enter an existing organization unit

Message Code	Message	Cause(s)	Possible Corrective Action
rm.ws.exception.staffingprofileNotValidOrgUnit	Not a valid linked organization unit	The linked organization unit is invalid	Enter a valid linked organization unit
rm.ws.exception.missingParentAsset	Linked asset is not found	The linked asset is not found	Enter an existing linked asset
rm.ws.exception.staffingprofileNotValidAsset	Not a valid linked asset	The linked asset is invalid	Enter a valid linked asset
rm.ws.exception.unknownType	Unknown staffing profile parent type	The staffing profile parent type is unknown	Make sure to enter a valid staffing profile parent type. The possible values are <ul style="list-style-type: none"> • Freestanding • Organization Unit. • Asset • Proposal • Project
rm.ws.exception.missingStatus	Missing status	The staffing profile's status is not specified	Specify a staffing profile status
rm.ws.exception.notValidStatus	Not a valid status	The staffing profile status is invalid	Enter a valid staffing profile status
rm.ws.exception.invalidCreateBaselineStatus	Not allowed to create a baseline	You try to create a baseline	Specify a valid staffing profile status except

Message Code	Message	Cause(s)	Possible Corrective Action
	staffing profile	staffing profile	the following: <ul style="list-style-type: none"> • Baseline • Approved
rm.ws.exception.invalidCreateApprovedStatus	Not allowed to create a approved staffing profile	You try to create a approved staffing profile	Specify a valid staffing profile status except the following: <ul style="list-style-type: none"> • Baseline • Approved
rm.ws.exception.notValidBudget	Not a valid budget to synchronize with staffing profile	The budget to synchronize with staffing profile is invalid	Specify a valid budget to synchronize with staffing profile
rm.ws.exception.notSpecifyBudget	You need specify a budget name when set the staffing profile synchronize with budget	The budget name is not specified when set the staffing profile synchronize with budget	Specify a budget name

readStaffingProfileACL

Purpose

This operation reads the ACL of a staffing profile header in PPM Center.

Function

This operation enables external applications to read the ACL for a staffing profile identified by staffing profile ID. On success, the operation returns the ACL of the staffing profile.

To perform this operation, the user must meet the following conditions:

- Have the view staffing profile access grant
- In the staffing profile ACL list

Related information

The following data types:

- `StaffingProfileACLType`
- `Entry`

Input

The staffing profile ID.

Return

A "[StaffingProfileACLType](#)" on page 274 object.

Java Interface

```
Public com.mercury.itg.ws.rm.vo.StaffingProfileSecurableWSBean  
readStaffingProfileACL(long id)
```

Parameters	Description
id	Staffing profile ID

Java Examples

Example: reads a staffing profile's ACL

```
private StaffingProfileACLType readStaffingProfileACL(String
serviceURL, Long id) throws Exception{

    StaffingProfileACLType spACL = null;

    StaffingProfileServiceStub stub = new
StaffingProfileServiceStub(ctx, serviceURL);
    ReadStaffingProfileACLDocument inDoc =
ReadStaffingProfileACLDocument.Factory.newInstance();
    ReadStaffingProfileACLDocument.ReadStaffingProfileACL
readSPACL = inDoc.addNewReadStaffingProfileACL();
    readSPACL.setId(id);

    // Invoke web service
    ReadStaffingProfileACLResponseDocument outDoc =
stub.readStaffingProfileACL(inDoc);
    spACL =
outDoc.getReadStaffingProfileACLResponse().getReturn();

    return spACL;
}
```

Errors and Exceptions

Possible root cause descriptions:

Message Code	Message	Cause(s)	Possible Corrective Action
rm.ws.exception.StaffingProfileNotExist	Staffing profile does not exist	The staffing profile cannot be found by using the staffing profile id	Specify a valid staffing profile ID
exception.authorization	You do not have the privilege to take this action. Please consult your PPM	You do not have the required access grants for this	Verify that your account has the required

Message Code	Message	Cause(s)	Possible Corrective Action
	Administrator	operation	access grants

updateStaffingProfileACL

Purpose

This operation updates the ACL of a staffing profile header in PPM Center.

Function

This operation enables external applications to update the ACL for a staffing profile identified by staffing profile ID. On success, the operation returns the ACL of the staffing profile.

To perform this operation, the user must meet the following conditions:

- Have the edit staffing profile access grant or the edit all staffing profiles access grant
- In the staffing profile ACL list

Related information

The following data types:

- StaffingProfileACLType
- Entry

Input

This operation requires the following inputs:

- A ["StaffingProfileACLType" on page 274](#) object.
- The staffing profile ID

Return

The staffing profile ID

Java Interface

```
public long updateStaffingProfileACL(long id,
com.mercury.itg.ws.rm.vo.StaffingProfileSecurableWSBean inputACL)
```

Parameters	Description
id	Staffing profile ID
inputACL	Wrapper of the StaffingProfileACLType object array

Java Examples

Example: updates a staffing profile's ACL

```
private Long updateStaffingProfileACL(String serviceURL, Long
id) throws Exception{
    StaffingProfileACLType spACL =
StaffingProfileACLType.Factory.newInstance();
    updateTestSPHeaderACL(spACL);
    StaffingProfileServiceStub stub = new
StaffingProfileServiceStub(ctx, serviceURL);
    UpdateStaffingProfileACLDocument inDoc =
UpdateStaffingProfileACLDocument.Factory.newInstance();
    UpdateStaffingProfileACLDocument.UpdateStaffingProfileACL
updateSPACL = inDoc.addNewUpdateStaffingProfileACL();
    updateSPACL.setInputACL(spACL);
    updateSPACL.setId(id);

    // Invoke web service
    UpdateStaffingProfileACLResponseDocument outDoc =
stub.updateStaffingProfileACL(inDoc);
    id =
```

```

outDoc.getUpdateStaffingProfileACLResponse().getStaffingProfile
Id();

    return id;
}

```

Errors and Exceptions

Possible root cause descriptions:

Message Code	Message	Cause(s)	Possible Corrective Action
exception.authorization	You do not have the privilege to take this action. Please consult your PPM Administrator	You do not have the required access grants for this operation	Verify that your account has the required access grants
rm.ws.exception.userName	User {0} does not exist	The user does not exist in PPM Center	Verify that the user exists in PPM Center
rm.ws.exception.canNotEditManager	You are not allowed to edit staffing profile manager {0}	You are trying to edit the access grants of a staffing profile manager	Remove the access grants of the staffing profile manager from the ACL list you input
rm.ws.exception.duplicateACLUsers	You added duplicate entries for User, please remove the duplicate user {0}	There are duplicate entries for the user in the inputted ACL list	Remove the duplicate entries for the user from the ACL list you input
rm.ws.exception.disabledResource	You are not allowed to add disabled user {0}	You are trying to add a disabled user	Remove the disabled user from the ACL list you input

readPositionLines

Purpose

This operation reads position lines in PPM Center.

Function

This operation enables external applications to read the position lines of the staffing profile identified by position ID in PPM Center. The operation returns the positionLineType object.

To perform this operation, the user must meet the following conditions:

- View staffing profile access grant
- In Staffing profile ACL list

Related information

The following data types:

- ["positionLineType" on page 275](#)
- ["effortContourType" on page 277](#)
- ["effortCellType" on page 277](#)
- ["assignmentType" on page 278](#)
- ["promisedAssignmentType" on page 278](#)
- ["skillProficiencyType" on page 279](#)
- ["userDataType" on page 279](#)
- ["resultObjectType" on page 288](#)

Input

An array of ["positionLineType" on page 275](#) object.

Return

An array of ["responseObjectType" on page 288](#) object

Java Interface

```
public OperationResultWSBean[]  
readPositionLines(PositionLineWSBean[] ids)
```

Parameters	Description
ids	Wrapper of the positionLineType object array

Java Examples

Example: reads position lines of staffing profile

```
private PositionLineType[] readPositionLines(String serviceURL,  
PositionIdentifierType[] ids) throws Exception{  
    PositionLineType[] plType = null;  
    StaffingProfileServiceStub stub = new  
StaffingProfileServiceStub(ctx, serviceURL);  
    ReadPositionLinesDocument inDoc =  
ReadPositionLinesDocument.Factory.newInstance();  
    ReadPositionLinesDocument.ReadPositionLines readPL =  
inDoc.addNewReadPositionLines();  
    readPL.setPositionIdArray(ids);  
  
    // Invoke web service  
    ReadPositionLinesResponseDocument outDoc =  
stub.readPositionLines(inDoc);  
    plType =  
outDoc.getReadPositionLinesResponse().getPositionLineArray();  
  
    return plType;
```

Errors and Exceptions

Possible root cause descriptions:

Message Code	Message	Cause(s)	Possible Corrective Action
exception.exceededThreshold	The web service call exceeds the threshold limit for processing. Please batch your updates.	The number of position lines in one Web service call is limited to 50. The web service call exceeds the threshold limit for processing	Reduce the number of position lines in the Web service call
exception.authorization	You do not have the privilege to take this action. Please consult your PPM Administrator	You do not have the required access grants for this operation	Verify that your account has the required access grants
exception.cannotLoadPosition	Position id {0} does not exist.	The position ID does not exist	Specify a different position ID
exception.cannotViewStaffingProfile	User cannot view staffing profile {0}	You do not have the required access grants for this operation	Verify that your account has the required access grants

addPositionLines

Purpose

This operation adds position lines to a staffing profile in PPM Center.

Function

This operation enables external applications to add position lines to a staffing profile in PPM Center identified by staffing profile ID. The operation returns the resultObjectType object.

To perform this operation, the user must meet the following conditions:

- Have the edit staffing profile access grant
- In the Staffing profile ACL list
- Be the project manager and resource manager

Related information

The following data types:

- ["positionLineType" on page 275](#)
- ["effortContourType" on page 277](#)
- ["effortCellType" on page 277](#)
- ["assignmentType" on page 278](#)
- ["promisedAssignmentType" on page 278](#)
- ["skillProficiencyType" on page 279](#)
- ["userDataType" on page 279](#)
- ["resultObjectType" on page 288](#)

Input

An array of ["positionLineType" on page 275](#) object.

Return

An array of ["resultObjectType" on page 288](#) object

Java Interface

```
public OperationResultWSBean[]  
addPositionLines(PositionLineWSBean[] ids)
```

Parameters	Description
ids	Wrapper of the positionLineType object array

Java Examples

Example: adds position lines

```
private ResultObjectType[] addPositionLines(String  
serviceURL, PositionLineType[] ids) throws Exception{  
    ResultObjectType[] resultType = null;  
    StaffingProfileServiceStub stub = new  
StaffingProfileServiceStub(ctx, serviceURL);  
    AddPositionLinesDocument inDoc =  
AddPositionLinesDocument.Factory.newInstance();  
    AddPositionLinesDocument.AddPositionLines addPL =  
inDoc.addNewAddPositionLines();  
    addPL.setPositionLineArray(ids);  
  
    // Invoke web service  
    AddPositionLinesResponseDocument outDoc =  
stub.addPositionLines(inDoc);  
    resultType =  
outDoc.getAddPositionLinesResponse().getPositionIdArray();  
  
    return resultType;  
}
```

Errors and Exceptions

Possible root cause descriptions:

Message Code	Message	Cause(s)	Possible Corrective Action
exception.exceededThreshold	The web service call exceeds the threshold limit for processing. Please batch your updates	The number of position lines in one Web service call is limited to 50. The web service call exceeds the threshold limit for processing	Reduce the number of position lines in the Web service call
exception.authorization	You do not have the privilege to take this action. Please consult your PPM Administrator	You do not have the required access grants for this operation	Verify that your account has the required access grants
exception.cannotLoadSP	Staffing profile id {0} does not exist	The staffing profile ID does not exist	Specify a different staffing profile ID
exception.cannotEditStaffingProfile	User cannot edit staffing profile {0}	You do not have the required access grants for this operation	Verify that your account has the required access grants
exception.cannotAssignResources	User cannot assign resources to position {0}	You are not the resource pool manager or resource manager	Verify that your account has the required access grants
exception.hasDuplicateAssignments	Position {0} has duplicate assignments in input XML	The input position has duplicate assignments in the Web service call	Specify a different assignment for the position
exception.hasDuplicateSkills	Position {0} has duplicate skills in input XML	The input position has duplicate skills in the Web service call	Specify a different skill for the position

updatePositionLines

Purpose

This operation updates position lines of a staffing profile in PPM Center.

Function

This operation enables external applications to update position lines of staffing profile identified by staffing profile ID in PPM Center. The operation returns the resultObjectType object.

To perform this operation, the user must meet the following conditions:

- Have the edit staffing profile access grant
- In the Staffing profile ACL list
- Be the project manager and resource manager

Related information

The following data types:

- ["positionLineType" on page 275](#)
- ["effortContourType" on page 277](#)
- ["effortCellType" on page 277](#)
- ["assignmentType" on page 278](#)
- ["promisedAssignmentType" on page 278](#)
- ["skillProficiencyType" on page 279](#)
- ["userDataType" on page 279](#)
- ["resultObjectType" on page 288](#)

Input

An array of ["positionLineType" on page 275](#) object.

Return

An array of ["responseObjectType" on page 288](#) object

Java Interface

```
public OperationResultWSBean[]
updatePositionLines(PositionLineWSBean[] ids)
```

Parameters	Description
ids	Wrapper of the positionLineType object array

Java Examples

Example: updates position lines

```
private ResultObjectType[] updatePositionLines(String
serviceURL, PositionLineType[] ids) throws Exception{
    ResultObjectType[] resultType = null;
    StaffingProfileServiceStub stub = new
StaffingProfileServiceStub(ctx, serviceURL);
    UpdatePositionLinesDocument inDoc =
UpdatePositionLinesDocument.Factory.newInstance();
    UpdatePositionLinesDocument.UpdatePositionLines updatePL
= inDoc.addNewUpdatePositionLines();
    updatePL.setPositionLineArray(ids);

    // Invoke web service
    UpdatePositionLinesResponseDocument outDoc =
stub.updatePositionLines(inDoc);
    resultType =
outDoc.getUpdatePositionLinesResponse().getPositionIdArray()
```



```

;

    return resultType;
}

```

Errors and Exceptions

Possible root cause descriptions:

Message Code	Message	Cause(s)	Possible Corrective Action
exception.exceededThreshold	The web service call exceeds the threshold limit for processing. Please batch your updates	The number of position lines in one Web service call is limited to 50. The web service call exceeds the threshold limit for processing	Reduce the number of position lines in the Web service call
exception.authorization	You do not have the privilege to take this action. Please consult your PPM Administrator	You do not have the required access grants for this operation	Verify that your account has the required access grants
exception.cannotLoadPosition	Position id {0} does not exist.	The position ID does not exist	Specify a different position ID
exception.invalidStaffingProfileId	Staffing profile id {0} is invalid for position id {1}	The position's staffing profile ID is not equal to the staffing profile ID in the positionLineType object	Make sure the position's s staffing profile id is equal to the staffing profile id in the positionLineType object
exception.cannotEditStaffingProfile	User cannot edit staffing profile {0}	You do not have the required access grants for this operation	Verify that your account has the required access grants
exception.cannotAssignResources	User cannot	You are not the	Verify that your

Message Code	Message	Cause(s)	Possible Corrective Action
	assign resources to position {0}	resource pool manager or resource manager	account has the required access grants
exception.hasDuplicateAssignments	Position {0} has duplicate assignments in input XML	The position has duplicate assignments in the Web service call	Specify a different assignment for the position
exception.hasDuplicateSkills	Position {0} has duplicate skills in input XML	The position has duplicate skills in the Web service call	Specify a different skill for the position

deletePositionLines

Purpose

This operation deletes position lines of a staffing profile in PPM Center.

Function

This operation enables external applications to delete position lines of a staffing profile identified by position id in PPM Center. The operation returns an array of the resultObjectType object.

To perform this operation, the user must meet the following conditions:

- Have the edit staffing profile access grant
- In the Staffing profile ACL list

Related information

The following data types:

- ["positionLineType" on page 275](#)
- ["effortContourType" on page 277](#)
- ["effortCellType" on page 277](#)
- ["assignmentType" on page 278](#)
- ["promisedAssignmentType" on page 278](#)
- ["skillProficiencyType" on page 279](#)
- ["userDataType" on page 279](#)
- ["resultObjectType" on page 288](#)

Input

An array of position IDs.

Return

An array of resultObjectType object

Java Interface

```
public OperationResultWSBean[]  
deletePositionLines (long[] ids)
```

Parameters	Description
ids	Array of position IDs

Java Examples

Example: deletes position lines

```
private ResultObjectType[] deletePositionLines(String  
serviceURL, long[] ids) throws Exception{
```

```

        ResultObjectType[] resultType = null;
        StaffingProfileServiceStub stub = new
StaffingProfileServiceStub(ctx, serviceURL);
        DeletePositionLinesDocument inDoc =
DeletePositionLinesDocument.Factory.newInstance();
        DeletePositionLinesDocument.DeletePositionLines deletePL
= inDoc.addNewDeletePositionLines();
        deletePL.setPositionIdArray(ids);

        // Invoke web service
        DeletePositionLinesResponseDocument outDoc =
stub.deletePositionLines(inDoc);
        resultType =
outDoc.getDeletePositionLinesResponse().getPositionIdArray()
;

        return resultType;
}

```

Errors and Exceptions

Possible root cause descriptions:

Message Code	Message	Cause(s)	Possible Corrective Action
exception.exceededThreshold	The web service call exceeds the threshold limit for processing. Please batch your updates	The number of position lines in one Web service call is limited to 50. The web service call exceeds the threshold limit for processing	Reduce the number of position lines in the Web service call
exception.authorization	You do not have the privilege to take this action. Please consult your PPM Administrator	You do not have the required access grants for this operation	Verify that your account has the required access grants
exception.cannotEditStaffingProfile	User cannot edit staffing profile {0}	You do not have the required access grants for this operation	Verify that your account has the required access grants

getPositionIdsforStaffingProfile

Purpose

This operation reads position IDs for a staffing profile in PPM Center.

Function

This operation enables external applications to read position ids for a staffing profile identified by staffing profile ID in PPM Center. The operation returns an array of position IDs.

To perform this operation, the user must meet the following conditions:

- Have the view staffing profile access grant
- In the Staffing profile ACL list

Input

The staffing profile ID.

Return

An array of position lines IDs

Java Interface

```
public Long[] getPositionIdsforStaffingProfile (long id)
```

Parameters	Description
id	Staffing profile ID

Java Examples

Example: reads position lines for a staffing profile

```
private Long[] getPositionIdsforStaffingProfile(String
serviceURL, long id) throws Exception{

    Long[] result = null;
    StaffingProfileServiceStub stub = new
StaffingProfileServiceStub(ctx, serviceURL);
    GetPositionIdsforStaffingProfileDocument inDoc =
GetPositionIdsforStaffingProfileDocument.Factory.newInstance();
GetPositionIdsforStaffingProfileDocument.GetPositionIdsforStaff
ingProfile getPositionIds =
inDoc.addNewGetPositionIdsforStaffingProfile();
    getPositionIds.setStaffingProfileId(id);

    // Invoke web service
    GetPositionIdsforStaffingProfileResponseDocument outDoc =
stub.getPositionIdsforStaffingProfile(inDoc);
    result =
outDoc.getPositionIdsforStaffingProfileResponse().getPositionId
Array();

    return result;
}
```

Errors and Exceptions

Possible root cause descriptions:

Message Code	Message	Cause(s)	Possible Corrective Action
exception.authorization	You do not have the privilege to take this action. Please consult your PPM Administrator	You do not have the required access grants for this operation	Verify that your account has the required access grants
rm.ws.exception.StaffingProfileNotExist	Staffing profile does not exist	The staffing profile cannot	Verify if the staffing

Message Code	Message	Cause(s)	Possible Corrective Action
		be found by using the staffing profile ID	profile ID exists
exception.cannotViewStaffingProfile	User cannot view staffing profile {0}	You do not have the access grants of viewing the staffing profile	Verify that your account has the required access grants

clearStaffingProfileForecastAndAssignmet

Purpose

The operation is used to clear the future forecast and assignment for a completed staffing profile.

Function

This operation clears the forecast and assignment from a specified date for a completed staffing profile. The operation cannot be undone.

To perform this operation, the user must meet the following conditions:

- Have the edit staffing profile access grant
- In the staffing profile ACL list

Limitations

- The staffing profile must be in the "completed" status.
- The date from which to clear forecast and assignment must be later than the staffing profile's start

date.

Input

A `ClearSPForecastAndAssignmentDocument` object that contains the header and the parameters to clear forecast and assignment.

Return

A `ClearSPForecastAndAssignmentResponseDocument` object that contains the header, a response message and a budget ID.

Java Interface

```
public com.mercury.itg.ws.rm.client.ClearSPForecastAndAssignmentResponseDocument  
clearStaffingProfileForecastAndAssignment(  
com.mercury.itg.ws.rm.client.ClearSPForecastAndAssignmentDocument param30)
```

Parameters	Description
<code>ClearSPForecastAndAssignmentDocument</code>	Wrapper of Staffing Profile's ID and the 'clear from date'
<code>ClearSPForecastAndAssignmentResponseDocument</code>	Contains nothing.

Java Examples

Example: clears the forecast and assignment from a specified date for a staffing profile

```
private void clearStaffingProfileForecastAndAssignment(String  
serviceURL, long staffingProfileId, java.util.Date  
clearFromDate) throws Exception{  
  
    StaffingProfileServiceStub stub = new  
StaffingProfileServiceStub(ctx, serviceURL);  
    ClearStaffingProfileForecastAndAssignmentDocument inDoc =  
ClearStaffingProfileForecastAndAssignmentDocument.Factory.newInstance();
```



```

ClearStaffingProfileForecastAndAssignmentDocument.ClearStaffing
ProfileForecastAnd Assignment clearSP =
inDoc.addNewClearStaffingProfileForecastAndAssignment();
    clearSP.setStaffingProfileId(id);
    clearSP.setClearFromDate(clearFromDate);

```

Errors and Exceptions

Possible root cause descriptions:

Message Code	Message	Cause(s)	Possible Corrective Action
exception.spStatusMustBeCompleted	Staffing profile status must be 'completed'.	The staffing profile specified is not in 'completed' status	Change the staffing profile's status to 'completed' before call this function.
exception.clearFromDateMustBeAfterSPStart	The clear from date must be after the staffing profile start date.	The clear from date is before the staffing profile's start date.	Change the clear from date to make sure it's after the staffing profile's start date.
rm.ws.exception.notAuthorized	You do not have the privilege to take this action. Please consult your PPM Administrator.	The user has no access granted to clear the forecast and assignment.	Grant the 'edit positions' access authority to the user and change a user who has the access.

Chapter 9: Time Management Web Services

- ["Overview" below](#)
- ["References" below](#)
- ["Terms and Concepts" on the next page](#)
- ["Data Types" on page 332](#)
- ["Operations" on page 373](#)

Overview

Time Management Web services provides interfaces for accessing, creating, and updating time sheets in PPM Center. Operations for searching, submitting, approving, rejecting, reworking, freezing, closing, and canceling time sheets are also supported. Additionally, operations such as getting the actual time for work items are provided.

References

Data types definition:

`webservice_toolkit\java\conf\xsd\Time.xsd`

Operations definition:

`webservice_toolkit\java\conf\wsdl\TimeService.wsdl`

Java sample code:

`webservice_toolkit\java\client\src\examples\tm\TimeServiceClient.java`

Terms and Concepts

Time Sheet Policy

Time sheet policies are rules that control the creation and operation of time sheets. You can configure different time sheet policies, and then apply them to different resources individually as needed. For example, you may want to apply different time sheet policies to different resources depending on their business units or whether they are employees or contractors. Every resource is assigned a time sheet policy. One time sheet policy serves as the default (global) time sheet policy.

Period Type

A period type specifies the interval over which time sheets report the time that users worked on work items. The available period types are:

- Weekly

If the system uses this period type, each time period covers one week, starting by default on a Monday.

- Bi-Weekly

If the system uses this period type, each time period covers two weeks, starting by default on a Monday.

- Semi-Monthly

If the system uses this period type, there are two time periods per month, and the first time period always ends on the 15th of the month.

- Monthly

If the system uses this period type, each time period covers a full month.

Time Period

A time period (period) is a particular date range of a period type.

Charge Code

Charge codes are entities used as links between work items and charge accounts.

Activity

Activities are used to categorize work performed against a work item, such as design work or coding. Activities can also be used to classify work as depreciable for financial accounting.

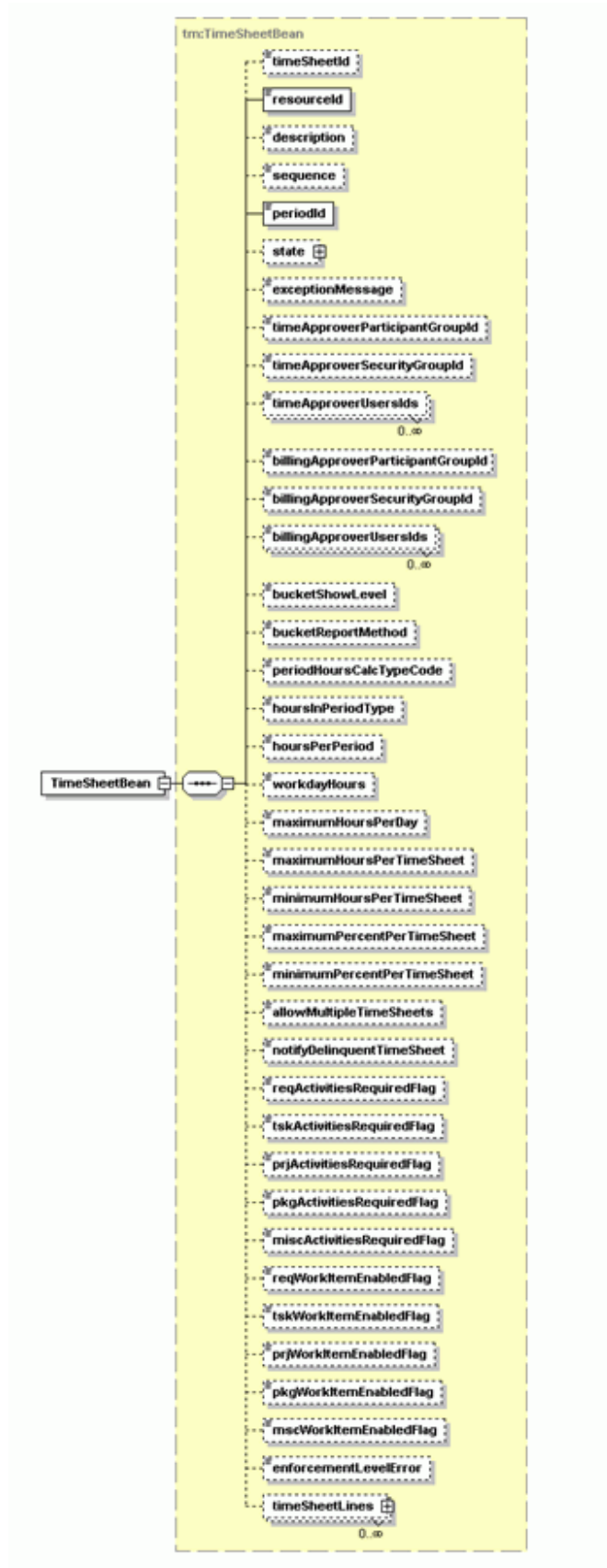
Data Types

Time Management Web services includes the following data types:

- ["TimeSheetBean" below](#)
- ["TimeSheetLineBean" on page 348](#)
- ["TimeActualsBean" on page 355](#)
- ["ChargeCodeBean" on page 356](#)
- ["UserDataBean" on page 357](#)
- ["TimeSheetStatus" on page 359](#)
- ["TimeSheetLineStatus" on page 360](#)
- ["TimeSheetPolicyBean" on page 361](#)
- ["TimeSheetSearchCriteriaBean" on page 368](#)
- ["TimeFilter" on page 370](#)
- ["WorkItemActualTime" on page 371](#)

TimeSheetBean

This is the common type in operations with time sheets. The policy fields are optional.



Property	Type	Description	Required	Default
timeSheetId	Long	Generated automatically when the time sheet object is created in PPM Center. This field is not required in the createTimeSheet operation, but it is required for other operations that use TimeSheetBean.	No	N/A
resourceId	Long	ID of the resource of the time sheet.	Yes	N/A
periodId	Long	Period ID of the Time Period of the time sheet.	Yes	N/A
description	String	Description of the time sheet. If you do not provide this property when you create a time sheet, it is automatically created with the default value: resourceName - periodName #sequence	No	resourceName - periodName #sequence
sequence	Integer	Read-only field. Numeric value starting from 1. Unique per resource and time period. If the policy allows multiple time sheets per resource and period, each subsequent time sheet that you create is automatically increment the sequence number.	No	1 (the first one)
state	TimeSheetStatus	Status of the time sheet. On the time sheet creation, the default value is "unsubmitted." See "TimeSheetStatus" on page 359 for more details.	No	unsubmitted
timeSheetLines	List	List of TimeSheetLineBean objects. See	No	N/A

Property	Type	Description	Required	Default
		"TimeSheetLineBean" on page 348		
bucketShowLevel	String	<p>Part of policy fields.</p> <p>The unit of measurement of each time bucket.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • PERIOD • DAY • HOURS <p>It is the only field that the system checks to determine whether policy fields are set. If you do not set this field, the system discards the values in the following fields marked with policy fields.</p>	No	N/A
bucketReportMethod	String	<p>Part of policy fields.</p> <p>Defines how time sheets report time.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • HOURS • DAYS • PERCENT <p>It is required when bucketShowLevel is set.</p>	No	N/A
hoursPerPeriod	Double	<p>Part of policy fields.</p> <p>Numeric value of how many hours should be per period.</p> <p>This field is relevant only when all of the following conditions are true:</p> <ul style="list-style-type: none"> • The value for bucketShowLevel is PERIOD. 	No	N/A

Property	Type	Description	Required	Default
		<ul style="list-style-type: none"> The value for bucketReportMethod is PERCENT. The value for periodHoursCalcTypeCode is FIXED. 		
periodHoursCalcTypeCode	String	Part of policy fields. Defines how period in hours should be calculated. Valid values: <ul style="list-style-type: none"> FIXED (for period) WORKDAY (for days). This field is relevant only when both of the following conditions are true: <ul style="list-style-type: none"> The value for bucketShowLevel is PERIOD. The value for bucketReportMethod is PERCENT. 	No	N/A
workdayHours	Double	Part of policy fields. Numeric value of how many hours should be in a workday. This field is relevant only when all of the following conditions are true: <ul style="list-style-type: none"> The value for bucketShowLevel is PERIOD. The value for bucketReportMethod is PERCENT. The value for periodHoursCalcTypeCode is WORKDAY. 	No	N/A

Property	Type	Description	Required	Default
allowMultipleTimeSheets	Boolean	<p>Part of policy fields.</p> <p>Boolean value - Set this field to true if the policy allows multiple time sheets.</p> <p>The default value for this field is False, which means multiple time sheets are not allowed by default.</p> <p>This field is required if you set the bucketShowLevel field.</p>	No	False
hoursInPeriodType	String	<p>Part of policy fields.</p> <p>Defines how period in hours should be calculated.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • FIXED (for period) • WORKDAY (for hours) 	No	N/A
notifyDelinquentTimeSheet	Boolean	<p>Part of policy fields.</p> <p>Boolean value - Set this field to true if you need the notification of delinquent time sheets. False otherwise.</p> <p>It is required if you set the bucketShowLevel field.</p>	No	False
maximumHoursPerDay	Double	<p>Part of policy fields.</p> <p>Numeric value of the max hours per day to enforce.</p> <p>This field is relevant only when both of the following conditions are true:</p> <ul style="list-style-type: none"> • The value for bucketShowLevel is Day. • The value for bucketReportMethod is HOURS. 	No	N/A
maximumDaysPerDay	Double	Part of policy fields.	No	N/A

Property	Type	Description	Required	Default
		<p>Numeric value of the max days per day to enforce.</p> <p>This field is relevant only when both of the following conditions are true:</p> <ul style="list-style-type: none"> The value for bucketShowLevel is Day. The value for bucketReportMethod is DAYS. 		
maximumHoursPerTimeSheet	Double	<p>Part of policy fields.</p> <p>Numeric value of the max hours per time sheet to enforce.</p> <p>This field is relevant only when the value for bucketReportMethod is HOURS.</p>	No	N/A
minimumHoursPerTimeSheet	Double	<p>Part of policy fields.</p> <p>Numeric value of the minimum hours per time sheet to enforce.</p> <p>This field is relevant only when the value for bucketReportMethod is HOURS.</p>	No	N/A
maximumDaysPerTimeSheet	Double	<p>Part of policy fields.</p> <p>Numeric value of the max days per time sheet to enforce.</p> <p>This field is relevant only when the value for bucketReportMethod is DAYS.</p>	No	N/A
minimumDaysPerTimeSheet	Double	<p>Part of policy fields.</p> <p>Numeric value of the</p>	No	N/A

Property	Type	Description	Required	Default
		<p>minimum days per time sheet to enforce.</p> <p>This field is relevant only when the value for bucketReportMethod is DAYS.</p>		
maximumPercentPerTimeSheet	Double	<p>Part of policy fields.</p> <p>Numeric value of the maximum percent per time sheet to enforce.</p> <p>This field is relevant only when both of the following conditions are true:</p> <ul style="list-style-type: none"> • The value for bucketShowLevel is PERIOD • The value for bucketReportMethod is PERCENT 	No	N/A
minimumPercentPerTimeSheet	Double	<p>Part of policy fields.</p> <p>Numeric value of the min percent per time sheet to enforce.</p> <p>This field is relevant only when both of the following conditions are true:</p> <ul style="list-style-type: none"> • The value for bucketShowLevel is PERIOD. • The value for bucketReportMethod is PERCENT. 	No	N/A
enforcementLevelError	Boolean	<p>Part of policy fields.</p> <p>Boolean value - Set this field to True if you want the system to prevent the submission of a time sheet when the</p>	No	False

Property	Type	Description	Required	Default
		<p>submission violates the policy enforcements. Set it to False if you want the system to display only a warning when a violation occurs.</p> <p>It is required if you set the bucketShowLevel field.</p>		
reqActivitiesRequiredFlag	Boolean	<p>Part of policy fields.</p> <p>Boolean value - Set this field to True if activities are required for Request work items, False otherwise.</p> <p>It is required if you set the bucketShowLevel field.</p>	No	False
tskActivitiesRequiredFlag	Boolean	<p>Part of policy fields.</p> <p>Boolean value - Set this field to True if activities are required for Task work items, False otherwise).</p> <p>It is required if you set the bucketShowLevel field.</p>	No	False
prjActivitiesRequiredFlag	Boolean	<p>Part of policy fields.</p> <p>Boolean value - Set this field to True if activities are required for Project work items, False otherwise.</p> <p>It is required if you set the bucketShowLevel field.</p>	No	False
pkgActivitiesRequiredFlag	Boolean	<p>Part of policy fields.</p> <p>Boolean value - Set this field to True if activities are required for Package work items, False otherwise.</p> <p>It is required when bucketShowLevel is set.</p>	No	False
miscActivitiesRequiredFla	Boolean	Part of policy fields.	No	False

Property	Type	Description	Required	Default
g		<p>Boolean value - Set this field to True if activities are required for Misc work items, False otherwise.</p> <p>It is required if you set the bucketShowLevel field.</p>		
reqWorkItemEnabledFlag	Boolean	<p>Part of policy fields.</p> <p>Boolean value - Set this field to True if you want to track Request work items for this time sheet, False otherwise. It is required if you set the bucketShowLevel field.</p> <p>The default value is taken from server.conf.</p> <p>com.kintana.core.server.PURGE_SERVER_PROPERTIES_TABLE=true</p> <p>com.kintana.core.server.ENABLE_TM_WORK_ITEM_REQUESTS=true</p>	No	Taken from server.conf
tskWorkItemEnabledFlag	Boolean	<p>Part of policy fields.</p> <p>Boolean value - Set this field to True if you want to track Task work items for this time sheet, False otherwise.</p> <p>It is required if you set the bucketShowLevel field.</p> <p>The default value is taken from server.conf.</p> <p>com.kintana.core.server.PURGE_SERVER_PROPERTIES_TABLE=true</p> <p>com.kintana.core.server.ENABLE_TM_WORK_ITEM_TASKS=true</p>	No	Taken from server.conf
prjWorkItemEnabledFlag	Boolean	<p>Part of policy fields.</p>	No	Taken from

Property	Type	Description	Required	Default
		<p>Boolean value - Set this field to True if you want to track Project work items for this time sheet, False otherwise.</p> <p>It is required if you set the bucketShowLevel field.</p> <p>The default value is taken from server.conf.</p> <p>com.kintana.core.server.PURGE_SERVER_PROPERTIES_TABLE=true</p> <p>com.kintana.core.server.ENABLE_TM_WORK_ITEM_PROJECTS=true</p>		server.conf
pkgWorkItemEnabledFlag	Boolean	<p>Part of policy fields.</p> <p>Boolean value - Set this field to True if you want to track Package work items in this time sheet, False otherwise.</p> <p>It is required if you set the bucketShowLevel field.</p> <p>The default value is taken from server.conf.com.kintana.core.server.</p> <p>PURGE_SERVER_PROPERTIES_TABLE=true</p> <p>com.kintana.core.server.ENABLE_TM_WORK_ITEM_PACKAGES =true</p>	No	Taken from server.conf
mscWorkItemEnabledFlag	Boolean	<p>Part of policy fields.</p> <p>Boolean value - True if you want to track Misc work items in this time sheet, False otherwise).</p> <p>It is required when bucketShowLevel is set.</p>	No	Taken from server.conf

Property	Type	Description	Required	Default
		<p>The default value is taken from server.conf.</p> <p>com.kintana.core.server.PURGE_SERVER_PROPERTIES_TABLE=true</p> <p>com.kintana.core.server.ENABLE_TM_WORK_ITEM_MISC =true</p>		
exceptionMessage	String	<p>Read-only field.</p> <p>If there is a violation of the policy, this field is set by the system automatically when you save the Time Sheet.</p>	No	N/A
timeApproverParticipantGroupId	Long	<p>Read-only field.</p> <p>The system reevaluates possible approvers when you save and load the time sheet. At the same time, the system sets this field automatically.</p> <p>This field is set only if the possible Time Approver of all the lines in the time sheet is a Participant Group (such as Project Managers). Otherwise it is null.</p>	No	N/A
timeApproverSecurityGroupId	Long	<p>Read-only field.</p> <p>The system reevaluates possible approvers when you save and load the time sheet. At the same time, the system sets this field automatically.</p> <p>This field is set only if the possible Time Approver of all the lines in the time sheet is a Security Group. Otherwise it is null.</p>	No	N/A
timeApproverUserIds	List	<p>Read-only field.</p> <p>List of Long object values,</p>	No	N/A

Property	Type	Description	Required	Default
		<p>each representing a User ID.</p> <p>The system reevaluates possible approvers when you save and load the time sheet. At the same time, the system sets this field automatically.</p> <p>This field must be set on reevaluation of approvers using the following order of precedence:</p> <ul style="list-style-type: none"> • If timeApproverParticipantGroupid is not null, this field is set to the IDs of all the users in this participant group. • Else, if timeApproverSecurityGroupid is not null, this field is set to the IDs of all the users in this security group. • Else, if all lines have the same user as possible time approver, this field is set to the IDs of this user ID. • Else, this field is set to the ID of the default time approver of the resource of this time sheet. 		
billingApproverParticipantGroupid	Long	<p>Read-only field.</p> <p>The system reevaluates possible approvers when you save and load the time sheet. At the same time, the system sets this field automatically.</p> <p>This field is set only if the possible Billing Approver of this time sheet is a Participant</p>	No	N/A

Property	Type	Description	Required	Default
		Group (such as Project Managers). Otherwise it is null.		
billingApproverSecurityGroupId	Long	<p>Read-only field.</p> <p>The system reevaluates possible approvers when you save and load the time sheet. At the same time, the system sets this field automatically.</p> <p>This field is set only if the possible Billing Approver of this time sheet is a Security Group. Otherwise it is null.</p>	No	N/A
billingApproverUserIds	List	<p>Read-only field.</p> <p>List of Long object values, each representing a User ID.</p> <p>The system reevaluates possible approvers when you save and load the time sheet. At the same time, the system sets this field automatically.</p> <p>This field must be set on re-evaluation of approvers using this order of precedence:</p> <ul style="list-style-type: none"> • If billingApproverParticipantGroupId is not null, this field is set to the IDs of all the users in this participant group. • Else, if billingApproverSecurityGroupId is not null, this field is set to the IDs of all the users in this security group. • Else, this field is set to the ID of the default billing approver as determined by 	No	N/A

Property	Type	Description	Required	Default
		the re-evaluation of the approvers.		

Related Information

TimeSheetBean is used as the INPUT in the following operations:

- createTimeSheet
- updateTimeSheet
- approveTimeSheet
- approveTimeSheetLine
- rejectTimeSheet
- rejectTimeSheetLine
- reworkTimeSheetLine
- freezeTimeSheet
- closeTimeSheet
- cancelTimeSheet

TimeSheetBean is used as the RETURN type in the following operations:

- createTimeSheet
- updateTimeSheet
- approveTimeSheet
- approveTimeSheetLine
- rejectTimeSheet
- rejectTimeSheetLine
- reworkTimeSheetLine
- freezeTimeSheet

- `closeTimeSheet`
- `cancelTimeSheet`

Note:

`TimeSheetBean` is used as the INPUT in the following operations:

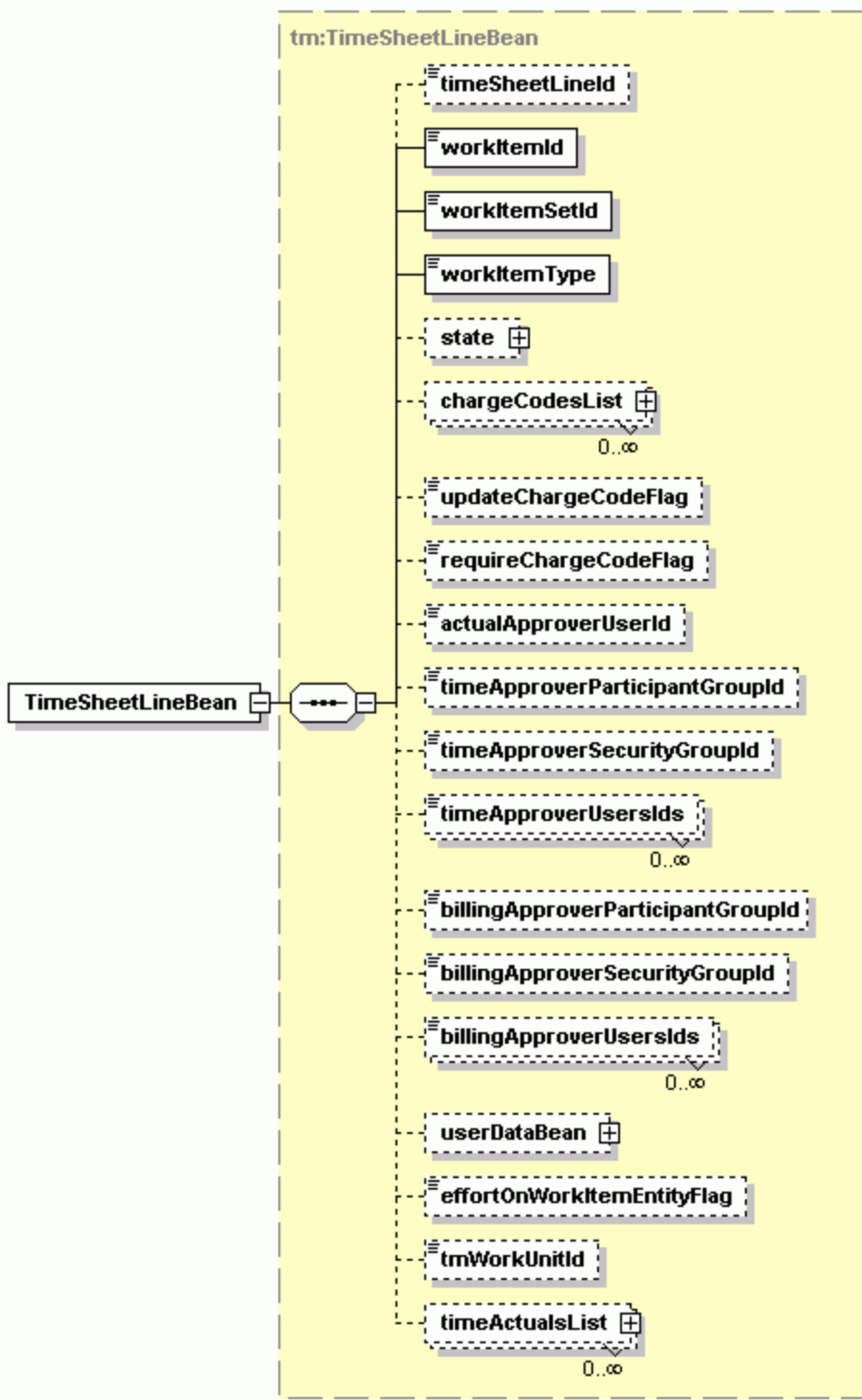
- `"createTimeSheet"` on page 374
- `"updateTimeSheet"` on page 386
- `"approveTimeSheet"` on page 411
- `"approveTimeSheetLine"` on page 415
- `"rejectTimeSheet"` on page 420
- `"rejectTimeSheetLine"` on page 424
- `"reworkTimeSheetLine"` on page 429
- `"freezeTimeSheet"` on page 434
- `"closeTimeSheet"` on page 438
- `"cancelTimeSheet"` on page 441

`TimeSheetBean` is used as the OUTPUT in the following operations:

- `"createTimeSheet"` on page 374
- `"updateTimeSheet"` on page 386
- `"approveTimeSheet"` on page 411
- `"approveTimeSheetLine"` on page 415
- `"rejectTimeSheet"` on page 420
- `"rejectTimeSheetLine"` on page 424
- `"reworkTimeSheetLine"` on page 429
- `"freezeTimeSheet"` on page 434
- `"closeTimeSheet"` on page 438
- `"cancelTimeSheet"` on page 441

TimeSheetLineBean

TimeSheetLineBean represents the time sheet line in a time sheet. It is used in TimeSheetBean. This data type is used in "[TimeSheetBean](#)" on page 332.



Property	Type	Description	Required	Default
timeSheetLineId	Long	Generated automatically by the system when a time sheet line object in PPM Center is created.	Yes if the timesheet line has already been existed	N/A
workItemId	String	String object which holds the ID of the work item of the line.	Yes	N/A
workItemSetId	String	String object which holds the ID of the parent of the work item of the line. The parent of a request should be the request type. The parent of a task should be the project. The parent of a project should be the same project. The parent of a package should be the workflow. The parent of a misc should be the constant MISC.	Yes	N/A
workItemType	String	String object which holds the type of the work item of the line. Valid values: <ul style="list-style-type: none"> • REQUEST • TASK • PROJECT • PACKAGE • MISC 	Yes	N/A
State	TimeSheetLine Status	This field is set by default to be "unsubmitted" during the creation of a line. See	No	unsubmitted

Property	Type	Description	Required	Default
		"TimeSheetLineStatus" on page 360 for more details.		
timeActualsList	List	<p>List of TimeActualsBean objects.</p> <p>See "TimeActualsBean" on page 355 for more details.</p> <p>For each Line there must be at least two records of Time Actuals list. The two records include same actuals with different total flag Y and N. If there are more than one non-total records (records with total flag as N), each non-total record must be associated with an Activity ID. If there is only one non-total record, it may or may not be associated with an Activity ID.</p>	No	N/A
chargeCodesList	List	<p>List of ChargeCodeBean objects.</p> <p>See "ChargeCodeBean" on page 356 for more details.</p> <p>The sum of all percentages in charge code list must be 100, otherwise an exception (ex.chargeCodePercentagesMustSum100) is thrown.</p>	No	N/A
updateChargeCodeFlag	Boolean	<p>Boolean value.</p> <p>True if charge codes can be updated in the time sheet, False otherwise.</p>	No	True
requireChargeCodeFlag	Boolean	<p>Boolean value.</p> <p>True if charge codes are required in the time sheet, False otherwise.</p>	No	False
userDataBean	UserDataBean	UserDataBean object.	No	N/A

Property	Type	Description	Required	Default
		<p>See "UserDataBean" on page 357 for more details.</p> <p>Holds an array of up to 20 user data hidden values and a corresponding array of up to 20 user data visible values.</p>		
actualApproverUserId	Long	<p>Long value of the user ID of the user who approves the actual time logged in the line.</p> <p>It is required when the state is "approved."</p>	Yes If line is approved	N/A
effortOnWorkItemEntityFlag	Boolean	<p>Boolean value.</p> <p>True if effort must be synched to Project Management or Demand Management when you save the effort, False otherwise.</p>	No	False
tmWorkUnitId	Long	<p>Long value of the ID of the TM work unit, which is a record in the staging table that helps in synching the effort between Time Management and Project Management or Demand Management.</p>	After first save	N/A
timeApproverParticipantGroupId	Long	<p>Read-only field.</p> <p>The system reevaluates possible approvers when you save and load the time sheet. At the same time, the system sets this field automatically.</p> <p>This field is set only if the possible Time Approver of the line is a Participant Group (such as Project Managers). Otherwise it is null.</p>	No	N/A
timeApproverSecurityGroupId	Long	<p>Read-only field.</p> <p>The system reevaluates possible approvers when you save and load the time sheet. At</p>	No	N/A

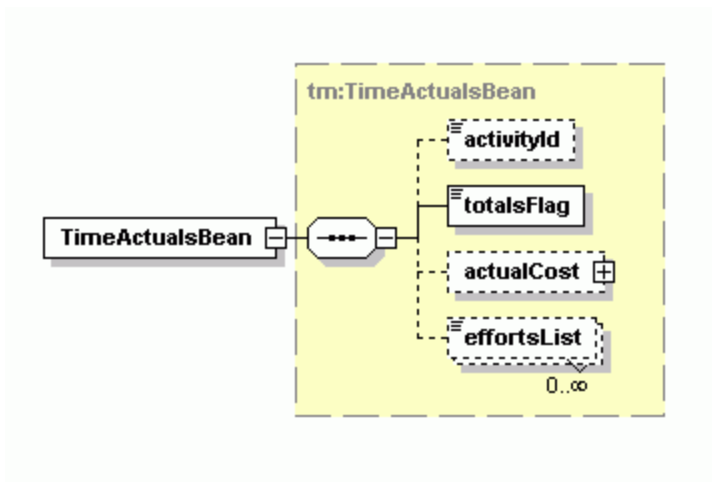
Property	Type	Description	Required	Default
		<p>the same time, the system sets this field automatically.</p> <p>This field is set only if the possible Time Approver of the line is a Security Group. Otherwise it is null.</p>		
timeApproverUsersIds	List	<p>Read-only field.</p> <p>List of Long object values, each representing a User ID.</p> <p>The system reevaluates possible approvers when you save and load the time sheet. At the same time, the system sets this field automatically.</p> <p>This field must be set on re-evaluation of approvers using this order of precedence:</p> <ul style="list-style-type: none"> • If timeApproverParticipantGroupid is not null, this field is set to the IDs of all the users in this participant group. • Else, if timeApproverSecurityGroupid is not null, this field is set to the IDs of all the users in this security group. • Else, this field is set to the ID of the time approver of the line as determined by the re-evaluation of the approvers. 	No	N/A
billingApproverParticipantGroupid	Long	<p>Read-only field.</p> <p>The system reevaluates possible approvers when you save and load the time sheet. At the same time, the system sets this field automatically.</p> <p>This field is set only if the</p>	No	N/A

Property	Type	Description	Required	Default
		possible Billing Approver of the time sheet of this line is a Participant Group (such as Project Managers). Otherwise it is null.		
billingApproverSecurityGroupId	Long	<p>Read-only field.</p> <p>The system reevaluates possible approvers when you save and load the time sheet. At the same time, the system sets this field automatically.</p> <p>This field is set only if the possible Billing Approver of the time sheet of this line is a Security Group. Otherwise it is null.</p>	No	N/A
billingApproverUserIds	List	<p>Read-only field.</p> <p>List of Long object values, each representing User ID.</p> <p>The system reevaluates possible approvers when you save and load the time sheet. At the same time, the system sets this field automatically.</p> <p>This field must be set on re-evaluation of approvers using this order of precedence:</p> <ul style="list-style-type: none"> • If billingApproverParticipantGroupId is not null, this field is set to the IDs of all the users in this participant group. • Else, if billingApproverSecurityGroupId is not null, this field is set to the IDs of all the users in this security group. • Else, this field is set to the ID of the time approver of the 	No	N/A

Property	Type	Description	Required	Default
		line as determined by the re-evaluation of the approvers.		

TimeActualsBean

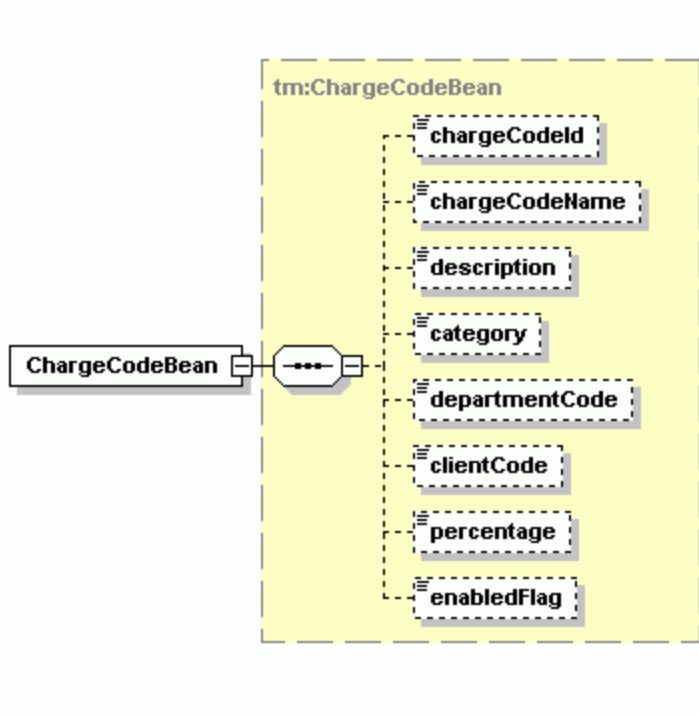
TimeActualsBean is used to hold the actual effort for each time sheet line. It is used in TimeSheetLineBean. This data type is used in ["TimeSheetLineBean" on page 348](#).



Property	Type	Description	Required	Default
activityId	Long	ID of the activity in PPM Center.	No	N/A
effortsList	List	List of objects with the Double type in the size of the time period of the time sheet. Each element holds the effort for the corresponding day by order. The effort must be provided according to the time sheet policy.	No	All zeros
totalsFlag	Boolean	Boolean value. True if this record is the totals record of the line, False otherwise.	Yes	N/A
actualCost	TaskCostBean	Object of "CostBean" on page 175 . See Web Services for Project Management.	No	N/A

ChargeCodeBean

ChargeCodeBean is used to hold the charge code and the percentage in a time sheet line. It is used in TimeSheetLineBean. This data type is used in ["TimeSheetLineBean" on page 348](#).

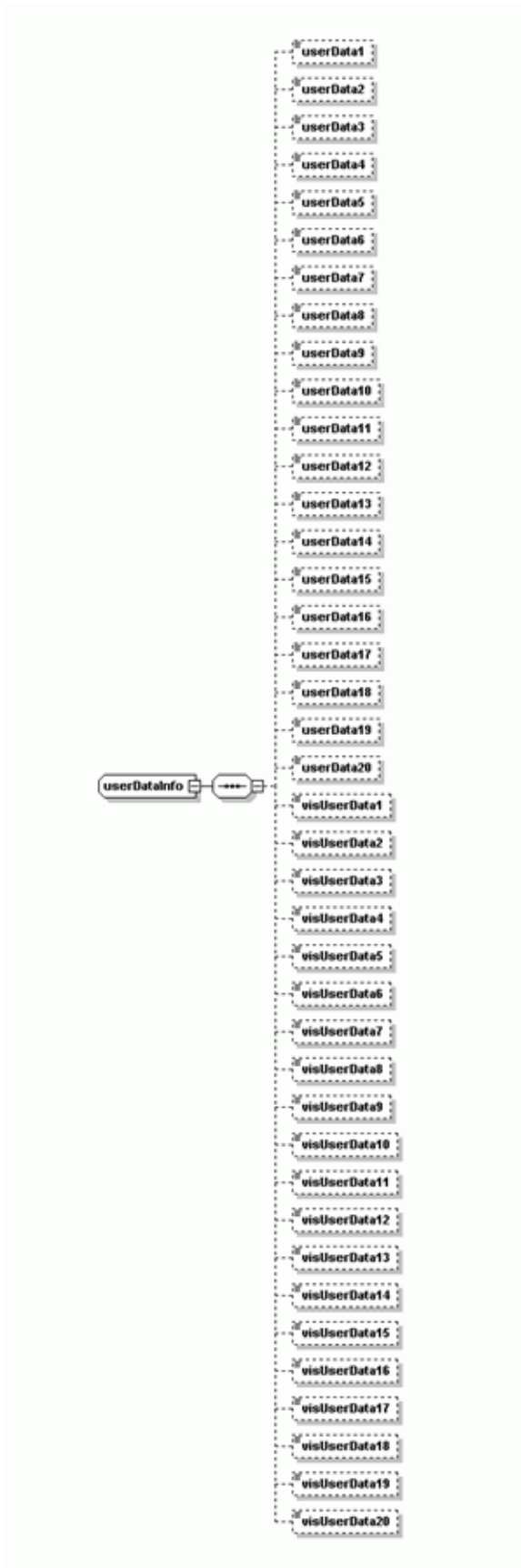


Property	Type	Description	Required	Default
chargeCodeId	Long	ID of the charge code.	Yes	N/A
chargeCodeName	String	String value of the charge code name.	No	N/A
Description	String	String value of the charge code description.	No	N/A
enabledFlag	Boolean	Boolean value. True if this charge code is enabled, False otherwise.	Yes	N/A
category	String	String value of the charge code category.	No	N/A
departmentCode	String	String value of the charge code department code.	No	N/A

Property	Type	Description	Required	Default
clientCode	String	String value of the charge code client code.	No	N/A
Percentage	Double	Double value of the percentage of this charge code with the range of 0 to 100. The sum of all percentages in the Charge Codes list must be 100.	Yes	N/A

UserDataBean

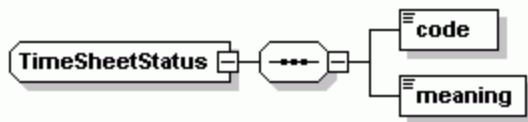
UserDataBean holds the additional fields in time sheet lines. It is used in TimeSheetLineBean. This data type is used in ["TimeSheetLineBean" on page 348](#).



Property	Type	Description	Required	Default
userData	String []	Array of 20 String values of the hidden value of user data.	No	N/A
visibleUserData	String []	Array of 20 String values of the visible value of user data. Each corresponds to the hidden value in the userData array, by order.	No	N/A

TimeSheetStatus

TimeSheetStatus is used to hold the status for each time sheet. It is used in TimeSheetBean. The code property value and of the meaning property value must match with each other. This data type is used in [" TimeSheetBean" on page 332.](#)

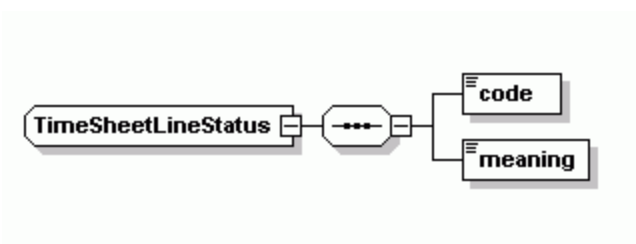


Property	Type	Description	Required	Default
code	int	Numeric value. Valid numeric values are (the corresponding meanings are listed in the brackets): <ul style="list-style-type: none"> • 1 (for unsubmitted) • 2 (for pending-approval) • 3 (for in-rework) • 4 (for approved) • 5 (for cancelled) • 6 (for frozen) • 7 (for closed) 	Yes	N/A

Property	Type	Description	Required	Default
meaning	String	String value. Valid values are: <ul style="list-style-type: none"> • unsubmitted • pending-approval • in-rework • approved • cancelled • frozen • closed 	Yes	N/A

TimeSheetLineStatus

TimeSheetLineStatus is used to hold the status for a time sheet line. It is used in TimeSheetLineBean. The code property value and of the meaning property value must match with each other. This data type is used in ["TimeSheetLineBean" on page 348](#).



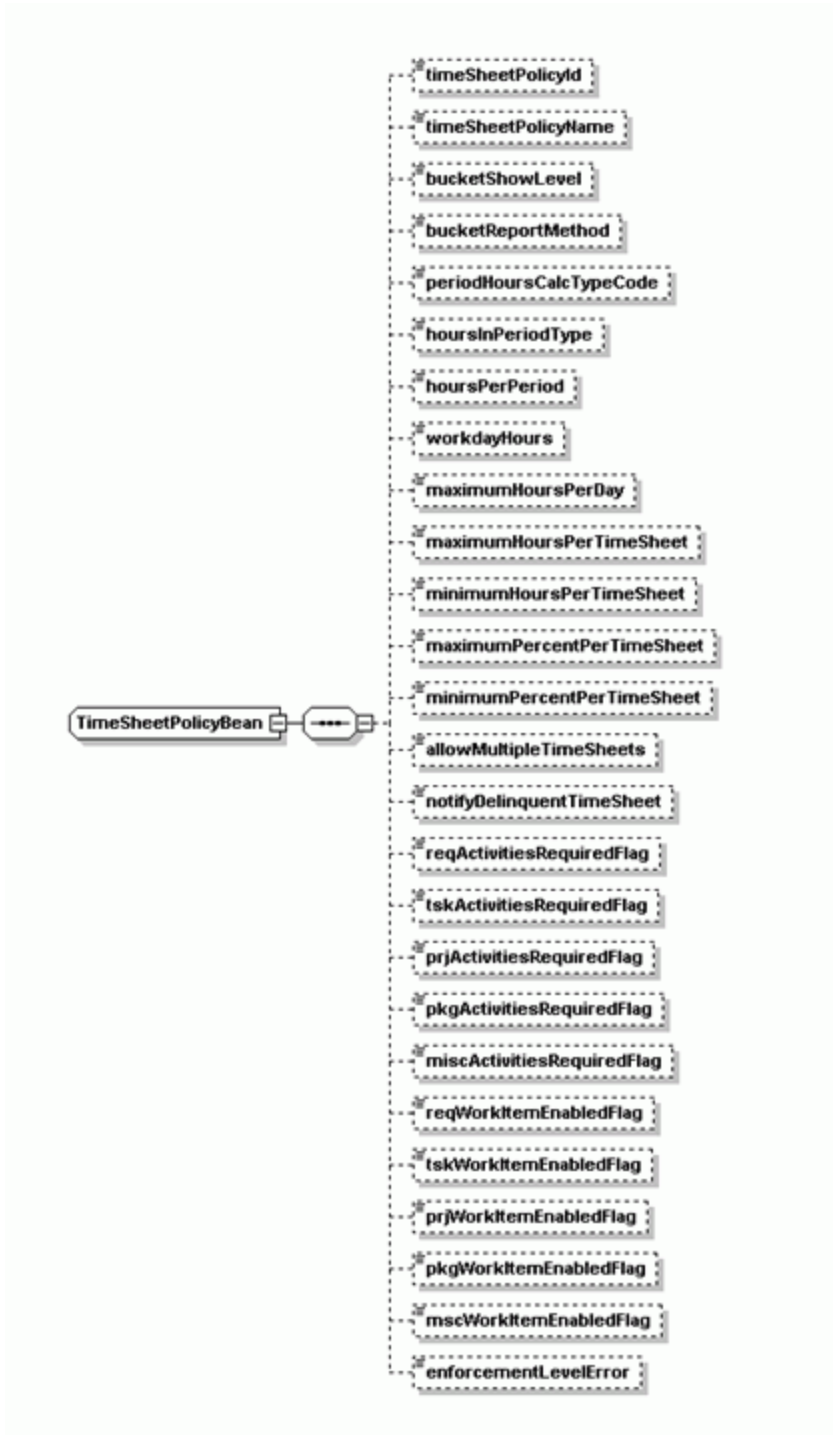
Property	Type	Description	Required	Default
code	int	Numeric value. Valid numeric values are (the corresponding meanings are listed in the brackets): <ul style="list-style-type: none"> • 1 (for unsubmitted) • 2 (for pending-approval) • 3 (for in-rework) • 4 (for approved) • 5 (for cancelled) 	Yes	N/A

Property	Type	Description	Required	Default
		<ul style="list-style-type: none">• 6 (for frozen)• 7 (for closed)		
meaning	String	String value. Valid values are: <ul style="list-style-type: none">• unsubmitted• pending-approval• in-rework• approved• cancelled• frozen• closed	Yes	N/A

TimeSheetPolicyBean

TimeSheetPolicyBean is used to hold the information for an existing time sheet policy.

Note: TimeSheetPolicyBean is used as the OUTPUT in ["getTimeSheetPolicy" on page 403](#).



Property	Type	Description	Required	Default
timeSheetPolicyId	Long	ID of the time sheet policy.	Yes	N/A
timeSheetPolicyName	String	String value of the time sheet policy name.	Yes	N/A
bucketShowLevel	String	Unit of measurement for each time bucket. Valid values: <ul style="list-style-type: none"> • PERIOD • DAY • HOURS 	Yes	N/A
bucketReportMethod	String	Defines how time sheets report time. Valid values: <ul style="list-style-type: none"> • HOURS • DAYS • PERCENT 	Yes	N/A
hoursPerPeriod	Double	Numeric value that defines how many hours a single period contains. This field is relevant only when all of the following conditions are true: <ul style="list-style-type: none"> • The value for bucketShowLevel is PERIOD • The value for bucketReportMethod is PERCENT • The value for periodHoursCalcTypeCode is FIXED 	No	N/A
periodHoursCalcTypeCode	String	Defines how period in hours should be calculated. Valid values: <ul style="list-style-type: none"> • FIXED (for period) 	No	N/A

Property	Type	Description	Required	Default
		<ul style="list-style-type: none"> • WORKDAY (for days). <p>This field is relevant only when both of the following conditions are true:</p> <ul style="list-style-type: none"> • The value for bucketShowLevel is PERIOD • The value for bucketReportMethod is PERCENT 		
workdayHours	Double	<p>Numeric value of how many hours are in a workday. Relevant when time is measured by day or hours.</p> <p>This field is relevant only when all of the following conditions are true:</p> <ul style="list-style-type: none"> • The value for bucketShowLevel is PERIOD. • The value for bucketReportMethod is PERCENT. • The value for periodHoursCalcTypeCode is WORKDAY. 	No	N/A
allowMultipleTimeSheets	Boolean	Boolean value - True if the policy allows multiple time sheets, False otherwise.	Yes	False
hoursInPeriodType	String	<p>Defines how period in hours should be calculated.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • FIXED (for period) • WORKDAY (for hours) 	No	N/A
notifyDelinquentTimeSheet	Boolean	Boolean value - True if you need the notification of delinquent time, False otherwise.	Yes	False

Property	Type	Description	Required	Default
maximumHoursPerDay	Double	Numeric value of the max hours per day to enforce. Relevant when time is measured by day. This field is relevant only when both of the following conditions are true: <ul style="list-style-type: none"> The value for bucketShowLevel is Day. The value for bucketReportMethod is HOURS. 	No	N/A
maximumDaysPerDay	Double	Numeric value of the max days per day to enforce. Relevant when time is measured by day. This field is relevant only when both of the following conditions are true: <ul style="list-style-type: none"> The value for bucketShowLevel is Day. The value for bucketReportMethod is DAYS. 	No	N/A
maximumHoursPerTimeSheet	Double	Numeric value of the max hours per time sheet to enforce. This field is relevant only when the value for bucketReportMethod is HOURS.	No	N/A
minimumHoursPerTimeSheet	Double	Numeric value of the min hours per time sheet to enforce. This field is relevant only when the value for bucketReportMethod is HOURS.	No	N/A
maximumDaysPerTimeSheet	Double	Numeric value of the max days per time sheet to enforce. This field is relevant only when the value for bucketReportMethod is DAYS.	No	N/A
minimumDaysPerTimeSheet	Double	Numeric value of the min days	No	N/A

Property	Type	Description	Required	Default
		per time sheet to enforce. This field is relevant only when the value for bucketReportMethod is DAYS.		
maximumPercentPerTimeSheet	Double	Numeric value of the max percent per time sheet to enforce. This field is relevant only when both of the following conditions are true: <ul style="list-style-type: none"> • The value for bucketShowLevel is PERIOD. • The value for bucketReportMethod is PERCENT. 	No	N/A
minimumPercentPerTimeSheet	Double	Numeric value of the min percent per time sheet to enforce. This field is relevant only when both of the following conditions are true: <ul style="list-style-type: none"> • The value for bucketShowLevel is PERIOD. • The value for bucketReportMethod is PERCENT. 	No	N/A
enforcementLevelError	Boolean	Boolean value - Set this field to True if you want the system to prevent the submission of a time sheet when the submission violates the policy enforcements. Set it to False if you want the system to display only a warning when a violation occurs.	Yes	False
reqActivitiesRequiredFlag	Boolean	Boolean value - True if activities are required for Request work items, False otherwise.	Yes	False
tskActivitiesRequiredFlag	Boolean	Boolean value - True if activities	Yes	False

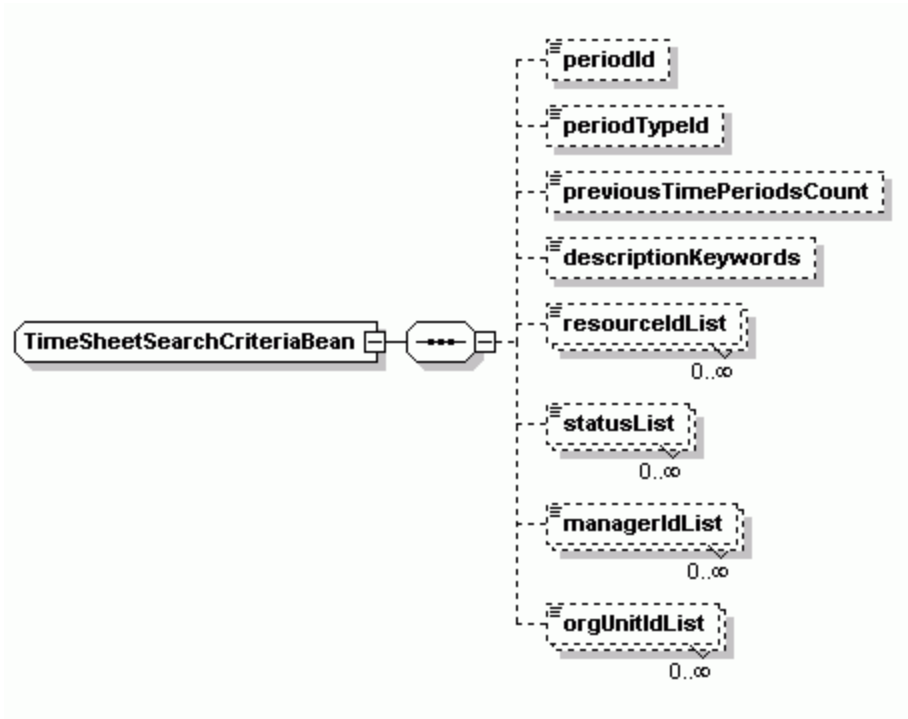
Property	Type	Description	Required	Default
	n	are required for Task work items, False otherwise.		
prjActivitiesRequiredFlag	Boolean	Boolean value - True if activities are required for Project work items, False otherwise.	Yes	False
pkgActivitiesRequiredFlag	Boolean	Boolean value - True if activities are required for Package work items, False otherwise.	Yes	False
miscActivitiesRequiredFlag	Boolean	Boolean value - True if activities are required for Misc work items, False otherwise.	Yes	False
reqWorkItemEnabledFlag	Boolean	Boolean value - True if Request work items are enabled for this time sheet, False otherwise. The default value is taken from server.conf. com.kintana.core.server.PURGE_SERVER_PROPERTIES_TABLE=true com.kintana.core.server.ENABLE_TM_WORK_ITEM_REQUESTS=true	Yes	Taken from server.conf
tskWorkItemEnabledFlag	Boolean	Boolean value - True if Task work items are enabled for this time sheet, False otherwise The default value is taken from server.conf. com.kintana.core.server.PURGE_SERVER_PROPERTIES_TABLE=true com.kintana.core.server.ENABLE_TM_WORK_ITEM_TASKS=true	Yes	Taken from server.conf
prjWorkItemEnabledFlag	Boolean	Boolean value - True if Project work items are enabled for this time sheet, False otherwise. The default value is taken from	Yes	Taken from server.conf

Property	Type	Description	Required	Default
		server.conf. com.kintana.core.server.PURGE_SERVER_PROPERTIES_TABLE=true com.kintana.core.server.ENABLE_TM_WORK_ITEM_PROJECTS=true		
pkgWorkItemEnabledFlag	Boolean	Boolean value - True if Package work items are enabled for this time sheet, False otherwise. The default value is taken from server.conf. com.kintana.core.server.PURGE_SERVER_PROPERTIES_TABLE=true com.kintana.core.server.ENABLE_TM_WORK_ITEM_PACKAGES =true	Yes	Taken from server.conf
mscWorkItemEnabledFlag	Boolean	Boolean value - True if Misc work items are enabled for this time sheet, False otherwise. The default value is taken from server.conf. com.kintana.core.server.PURGE_SERVER_PROPERTIES_TABLE=true com.kintana.core.server.ENABLE_TM_WORK_ITEM_MISC =true	Yes	Taken from server.conf

TimeSheetSearchCriteriaBean

TimeSheetSearchCriteriaBean holds the parameters which are used for searching time sheets.

Note: TimeSheetSearchCriteriaBean is used as the INPUT in ["searchTimeSheets"](#) on page 397.

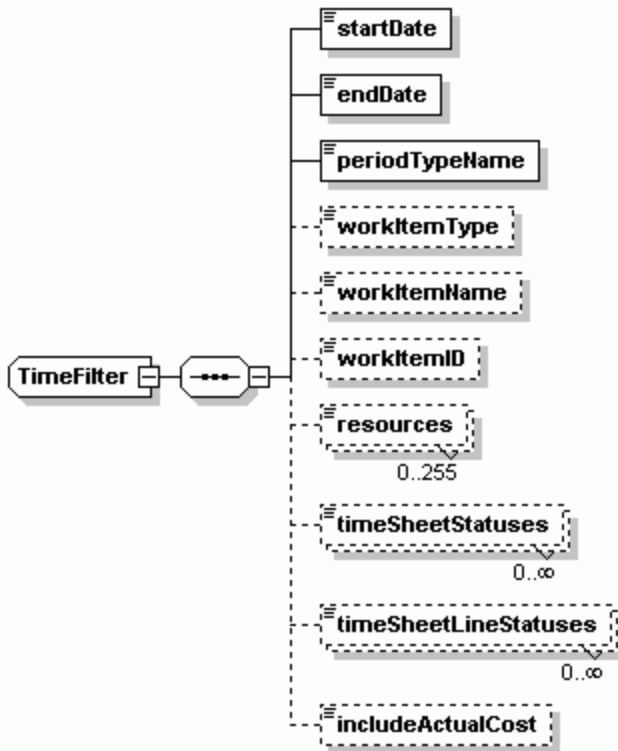


Property	Type	Description	Required	Default
periodTypeId	Long	Numeric value of a Time Period Type ID.	No	N/A
periodId	Long	Numeric value of a Time Period ID. Must correspond to the period type ID, if the period type ID exists.	No	N/A
previousTimePeriodsCount	Long	Numeric value that defines how many previous time periods to include in the results.	No	0
resourceIdList	List	List of Long values of Resource IDs.	No	N/A
descriptionKeywords	String	String value of keywords to search from time sheets descriptions.	No	N/A
statusList	List	List of Long values of statuses. Must correspond to the statuses constants in the code field of "TimeSheetStatus" on page 359 .	No	N/A
managerIdList	List	List of Long values of User IDs of resource managers.	No	N/A
orgUnitIdList	List	List of Long values of Org Unit IDs.	No	N/A

TimeFilter

TimeFilter is holds the parameters which are used to get the work item actuals.

Note: TimeFilter is used as the INPUT in "getActualTime" on page 400.



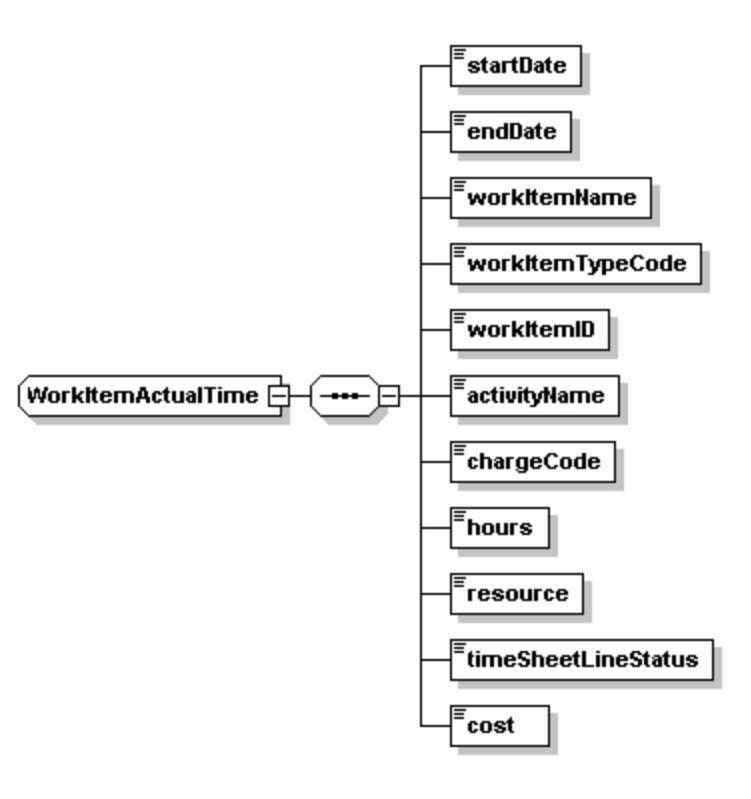
Property	Type	Description	Required	Default
startDate	Date	Date value of the start of the work item effort.	Yes	N/A
endDate	Date	Date value of the end of the work item effort.	Yes	N/A
periodTypeName	String	String value of the time period type name. Valid values: <ul style="list-style-type: none"> Weekly 	Yes	N/A

Property	Type	Description	Required	Default
		<ul style="list-style-type: none"> • Bi-Weekly • Semi-Monthly • Monthly 		
workItemType	String	String value of the work item type. Valid values: <ul style="list-style-type: none"> • REQUEST • TASK • PROJECT • PACKAGE • MISC 	No	N/A
workItemName	String	String value of the work item name.	No	N/A
workItemID	String	String value of the work item ID.	No	N/A
resources	List	List of resources usernames.	No	N/A
timeSheetStatuses	List	List of time sheet statuses. Constant must correspond to the statuses constants in meaning field of "TimeSheetStatus" on page 359 .	No	N/A
timeSheetLineStatuses	List	List of time sheet line statuses Constant must correspond to the statuses constants in meaning field of "TimeSheetLineStatus" on page 360 .	No	N/A
includeActualCost	boolean	True if you want to get the actual cost information in the results, False otherwise.	No	N/A

WorkItemActualTime

It is used to hold the results of the `getActualTime` operation.

Note: `TimeFilter` is used as the OUTPUT in ["getActualTime" on page 400](#).



Property	Type	Description	Required	Default
startDate	Date	Date value of the start of the work item effort.	No	N/A
endDate	Date	Date value of the end of the work item effort.	No	N/A
workItemName	String	String value of the work item name.	No	N/A
workItemTypeCode	String	String value of work item type code. Valid values: <ul style="list-style-type: none"> • REQUEST • TASK • PROJECT • PACKAGE • MISC 	No	N/A
workItemID	String	String value of the work item ID.	No	N/A
activityName	String	String value of an activity name.	No	N/A
chargeCode	String	String value of a charge code name.	No	N/A

Property	Type	Description	Required	Default
Hours	String	String value of amount of hours for this work item.	No	N/A
Resource	String	String value of the resource username.	No	N/A
timeSheetLineStatus	String	String value of the statuses of the time sheet lines. Must correspond to the constants in code field in "TimeSheetLineStatus" on page 360.	No	N/A
Cost	String	String value of the cost of the effort.	No	N/A

Operations

The following operations are included in Time Management Web services:

- ["createTimeSheet"](#) on the next page
- ["updateTimeSheet"](#) on page 386
- ["getTimeSheet"](#) on page 395
- ["searchTimeSheets"](#) on page 397
- ["getActualTime"](#) on page 400
- ["getTimeSheetPolicy"](#) on page 403
- ["submitTimeSheet"](#) on page 407
- ["approveTimeSheet"](#) on page 411
- ["approveTimeSheetLine"](#) on page 415
- ["rejectTimeSheet"](#) on page 420
- ["rejectTimeSheetLine"](#) on page 424
- ["reworkTimeSheetLine"](#) on page 429
- ["freezeTimeSheet"](#) on page 434
- ["closeTimeSheet"](#) on page 438
- ["cancelTimeSheet"](#) on page 441

createTimeSheet

Purpose

Create a new time sheet, including time sheet lines in PPM Center.

Note: This operation supports the Day in Days option and Period in Days option, which were introduced since PPM Center version 8.00 Service Pack 1. For more information, see the Time Sheet Update Service section in *Project and Portfolio Management Center Version 8.00.01 (Service Pack 1) Release Notes*.

Time Management Web services is not built for the purpose of migrating large volume of time sheets data from a legacy system into PPM Center. To migrate large volume of time sheets, use TM Bulk Importer.

Creating a time sheet or any other operation is expected to take just slightly less than the same operation in the PPM application. Migration of many time sheets takes too much time with the current web services.

Function

This operation creates a new time sheet in PPM Center.

The user performing this operation must have the permission to create this particular time sheet, meaning that the user must meet the following conditions:

- Have the Time Mgmt: Edit Time Sheets access grant.
- Be one of the following:
 - The resource of this time sheet.
 - The delegate of the resource.
 - The manager of the resource.

The operation can do the following:

- Create a time sheet.
- Attach a time sheet policy to this time sheet.
- Enter effort for Period in Hours, Period In Days, or Period In Percent.
- Create time sheet lines in the time sheet.

The system creates audit information throughout the whole create and update process.

Once the new time sheet is created, the system performs the following functionalities:

- The status of the time sheet is "unsubmitted."
- The possible time and billing approvers are reevaluated.

This process fills the following fields:

- timeApproverParticipantGroupId
 - timeApproverSecurityGroupId
 - timeApproverUsersIds
 - billingApproverParticipantGroupId
 - billingApproverSecurityGroupId
 - billingApproverUsersIds
- The actual effort set to Task or Request work items is synched with those Tasks or Requests in Project Management or Demand Management accordingly.

The following field is set only by the functionality of this operation, and cannot be set by the user:

- Sequence: The sequence number set in the new time sheet would be the next sequence number available, meaning the system automatically increments the number of time sheets that exist for this resource and period by 1.

If the policy does not allow multiple time sheets per resource and period, and a time sheet has already existed for this resource and period, an exception (`ex.timesheetExist`) is thrown.

Input

TimeSheetBean

The following fields are must be set to complete this operation:

- resourceId: Existing Resource ID in PPM Center for whom you log the time sheet.
- periodId: Existing Time Period ID in PPM Center with which you log the time period.

Settings for the other fields depend on the functionality you want to achieve. For details, refer to the following:

- ["Create a Time Sheet" below](#)
- ["Attach a Time Sheet Policy" below](#)
- ["Create a Time Sheet Line" on the next page](#)

Create a Time Sheet

To create a time sheet, set the following fields.

- description: If the description field is not set, a default description is given, composed of Resource name - Time Period name and #Sequence.

Note: After the creation, the time sheet status (the state field) is set to the default value "unsubmitted."

Attach a Time Sheet Policy

To attach a time sheet policy, set the following fields.

- Policy fields

(bucketShowLevel, bucketReportMethod, periodHoursCalcTypeCode, hoursInPeriodType, hoursPerPeriod, workdayHours, maximumHoursPerDay, maximumHoursPerTimeSheet, minimumHoursPerTimeSheet, maximumPercentPerTimeSheet, minimumPercentPerTimeSheet, allowMultipleTimeSheets, notifyDelinquentTimeSheet, reqActivitiesRequiredFlag, tskActivitiesRequiredFlag, prjActivitiesRequiredFlag, pkgActivitiesRequiredFlag, miscActivitiesRequiredFlag, reqWorkItemEnabledFlag, tskWorkItemEnabledFlag, prjWorkItemEnabledFlag, pkgWorkItemEnabledFlag, mscWorkItemEnabledFlag, enforcementLevelError)

The BucketShowLevel field is used to determine if policy fields are set. If BucketShowLevel field is not set, the system considers you specify no policy fields.

- If no policy fields are specified, the policy set for this time sheet is the policy associated with the resource. If no policy is associated with the resource, the derived policy is based on the global

policy.

- If policy fields are specified, then those fields are set as the policy fields in the time sheet.

In both cases, the Time Period Type ID of the policy must match the Period Type ID associated with the Time Period ID which is a required parameter in this operation. All of the fields must be set in order to avoid confusion with the default policy fields that would be set on initial creation.

Create a Time Sheet Line

To create a time sheet line, set the following fields:

- timeSheetLines:
 - Each new time sheet line must have the following required fields set:
 - workItemId
 - workItemSetId

For the other TimeSheetLineBean fields regarding a time sheet line, refer to the following.

It is recommended that you update the time sheet line status by using the ["approveTimeSheetLine" on page 415](#), ["rejectTimeSheetLine" on page 424](#), and ["reworkTimeSheetLine" on page 429](#) operations.

To add Charge Codes in a time sheet line, set the following field:

- chargeCodesList: Each charge code parameter must have an ID of an existing Charge Code in PPM Center, and a percentage attached to it.

Note: The percentages of the charge codes in the charge code list should sum up to 100.

Note: After the creation, the time sheet line status (the state field) is set to the default value "unsubmitted." To update the time sheet line status, use the ["approveTimeSheetLine" on page 415](#), ["rejectTimeSheetLine" on page 424](#), and ["reworkTimeSheetLine" on page 429](#) operations.

To input actual effort in the time sheet line, set the following field.

- timeActualsList

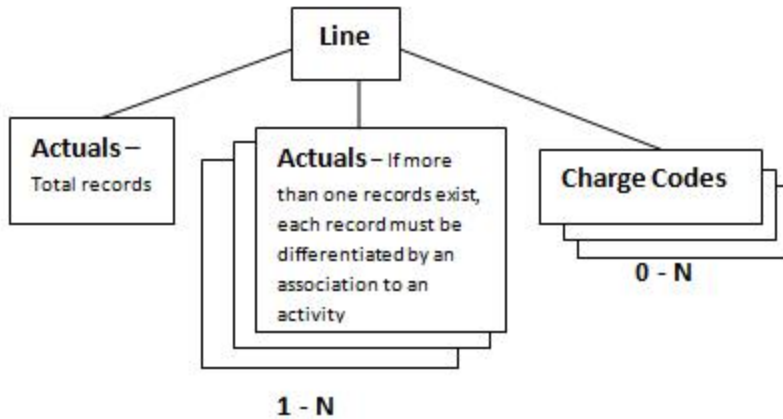
For each line, there must be at least one non-total record. Records with the total flag set to true are ignored.

Non-total records:

There can be one or more non-total records. If there is one non-total record, it may or may not be associated with an activity. If there are more than one non-total records, each record must be associated with an Activity ID.

When data is added to each non-total record, the totals record is automatically recalculated on the update of the time sheet object.

These relationships are shown in the following diagrams.



A line with one non-total actuals record, which is not associated with an activity:

Time Sheet Details (All times shown in hours)		Time Breakdown		Other Actuals						
Item	Activity	Expected Hours	Mon 1/14	Tue 1/15	Wed 1/16	Thu 1/17	Fri 1/18	Sat 1/19	Sun 1/20	Total
<input type="checkbox"/> gur project 2										
<input type="checkbox"/> Prj gur project 2		8.0	2.00	0.00	0.00	0.00	0.00	0.00	0.00	2.00
Line Actions:			2.00	0.00	0.00	0.00	0.00	0.00	0.00	2.00
<input type="button" value="Line Details"/> <input type="button" value="Remove"/> <input type="button" value="Rework"/>										

A line with one non-total actuals record, which is associated with one activity:

Time Sheet Details (All times shown in hours)				Time Breakdown		Other Actuals				
Item	Activity	Expected Hours	Mon 1/14	Tue 1/15	Wed 1/16	Thu 1/17	Fri 1/18	Sat 1/19	Sun 1/20	Total
<input type="checkbox"/> gur project 2										
<input type="checkbox"/> Prj: gur project 2	gur activity 1	8.0	2.00	0.00	0.00	0.00	0.00	0.00	0.00	2.00
Line Actions:			2.00	0.00	0.00	0.00	0.00	0.00	0.00	2.00
<input type="button" value="Line Details"/> <input type="button" value="Remove"/> <input type="button" value="Rework"/>										

A line with several non-total actuals records, where each is associated with one activity:

Time Sheet Details (All times shown in hours)				Time Breakdown		Other Actuals				
Item	Activity	Expected Hours	Mon 1/14	Tue 1/15	Wed 1/16	Thu 1/17	Fri 1/18	Sat 1/19	Sun 1/20	Total
<input type="checkbox"/> gur project 2										
<input type="checkbox"/> Prj: gur project 2		8.0	5.00	0.00	0.00	0.00	0.00	0.00	0.00	5.00
	gur activity 1		2.00	0.00	0.00	0.00	0.00	0.00	0.00	
	gur activity 2		3.00	0.00	0.00	0.00	0.00	0.00	0.00	
Line Actions:			5.00	0.00	0.00	0.00	0.00	0.00	0.00	5.00
<input type="button" value="Line Details"/> <input type="button" value="Remove"/> <input type="button" value="Rework"/>										

The following are additional fields that can be set for a time sheet line:

- updateChargeCodeFlag: Flag that defines whether charge codes can be updateable (Y/N).
- requireChargeCodeFlag: Flag that defines whether charge codes are required (Y/N).
- userDataBean: An array of up to 20 user data hidden values and a corresponding array of up to 20 user data visible values.

Return

An object of TimeSheetBean with updated fields. The data is returned as per policy.

Limitations

- All data, specified in the properties, must comply with what is expected in PPM Center - no data validation is performed by this Web service operation.
- Notes are not implemented in this Web service operation.

Related Information

["updateTimeSheet" on page 386](#) updates fields of an existing time sheet.

Java Interface

```
CreateTimeSheetResponseDocument createTimeSheet(CreateTimeSheetDocument in)
```

Parameters	Description
CreateTimeSheetDocument	<p>Wrapper for TimeSheetBean. See the following example for the construction. The bean includes:</p> <ul style="list-style-type: none">• Long timeSheetId;• Long resourceId;• Long periodId;• String description;• Integer sequence;• TimeSheetStatus state; // constants correspond to those in model.TimeSheet• List timeSheetLines; // list of TimeSheetLineBean objects• String bucketShowLevel ^a;• String bucketReportMethod;^a• String periodHoursCalcTypeCode;^a• Double hoursPerPeriod;^a• Double workdayHours;^a

Parameters	Description
	<ul style="list-style-type: none"> • Boolean allowMultipleTimeSheets;^a • String hoursInPeriodType;^a • Double maximumHoursPerDay;^a • Double maximumHoursPerTimeSheet;^a • Double minimumHoursPerTimeSheet;^a • Double maximumPercentPerTimeSheet;^a • Double minimumPercentPerTimeSheet;^a • Boolean notifyDelinquentTimeSheet;^a • Boolean reqActivitiesRequiredFlag;^a • Boolean tskActivitiesRequiredFlag;^a • Boolean prjActivitiesRequiredFlag;^a • Boolean pkgActivitiesRequiredFlag;^a • Boolean miscActivitiesRequiredFlag;^a • Boolean reqWorkItemEnabledFlag;^a • Boolean tskWorkItemEnabledFlag;^a • Boolean prjWorkItemEnabledFlag;^a • Boolean pkgWorkItemEnabledFlag;^a • Boolean mscWorkItemEnabledFlag;^a • Boolean enforcementLevelError;^a • String exceptionMessage ^b; • Long timeApproverParticipantGroupId;^b • Long timeApproverSecurityGroupId;^b • List timeApproverUsersIds;^b // list of Long user ids • Long billingApproverParticipantGroupId;^b • Long billingApproverSecurityGroupId;^b • List billingApproverUsersIds; ^b// list of Long user ids
CreateTimeSheetResponseDocument	Wrapper for TimeSheetBean. See the following example for retrieving the bean and its fields.
<p>a. Policy fields - for creation only</p> <p>b. Read-only fields</p>	

Java Examples

Example: create a new time sheet.

```
/**
 *
 * We are assuming existence of basic data that must exist in
 the app. Meaning:
 * - admin resource with user id 1
 * - time periods starting with period id 30000, at least one
 * - base time policy named 'Semi-Monthly - Day - Hours'
 *
 */

    TimeSheetBean createdTimeSheetBean = null;
    public void testCreateTimeSheet() {
        System.out.println("testCreateTimeSheet started ...");
        try {
            TimeServiceStub stub = new TimeServiceStub(ctx,
WSURL);
            CreateTimeSheetDocument createTimeSheetDocument =
CreateTimeSheetDocument.Factory.newInstance();
            final TimeSheetBean newTimeSheetBean =
createTimeSheetDocument.addNewCreateTimeSheet().addNewTimeSheet
Bean();

            newTimeSheetBean.setResourceId(1); // assuming admin
user exists with id 1
            newTimeSheetBean.setPeriodId(30000); // assuming
this time period exists

            newTimeSheetBean.setBucketShowLevel("DAY");
            newTimeSheetBean.setBucketReportMethod("HOURS");

newTimeSheetBean.setPeriodHoursCalcTypeCode("WORKDAY");
            newTimeSheetBean.setHoursPerPeriod(8);
            newTimeSheetBean.setWorkdayHours(5.5);
            newTimeSheetBean.setReqActivitiesRequiredFlag(true);
            newTimeSheetBean.setTskActivitiesRequiredFlag(true);
            newTimeSheetBean.setPkgActivitiesRequiredFlag(true);

newTimeSheetBean.setMiscActivitiesRequiredFlag(true);
            newTimeSheetBean.setAllowMultipleTimeSheets(true);
            newTimeSheetBean.setEnforcementLevelError(true);

            TimeSheetLineBean timeSheetLineBean =
newTimeSheetBean.addNewTimeSheetLines();
            timeSheetLineBean.setWorkItemId("2");
            timeSheetLineBean.setWorkItemSetId("1");
```

```
        timeSheetLineBean.setWorkItemType("MISC");
        TimeSheetLineStatus timeSheetLineStatus =
timeSheetLineBean.addNewState();
        timeSheetLineStatus.setCode(new BigInteger("1"));
        timeSheetLineStatus.setMeaning("unsubmitted");

        ChargeCodeBean chargeCodeBean =
timeSheetLineBean.addNewChargeCodesList();
        chargeCodeBean.setChargeCodeId(30000);
        chargeCodeBean.setChargeCodeName("my cc name");
        chargeCodeBean.setCategory("BILLABLE");
        chargeCodeBean.setClientCode("CLIENT_1");
        chargeCodeBean.setDepartmentCode("FINANCE");
        chargeCodeBean.setDescription("my cc name");
        chargeCodeBean.setEnabledFlag(true);
        chargeCodeBean.setPercentage(100.d);

        timeSheetLineBean.setUpdateChargeCodeFlag(true);
        timeSheetLineBean.setRequireChargeCodeFlag(false);
        timeSheetLineBean.setActualApproverUserId(1);
        UserDataInfo ud =
timeSheetLineBean.addNewUserDataBean();
        ud.setUserData1("UD1");
        ud.setVisUserData1("VisUD1");

        TimeActualsBean timeActualsBean1 =
timeSheetLineBean.addNewTimeActualsList();
        timeActualsBean1.setTotalsFlag(true);
        timeActualsBean1.setEffortsListArray(new double[]
{2,4,6,8,10,0,0,2,4,6,8,10,0,0,20});

        TimeActualsBean timeActualsBean2 =
timeSheetLineBean.addNewTimeActualsList();
        timeActualsBean2.setTotalsFlag(false);
        timeActualsBean2.setEffortsListArray(new double[]
{1,2,3,4,5,0,0,1,2,3,4,5,0,0,10});

        TimeActualsBean timeActualsBean3 =
timeSheetLineBean.addNewTimeActualsList();
        timeActualsBean3.setTotalsFlag(false);
        timeActualsBean2.setActivityId(30000);
        timeActualsBean3.setEffortsListArray(new double[]
{1,2,3,4,5,0,0,1,2,3,4,5,0,0,10});

        CreateTimeSheetResponseDocument response =
stub.createTimeSheet(createTimeSheetDocument);
        createdTimeSheetBean =
response.getCreateTimeSheetResponse().getReturn();
        System.out.println("Newly created TimeSheet with id
" + createdTimeSheetBean.getTimeSheetId());
```

```

    }
    catch(Exception e) {
        e.printStackTrace();
    }
    System.out.println("testCreateTimeSheet completed");
}
    
```

Errors and Exceptions

When an error occurs during this operation, a description of the root cause will be logged and a response message generated.

The server log file content is similar to the following:

Exception in thread "main" org.apache.axis2.AxisFault: [root cause description]

Response message:

```

<exception:exceptionDetails xmlns:exception="http://
www.mercury.com/ppm/ws/exception">
<exception:detail>[root cause description] </exception:detail>
</exception:exceptionDetails>
    
```

Possible root cause descriptions:

Message Code	Message	Cause(s)	Possible Corrective Action
ex.cannotEditTimesheet	User cannot edit Timesheet	You may not have the required access grant or Time Management License.	Check your licenses and privileges (access grants).
ex.resourceNotExist	Resource does not exist	Resource ID is invalid or resource does not exist anymore in PPM Center.	Check the validity of the resource ID.
ex.periodNotExist	Time Period does not exist	Time Period ID is invalid or Time Period ID does not exist anymore in the	Check the validity of the time period ID.

Message Code	Message	Cause(s)	Possible Corrective Action
		database.	
ex.globalPolicyNotExist	Global timesheet policy does not exist	Corrupt database.	Check for an existence of the global policy; must exist in PPM Center.
ex.periodTypeNotMatching	Period type in time period does not match period type in timesheet policy	Invalid Time Period data.	Check for the correspondence between the time period and the period type of the policy.
ex.timesheetExist	Timesheet already exists for this resource and period, policy does not allow multiple timesheets	You try to create multiple time sheets when the policy does not allow it.	Verify that either the policy allows multiple time sheets or a time sheet does not already exist for the period and resource for which you are trying to create a new time sheet.
ex.workitemInvalid	Timesheet line could not be created, item added is not valid for this resource	Invalid data in the new time sheet line. It could be caused by a certain field or one of the following these causes: <ul style="list-style-type: none"> • Work item ID does not exist in the type. • Work item ID does not exist in the work item set ID. 	If work item type is 'PROJECT' or 'TASK', verify that the resource has access to the work item ID; or if the work item type is 'REQUEST', verify that the work item set ID is the request type ID.
ex.chargeCodeNotExist	Charge code does not exist	Charge code ID is invalid or charge code	Check the validity of the charge code ID.

Message Code	Message	Cause(s)	Possible Corrective Action
		does not exist anymore in the database.	
ex.chargeCodePercentagesMustSum100	Charge codes percentages must sum to 100	For each line, a list of charge codes can be defined. Each charge code must have a percentage. The sum of all the percentages per each line must sum to 100.	Check the sum of all charge codes percentages for each line.
ex.MissingTimeActuals	Incorrect number of Effort Data in Time Actuals	For each actual list, Web service checks the number of items with the calendar date.	Check the number of items in the actual list.

updateTimeSheet

Purpose

Update an existing time sheet in PPM Center.

Note: This operation supports the Day in Days option and Period in Days option, which were introduced since PPM Center version 8.00 Service Pack 1. For more information, see the Time Sheet Update Service section in *Project and Portfolio Management Center Version 8.00.01 (Service Pack 1) Release Notes*.

Function

This operation updates an unsubmitted time sheet object, identified by time sheet ID in PPM Center.

The user performing this operation must have the permission to update this particular time sheet, meaning that the user must meet one of the following conditions:

- Have the Time Mgmt: Edit Time Sheets access grant.
- Be one of the following:
 - The resource of this time sheet
 - The delegate of the resource
 - The manager of the resource

This operation can do the following:

- Update the description of the time sheet.
- Delete the time sheet line.
Note: Missing unsubmitted timesheet line will be removed.
- Add the time sheet line. The status of the newly-added line is "unsubmitted."
- Update the time sheet lines.

The following fields cannot be updated through this operation.

- Resource
- Time Period
- Policy fields
- Sequence
- Time sheet status
- Time sheet line status

The system creates audit information throughout the whole process.

Once the time sheet is updated, the system performs the following functionalities:

- The system reevaluates the status of the time sheet according to the statuses of its lines.
- The actual effort set to Task or Request work items is synchronized with those Tasks or Requests in Project Management or Demand Management accordingly.

Input

An object of TimeSheetBean that must have the following fields:

- timeSheetID: Existing time sheet ID in PPM Center.

Settings for the other fields depend on the functionality you want to achieve. For details, refer to the following:

- ["Update the Description for a Time Sheet" below](#)
- ["Delete a Time Sheet Line" below](#)
- ["Add a new Time Sheet Line" below](#)

Update the Description for a Time Sheet

To update the description for a time sheet, set the following field.

- description: If the description field is null or blank, the time sheet keeps the original description.

Delete a Time Sheet Line

To delete a specific time sheet line, remove the TimeSheetLineBean together with the specific line ID from TimeSheetBean.

Add a new Time Sheet Line

To add a new time sheet line, refer to the ["Create a Time Sheet Line" on page 377](#) section in operation ["createTimeSheet" on page 374](#).

Important: If you want to update a time sheet or a time sheet line within the existing TimeSheetBean, you have to input the whole TimeSheetBean. For example, if the time sheet contains 100 lines, you should input all 100 lines using Web service, not only the one that you want to update. Otherwise, all the lines that you did not input will be missing from the time sheet.

Return

The updated TimeSheetBean.

Limitations

- All data, specified in the properties, must comply with what is expected in PPM Center, no data validation is performed by the Web service operation.
- Notes are not implemented in this Web service operation.

Java Interface

UpdateTimeSheetResponseDocument updateTimeSheet(UpdateTimeSheetDocument in)

Parameters	Description
UpdateTimeSheetDocument	<p>Wrapper for TimeSheetBean. See the following example for the construction. The bean includes:</p> <ul style="list-style-type: none">• Long timeSheetId;• Long resourceId;• Long periodId;• String description;• Integer sequence;• TimeSheetStatus state; // constants correspond to those in model.TimeSheet• List timeSheetLines; // list of TimeSheetLineBean objects• String bucketShowLevel^a;• String bucketReportMethod;^a• String periodHoursCalcTypeCode;^a• Double hoursPerPeriod;^a

Parameters	Description
	<ul style="list-style-type: none"> • Double workdayHours;^a • Boolean allowMultipleTimeSheets;^a • String hoursInPeriodType;^a • Double maximumHoursPerDay;^a • Double maximumHoursPerTimeSheet;^a • Double minimumHoursPerTimeSheet;^a • Double maximumPercentPerTimeSheet;^a • Double minimumPercentPerTimeSheet;^a • Boolean notifyDelinquentTimeSheet;^a • Boolean reqActivitiesRequiredFlag;^a • Boolean tskActivitiesRequiredFlag;^a • Boolean prjActivitiesRequiredFlag;^a • Boolean pkgActivitiesRequiredFlag;^a • Boolean miscActivitiesRequiredFlag;^a • Boolean reqWorkItemEnabledFlag;^a • Boolean tskWorkItemEnabledFlag;^a • Boolean prjWorkItemEnabledFlag;^a • Boolean pkgWorkItemEnabledFlag;^a • Boolean mscWorkItemEnabledFlag;^a • Boolean enforcementLevelError;^a • String exceptionMessage;^b • Long

Parameters	Description
	timeApproverParticipantGroupId; ^b <ul style="list-style-type: none"> • Long timeApproverSecurityGroupId;^b • List timeApproverUsersIds;^b // list of Long user ids • Long billingApproverParticipantGroupId;^b • Long billingApproverSecurityGroupId;^b • List billingApproverUsersIds; ^b// list of Long user ids
UpdateTimeSheetResponseDocument	Wrapper for TimeSheetBean. See the following example for retrieving the bean and its fields.
a. Policy fields - for creation only b. Read-only fields	

Java Examples

Example: update an existing time sheet.

```

TimeSheetBean createdTimeSheetBean = null;
public void testUpdateTimeSheet() {
    System.out.println("testUpdateTimeSheet started ...");
    try {
        if(createdTimeSheetBean == null)
            testCreateTimeSheet();

        TimeServiceStub stub = new TimeServiceStub(ctx,
WSURL);
        UpdateTimeSheetDocument updateTimeSheetDocument =
UpdateTimeSheetDocument.Factory.newInstance();
        final TimeSheetBean updateTimeSheetBean =
updateTimeSheetDocument.addNewUpdateTimeSheet().addNewTimeSheet
Bean();

        // set required fields

updateTimeSheetBean.setTimeSheetId(createdTimeSheetBean.getTime
SheetId());
  
```

```
updateTimeSheetBean.setResourceId(createdTimeSheetBean.getResourceId());

updateTimeSheetBean.setPeriodId(createdTimeSheetBean.getPeriodId());

        // set all other fields
        updateTimeSheetBean.setDescription("my desc");
        TimeSheetStatus timeSheetStatus =
updateTimeSheetBean.addNewState();
        timeSheetStatus.setCode(new BigInteger("1"));
        timeSheetStatus.setMeaning("unsubmitted");
        TimeSheetLineBean timeSheetLineBean =
updateTimeSheetBean.addNewTimeSheetLines();

        // set required fields
timeSheetLineBean.setTimeSheetLineId(createdTimeSheetBean.getTimeSheetLinesArray(0).getTimeSheetLineId());

        timeSheetLineBean.setWorkItemId("2");
        timeSheetLineBean.setWorkItemSetId("1");
        timeSheetLineBean.setWorkItemType("MISC"); // TASK,
REQUEST, PACKAGE, MISC
        TimeSheetLineStatus timeSheetLineStatus =
timeSheetLineBean.addNewState();
        timeSheetLineStatus.setCode(new BigInteger("1"));
        timeSheetLineStatus.setMeaning("unsubmitted");

        UserDataInfo userDataInfo =
timeSheetLineBean.addNewUserDataBean();
        userDataInfo.setUserData1("my userdata1");
        userDataInfo.setVisUserData1("my visible
userdata1");

        TimeActualsBean timeActualsBean1 =
timeSheetLineBean.addNewTimeActualsList();
        timeActualsBean1.setTotalsFlag(true);
        timeActualsBean1.setEffortsListArray(new double[]
{1,2,3,4,5,0,0});
        TimeActualsBean timeActualsBean2 =
timeSheetLineBean.addNewTimeActualsList();
        timeActualsBean2.setTotalsFlag(false);
        timeActualsBean2.setEffortsListArray(new double[]
{1,2,3,4,5,0,0});

        timeActualsBean2.setEffortsListArray(new double[]
{1,2,3,4,5,0,0});

        UpdateTimeSheetResponseDocument response =
stub.updateTimeSheet(updateTimeSheetDocument);
        TimeSheetBean returnTimeSheetBean =
response.getUpdateTimeSheetResponse().getReturn();
```



```
        System.out.println("Time Sheet was updated " +  
returnTimeSheetBean.getTimeSheetId());  
    }  
    catch(Exception e) {  
        e.printStackTrace();  
    }  
    System.out.println("testUpdateTimeSheet completed");  
}
```

Errors and Exceptions

When an error occurs on this operation, you will receive a description of the root cause in the log or in the response message.

The server log file content is similar to the following:

```
Exception in thread "main" org.apache.axis2.AxisFault: [root cause description]
```

Response message:

```
<exception:exceptionDetails xmlns:exception="http://  
www.mercury.com/ppm/ws/exception">  
<exception:detail>[root cause description] </exception:detail>  
</exception:exceptionDetails>
```

Possible root cause descriptions:

Message Code	Message	Cause(s)	Possible Corrective Action
ex.cannotEditTimesheet	User cannot edit Timesheet	You may not have the required access grant or Time Management License.	Check the user licenses and privileges (access grants).
ex.timesheetNotExist	Timesheet does not exist	Time Sheet ID is invalid (Cancelled time sheets must still be kept in the database).	Check the validity of the time sheet ID.
ex.workitemInvalid	Timesheet line could not be	Invalid data in the new time sheet line. It could be	If work item type is 'PROJECT' or 'TASK', check if the

Message Code	Message	Cause(s)	Possible Corrective Action
	created, item added is not valid for this resource	caused by a certain field or one of the following causes: <ul style="list-style-type: none"> • Work item ID does not exist in the type. • Work item ID does not exist in the work item set ID. 	resource has the access to the work item ID; or if the work item type is 'REQUEST', check if the work item set ID is the request type ID.
ex.chargeCodeNotExist	Charge code does not exist	Charge code ID is invalid or charge code does not exist anymore in the database.	Check the validity of the charge code ID.
ex.chargeCodePercentagesMustSum100	Charge codes percentages must sum to 100, but are {0}	For each line, a list of charge codes can be defined. Each charge code must have a percentage. The sum of all the percentages per each line must sum to 100.	Check the sum of all charge codes percentages for each line.
ex.MissingTimeActuals	Incorrect number of Effort Data in Time Actuals	For each actual list, Web service checks the number of items with the calendar date.	Check the number of items in the actual list.

getTimeSheet

Purpose

Read an existing time sheet object in PPM Center.

Note: This operation supports the Day in Days option and Period in Days option, which were introduced since PPM Center version 8.00 Service Pack 1. For more information, see the Time Sheet Update Service section in *Project and Portfolio Management Center Version 8.00.01 (Service Pack 1) Release Notes*.

Function

This operation reads an existing time sheet object, identified by time sheet ID in PPM Center.

The user performing this operation must have the permission to view this particular time sheet, meaning that the user must meet the following conditions:

- Have one of the following access grants:
 - Time Mgmt: View Time Sheets
 - Time Mgmt: Edit Time Sheets
- Be one of the following:
 - The resource of the time sheet
 - The delegate of the resource
 - The manager of the resource

Input

A long value of an ID of an existing time sheet.

Return

An object of `TimeSheetBean`, with all fields values completed. The data is returned according to the time sheet policy.

Java Interface

```
GetTimeSheetResponseDocument getTimeSheet(GetTimeSheetDocument in)
```

Parameters	Description
<code>GetTimeSheetDocument</code>	Wrapper for a long value which holds the <code>TimeSheetId</code> . See the following example for the construction.
<code>GetTimeSheetResponseDocument</code>	Wrapper for <code>TimeSheetBean</code> . See the following example for retrieving the bean and its fields.

Java Examples

Example: read an existing time sheet.

```
TimeSheetBean createdTimeSheetBean = null;
public void testGetTimeSheet() {
    System.out.println("testGetTimeSheet started ...");
    try {
        if(createdTimeSheetBean == null)
            testCreateTimeSheet();

        TimeServiceStub stub = new TimeServiceStub(ctx,
WSURL);
        GetTimeSheetDocument getTimeSheetDocument =
GetTimeSheetDocument.Factory.newInstance();

        getTimeSheetDocument.addNewGetTimeSheet().setTimeSheetId(create
dTimeSheetBean.getTimeSheetId());

        GetTimeSheetResponseDocument response =
stub.getTimeSheet(getTimeSheetDocument);
        final TimeSheetBean returnTimeSheetBean =
response.getGetTimeSheetResponse().getReturn();

        System.out.println("Time Sheet was found " +
returnTimeSheetBean.getTimeSheetId());
```

```
    }  
    catch(Exception e) {  
        e.printStackTrace();  
    }  
    System.out.println("testGetTimeSheet completed");  
}
```

Errors and Exceptions

When an error occurs on this operation, you will receive a description of the root cause in the log or in the response message.

The server log file content is similar to the following:

```
Exception in thread "main" org.apache.axis2.AxisFault: [root cause description]
```

Response message:

```
<exception:exceptionDetails xmlns:exception="http://  
www.mercury.com/ppm/ws/exception">  
<exception:detail>[root cause description] </exception:detail>  
</exception:exceptionDetails>
```

Possible root cause descriptions:

Message Code	Message	Cause(s)	Possible Corrective Action
ex.timesheetNotExist	Timesheet does not exist	Time sheet ID is invalid.	Check the validity of the time sheet ID.
ex.cannotEditTimesheet	User cannot edit Timesheet	You may not have the required access grant or Time Management License.	Check the user licenses and privileges (access grants).

searchTimeSheets

Purpose

Search for a list of time sheets in PPM Center by using specific criteria.

Note: This operation supports the Day in Days option and Period in Days option, which were introduced since PPM Center version 8.00 Service Pack 1. For more information, see the Time Sheet Update Service section in *Project and Portfolio Management Center Version 8.00.01 (Service Pack 1) Release Notes*.

Function

This operation searches for time sheets, according to the search criteria.

The user performing this operation must have the Time Mgmt: View Time Sheets access grant.

Input

TimeSheetSearchCriteriaBean

Return

A list of String values of the time sheet IDs.

Limitation

The maximum number of results cannot exceed 1,000.

Java Interface

```
SearchTimeSheetsResponseDocument searchTimeSheets(SearchTimeSheetsDocument in)
```

Parameters	Description
SearchTimeSheetsDocument	Wrapper for TimeSheetSearchCriteriaBean. See the following example for the construction. The criteria bean includes the following fields: <ul style="list-style-type: none">• Long periodTypeId;

Parameters	Description
	<ul style="list-style-type: none">• Long periodId;• Long previousTimePeriodsCount;• List resourceIdList; // list of Long resource ids• String descriptionKeywords;• List statusList; // list of Long, statuses from model.TimeSheet• List managerIdList; // list of Long user ids• List orgUnitIdList; // list of Long orgUnit ids
SearchTimeSheetsResponseDocument	Wrapper for an array of long values, each holding a time sheet ID. See the following example for retrieving the array.

Java Examples

Example: search for time sheets.

```
public void testSearchTimeSheets() {
    System.out.println("testSearchTimeSheets started ...");
    try {
        TimeServiceStub stub = new TimeServiceStub(ctx,
WSURL);
        SearchTimeSheetsDocument searchTimeSheetsDocument =
SearchTimeSheetsDocument.Factory.newInstance();
        TimeSheetSearchCriteriaBean
timeSheetSearchCriteriaBean =
searchTimeSheetsDocument.addNewSearchTimeSheets().addNewSearchC
riteriaBean();

timeSheetSearchCriteriaBean.setPeriodTypeId(tmPeriod.getPeriodT
ypeId().longValue());

timeSheetSearchCriteriaBean.setPeriodId(tmPeriod.getPeriodId().
longValue());

timeSheetSearchCriteriaBean.setPreviousTimePeriodsCount(10);

timeSheetSearchCriteriaBean.setResourceIdListArray(new
long[]{1}); // list of Long resource ids // admin

timeSheetSearchCriteriaBean.setDescriptionKeywords("description
");
        timeSheetSearchCriteriaBean.setStatusListArray(new
long[]{1}); // TimeSheet.STATUS_UNSUBMITTED from model.TimeSheet
```

```
        SearchTimeSheetsResponseDocument response =
stub.searchTimeSheets(searchTimeSheetsDocument);
        long[] returnArray =
response.getSearchTimeSheetsResponse().getTimeSheetIdArray();

        System.out.println("Time Sheet search brought " +
returnArray.length + " results");
    }
    catch(Exception e) {
        e.printStackTrace();
    }
    System.out.println("testSearchTimeSheets completed");
}
```

Errors and Exceptions

N/A

getActualTime

Purpose

Read actual time data of the work items between a specific time range in PPM Center.

Function

This operation filters for actual data of the work items in Time Management in PPM Center.

The user performing this operation must have the Time Mgmt: View Time Sheets access grant.

Input

An object of TimeFilter, which holds all the fields described in the TimeFilter data type.

Return

An array of `WorkItemActualTime` objects.

Java Interface

```
GetActualTimeResponseDocument getActualTime(GetActualTimeDocument in)
```

Parameters	Description
<code>GetActualTimeDocument</code>	<p>Wrapper for a <code>TimeFilter</code> object.</p> <p>See the following example for the construction.</p> <p>The <code>TimeFilter</code> object includes the following fields:</p> <ul style="list-style-type: none">• Date <code>startDate</code>• Date <code>endDate</code>• String <code>periodTypeName</code>• String <code>workItemType</code>• String <code>workItemName</code>• String <code>workItemID</code>• List <code>resources</code>• List <code>timeSheetStatuses</code>• List <code>timeSheetLineStatuses</code>• boolean <code>includeActualCost</code>
<code>GetActualTimeResponseDocument</code>	<p>Wrapper for an array of <code>WorkItemActualTime</code> objects.</p> <p>See the following example for retrieving the array.</p> <p>Each <code>WorkItemActualTime</code> includes the following fields:</p> <ul style="list-style-type: none">• Date <code>startDate</code>• Date <code>endDate</code>• String <code>workItemName</code>• String <code>workItemTypeCode</code>• String <code>workItemID</code>

Parameters	Description
	<ul style="list-style-type: none">• String activityName• String chargeCode• String hours• String resource• String timeSheetLineStatus• String cost

Java Examples

Example: get actual time data.

```
public void testGetActualTime() throws Exception {
    System.out.println("testGetActualTime started ...");

    // get stub
    TimeServiceStub stub = new TimeServiceStub(ctx, WSURL);

    // create input message
    GetActualTimeDocument getActualTimeDoc =
    GetActualTimeDocument.Factory.newInstance();

    // set filter
    TimeFilter filter =
    getActualTimeDoc.addNewGetActualTime().addNewParam0();

    // setup the filters
    Calendar startDate = Calendar.getInstance();
    Calendar endDate = Calendar.getInstance();
    endDate.add(Calendar.DATE, 14);
    filter.setStartDate(startDate);
    filter.setEndDate(endDate);
    filter.setPeriodTypeName("Semi-Monthly");
    filter.setIncludeActualCost(true);
    filter.addResources("admin");

    // invoke the service
    GetActualTimeResponseDocument response =
    stub.getActualTime(getActualTimeDoc);

    // process response
    System.out.println("Read " +
    response.getGetActualTimeResponse().sizeOfReturnArray() + "
    work item rows");
    WorkItemActualTime[] workItems =
    response.getGetActualTimeResponse().getReturnArray();
}
```

```
    if (workItems.length > 0) {  
        WorkItemActualTime workItem = workItems[0];  
        System.out.println("total hours for misc. item = " +  
workItem.getHours());  
    }  
        System.out.println("testGetActualTime completed");  
    }
```

Errors and Exceptions

When an error occurs on this operation, you will receive a description of the root cause in the log or in the response message.

The server log file content is similar to the following:

```
Exception in thread "main" org.apache.axis2.AxisFault: [root cause description]
```

Response message:

```
<exception:exceptionDetails xmlns:exception="http://  
www.mercury.com/ppm/ws/exception">  
<exception:detail>[root cause description] </exception:detail>  
</exception:exceptionDetails>
```

Possible root cause descriptions:

Message Code	Message	Cause(s)	Possible Corrective Action
ex.cannotViewTimesheet	User cannot view Timesheet	You may not have the required access grant or Time Management License.	Check the user licenses and privileges (access grants).

getTimeSheetPolicy

Purpose

Read an existing Time Sheet Policy object in PPM Center.

Note: This operation supports the Day in Days option and Period in Days option, which were

introduced since PPM Center version 8.00 Service Pack 1. For more information, see the Time Sheet Update Service section in *Project and Portfolio Management Center Version 8.00.01 (Service Pack 1) Release Notes*.

Function

This operation reads an existing Time Sheet Policy object, identified by the unique time sheet policy name in PPM Center.

Input

A String value of the name of an existing time sheet policy

Return

An object of TimeSheetPolicyBean

Java Interface

```
GetTimeSheetPolicyResponseDocument getTimeSheetPolicy(GetTimeSheetPolicyDocument in)
```

Parameters	Description
GetTimeSheetPolicyDocument	Wrapper for a String value which holds the policy name. See the following example for the construction.
GetTimeSheetPolicyResponseDocument	Wrapper for a TimeSheetPolicyBean. See the following example for retrieving the bean and its fields. Each bean includes the following fields: <ul style="list-style-type: none">• kkkkLong timeSheetPolicyId• kkkkString timeSheetPolicyName• String bucketShowLevel

Parameters	Description
	<ul style="list-style-type: none">• String bucketReportMethod• String periodHoursCalcTypeCode• Double hoursPerPeriod• Double workdayHours• Boolean allowMultipleTimeSheets• String hoursInPeriodType• Double maximumHoursPerDay• Double maximumHoursPerTimeSheet• Double minimumHoursPerTimeSheet• Double maximumPercentPerTimeSheet• Double minimumPercentPerTimeSheet• Boolean notifyDelinquentTimeSheet• Boolean reqActivitiesRequiredFlag• Boolean tskActivitiesRequiredFlag• Boolean prjActivitiesRequiredFlag• Boolean pkgActivitiesRequiredFlag• Boolean miscActivitiesRequiredFlag• Boolean reqWorkItemEnabledFlag• Boolean tskWorkItemEnabledFlag• Boolean prjWorkItemEnabledFlag• Boolean pkgWorkItemEnabledFlag• Boolean mscWorkItemEnabledFlag• Boolean enforcementLevelError

Java Examples

Example: read an existing time sheet policy.

```
public void testGetTimeSheetPolicy() {  
    System.out.println("testGetTimeSheetPolicy started  
...");  
    try {
```

```
        TimeServiceStub stub = new TimeServiceStub(ctx,
WSURL);

        GetTimeSheetPolicyDocument getTimeSheetPolicyDocument
= GetTimeSheetPolicyDocument.Factory.newInstance();

getTimeSheetPolicyDocument.addNewGetTimeSheetPolicy().setPolicy
Name("Semi-Monthly - Day - Hours");

        GetTimeSheetPolicyResponseDocument response =
stub.getTimeSheetPolicy(getTimeSheetPolicyDocument);
        final TimeSheetPolicyBean returntimesheetPolicy =
response.getGetTimeSheetPolicyResponse().getReturn();

        System.out.println("Time Sheet Policy was found " +
returntimesheetPolicy.getTimeSheetPolicyId());
    }
    catch(Exception e) {
        e.printStackTrace();
    }
    System.out.println("testGetTimeSheetPolicy completed");
}
```

Errors and Exceptions

When an error occurs on this operation, you will receive a description of the root cause in the log or in the response message.

The server log file content is similar to the following:

```
Exception in thread "main" org.apache.axis2.AxisFault: [root cause description]
```

Response message:

```
<exception:exceptionDetails xmlns:exception="http://
www.mercury.com/ppm/ws/exception">
<exception:detail>[root cause description] </exception:detail>
</exception:exceptionDetails>
```

Possible root cause descriptions:

Message Code	Message	Cause(s)	Possible Corrective Action
ex.policyNotExist	Policy does not exist with name {0}	Policy does not exist, or corrupt database.	Check the validity of the policy name (must be unique in PPM Center)

* All the following operations are convenience helper methods to change the status of a time sheet or a time sheet line:

- ["submitTimeSheet" below](#)
- ["approveTimeSheet" on page 411](#)
- ["approveTimeSheetLine" on page 415](#)
- ["rejectTimeSheet" on page 420](#)
- ["rejectTimeSheetLine" on page 424](#)
- ["reworkTimeSheetLine" on page 429](#)
- ["freezeTimeSheet" on page 434](#)
- ["closeTimeSheet" on page 438](#)
- ["cancelTimeSheet" on page 441](#)

Since status change can occur only from a specific existing status, you must perform a validity check for this existing status first. If the existing status does not accord with the functionality you want to achieve (for example, you try to submit an approved time sheet), an exception is thrown.

Note: If you want to update a time sheet or a time sheet line within the existing TimeSheetBean, you must input the whole TimeSheetBean. For example, if time sheet contains 100 lines, you must input all 100 lines by using Web service, not only the one that you want to update. Otherwise, all the lines that you did not input will be missing from the time sheet.

submitTimeSheet

Purpose

Submit an existing Time Sheet object in PPM Center.

Note: This operation supports the Day in Days option and Period in Days option, which were introduced since PPM Center version 8.00 Service Pack 1. For more information, see the Time Sheet Update Service section in *Project and Portfolio Management Center Version 8.00.01 (Service Pack 1) Release Notes*.

Function

This operation updates the status of all the lines of an existing time sheet, and the status of the time sheet to "submitted."

Before this operation, the status of the time sheet must be "unsubmitted" or "in-rework." Additionally, you must set the billing approver or time approver for the resource of the time sheet.

During the submission, you can update any existing line item data. To submit all line items with out any change, include all lines or include none of the lines. If the operation has one or more lines included, any missing unsubmitted lines will be removed.

The user performing this operation must have the permission to submit this particular time sheet, meaning that the user must meet the following conditions:

- Have the Time Mgmt: Edit Time Sheets access grant.
- Be one of the following:
 - The resource of this time sheet.
 - The delegate of the resource.
 - The manager of the resource.

Input

An object of whole TimeSheetBean to submit.

Return

An object of TimeSheetBean.

Java Interface

```
SubmitTimeSheetResponseDocument submitTimeSheet(SubmitTimeSheetDocument in)
```


Parameters	Description
SubmitTimeSheetDocument	Wrapper for TimeSheetBean. See the following example for the construction.
SubmitTimeSheetResponseDocument	Wrapper for TimeSheetBean. See the following example on how to retrieve the bean.

Java Examples

Example: submit an existing time sheet.

```
TimeSheetBean createdTimeSheetBean = null;
public void testSubmitTimeSheet() {
    System.out.println("testSubmitTimeSheet started ...");
    try {
        if(createdTimeSheetBean == null)
            testCreateTimeSheet();

        TimeServiceStub stub = new TimeServiceStub(ctx,
WSURL);
        SubmitTimeSheetDocument submitTimeSheetDocument =
SubmitTimeSheetDocument.Factory.newInstance();
        TimeSheetBean submitTimeSheetBean =
submitTimeSheetDocument.addNewSubmitTimeSheet().addNewTimeSheet
Bean();

submitTimeSheetBean.setTimeSheetId(createdTimeSheetBean.getTime
SheetId());

submitTimeSheetBean.setResourceId(createdTimeSheetBean.getResou
rceId());

submitTimeSheetBean.setPeriodId(createdTimeSheetBean.getPeriodI
d());
        TimeSheetStatus timeSheetStatus =
submitTimeSheetBean.addNewState();
        timeSheetStatus.setCode(new BigInteger("1"));
        timeSheetStatus.setMeaning("unsubmitted");
        TimeSheetLineBean timeSheetLineBean =
submitTimeSheetBean.addNewTimeSheetLines();

timeSheetLineBean.setTimeSheetLineId(createdTimeSheetBean.getTi
meSheetLinesArray(0).getTimeSheetLineId());

timeSheetLineBean.setWorkItemId(createdTimeSheetBean.getTimeShe
etLinesArray(0).getWorkItemId());

timeSheetLineBean.setWorkItemSetId(createdTimeSheetBean.getTime
SheetLinesArray(0).getWorkItemSetId());
```

```
timeSheetLineBean.setWorkItemType(createdTimeSheetBean.getTimeS  
heetLinesArray(0).getWorkItemType());  
    TimeActualsBean timeActualsBean1 =  
timeSheetLineBean.addNewTimeActualsList();  
    timeActualsBean1.setTotalsFlag(true);  
    timeActualsBean1.setEffortsListArray(new double[]  
{1,2,3,4,5,0,0,1,2,3,4,5,0,0,10});  
  
    TimeActualsBean timeActualsBean2 =  
timeSheetLineBean.addNewTimeActualsList();  
    timeActualsBean2.setTotalsFlag(false);  
    timeActualsBean2.setEffortsListArray(new double[]  
{1,2,3,4,5,0,0,1,2,3,4,5,0,0,10});  
  
    SubmitTimeSheetResponseDocument response =  
stub.submitTimeSheet(submitTimeSheetDocument);  
    TimeSheetBean returnTimeSheet =  
response.getSubmitTimeSheetResponse().getReturn();  
  
    System.out.println("Time Sheet was submitted " +  
returnTimeSheet.getTimeSheetId());  
    }  
    catch(Exception e) {  
        e.printStackTrace();  
    }  
    System.out.println("testSubmitTimeSheet completed");  
}
```

Errors and Exceptions

When an error occurs on this operation, you will receive a description of the root cause in the log or in the response message.

The server log file content is similar to the following:

```
Exception in thread "main" org.apache.axis2.AxisFault: [root cause description]
```

Response message:

```
<exception:exceptionDetails xmlns:exception="http://  
www.mercury.com/ppm/ws/exception">  
<exception:detail>[root cause description] </exception:detail>  
</exception:exceptionDetails>
```

Possible root cause descriptions:

Message Code	Message	Cause(s)	Possible Corrective Action
ex.cannotEditTimesheet	User cannot edit Timesheet	User may not have the required access grant or Time Management License.	Check the user licenses and privileges (access grants).
ex.timesheetCannotbeSubmitted	Timesheet is not in unsubmitted or rework state, it cannot be submitted	Time sheets can be submitted only if they are unsubmitted or in rework.	Check the state of the time sheet that you are trying to submit.
ex.timesheetNullCannotbeSubmitted	Timesheet is null, cannot be submitted	Time sheet ID is invalid, database might be corrupt.	Check the validity of the time sheet ID.
ex.tsIncorrectSubmitStatus	This timesheet can not be submitted as timesheet status is submitted or there is no unsubmitted timesheet line	Time sheet status is incorrect to submit.	Perform the getTimeSheet operation to check the time sheet status, and then perform other appropriate operations.
ex.timesheetApproverNotSet	Billing or Time Approver for this resource is not set	Billing or time approver is needed for submit operation.	Set the billing or time approver in PPM Center for the resource of the timesheet.

approveTimeSheet

Purpose

Approve an existing Time Sheet object in PPM Center.

Note: This operation supports the Day in Days option and Period in Days option, which were introduced since PPM Center version 8.00 Service Pack 1. For more information, see the Time Sheet Update Service section in *Project and Portfolio Management Center Version 8.00.01 (Service Pack 1) Release Notes*.

Function

This operation updates the status of all the lines of an existing time sheet, and the status of the time sheet to "approved."

Before this operation, the status of the time sheet must be "pending-approval" or "in-rework" and the statuses of each Line must be "submitted" or "rejected."

The user performing this operation must have the permission to approve this particular time sheet, meaning that the user must meet the following conditions:

- Have the Time Mgmt: Edit Time Sheets access grant.
- Be one of the following:
 - The time sheet line approver.
 - The delegate of the time sheet line approver.

An additional field updated in this operation is the Actual Approver, which is set to be the approver's user ID.

Input

The ID of the time sheet that you want to approve.

Return

An object of TimeSheetBean, complete with all the fields.

Java Interface

```
ApproveTimeSheetResponseDocument approveTimeSheet(ApproveTimeSheetDocument in)
```

Parameters	Description
ApproveTimeSheetDocument	Wrapper for TimeSheetBean. See the following example for the construction.
ApproveTimeSheetResponseDocument	Wrapper for TimeSheetBean. See the following example on how to retrieve the bean.

Java Examples

Example: approve an existing time sheet.

```
TimeSheetBean createdTimeSheetBean = null;
    public void testApproveTimeSheetAllSubmitted() {
        System.out.println("testApproveTimeSheetAllSubmitted
started ...");
        try {
            if(createdTimeSheetBean == null)
                testSubmitTimeSheet();

            TimeServiceStub stub = new TimeServiceStub(ctx,
WSURL);
            ApproveTimeSheetDocument approveTimeSheetDocument =
ApproveTimeSheetDocument.Factory.newInstance();
            TimeSheetBean approveTimeSheetBean =
approveTimeSheetDocument.addNewApproveTimeSheet().addNewTimeShe
etBean();

approveTimeSheetBean.setTimeSheetId(createdTimeSheetBean.getTim
eSheetId());

approveTimeSheetBean.setResourceId(createdTimeSheetBean.getReso
urceId());

approveTimeSheetBean.setPeriodId(createdTimeSheetBean.getPeriod
Id());
            TimeSheetStatus timeSheetStatus =
approveTimeSheetBean.addNewState();
            timeSheetStatus.setCode(new BigInteger("2"));
            timeSheetStatus.setMeaning("pending-approval");
            TimeSheetLineBean timeSheetLineBean =
approveTimeSheetBean.addNewTimeSheetLines();

timeSheetLineBean.setTimeSheetLineId(createdTimeSheetBean.getTi
meSheetLinesArray(0).getTimeSheetLineId());

timeSheetLineBean.setWorkItemId(createdTimeSheetBean.getTimeShe
etLinesArray(0).getWorkItemId());
```

```
timeSheetLineBean.setWorkItemSetId(createdTimeSheetBean.getTimeSheetLinesArray(0).getWorkItemSetId());

timeSheetLineBean.setWorkItemType(createdTimeSheetBean.getTimeSheetLinesArray(0).getWorkItemType());
    TimeSheetLineStatus timeSheetLineStatus =
timeSheetLineBean.addNewState();
    timeSheetLineStatus.setCode(new BigInteger("2"));
    timeSheetLineStatus.setMeaning("submitted");
    TimeActualsBean timeActualsBean1 =
timeSheetLineBean.addNewTimeActualsList();
    timeActualsBean1.setTotalsFlag(true);
    timeActualsBean1.setEffortsListArray(new double[]
{1,2,3,4,5,0,0,1,2,3,4,5,0,0,10});

    TimeActualsBean timeActualsBean2 =
timeSheetLineBean.addNewTimeActualsList();
    timeActualsBean2.setTotalsFlag(false);
    timeActualsBean2.setEffortsListArray(new double[]
{1,2,3,4,5,0,0,1,2,3,4,5,0,0,10});

    ApproveTimeSheetResponseDocument response =
stub.approveTimeSheet(approveTimeSheetDocument);
    TimeSheetBean returnTimeSheet =
response.getApproveTimeSheetResponse().getReturn();

    System.out.println("Time Sheet was approved " +
returnTimeSheet.getTimeSheetId());
    }
    catch(Exception e) {
        e.printStackTrace();
    }
    System.out.println("testApproveTimeSheetAllSubmitted
completed");
}
```

Errors and Exceptions

When an error occurs on this operation, you will receive a description of the root cause in the log or in the response message.

The server log file content is similar to the following:

```
Exception in thread "main" org.apache.axis2.AxisFault: [root cause description]
```

Response message:

```
<exception:exceptionDetails xmlns:exception="http://
www.mercury.com/ppm/ws/exception">
```

```
<exception:detail>[root cause description] </exception:detail>
</exception:exceptionDetails>
```

Possible root cause descriptions:

Message Code	Message	Cause(s)	Possible Corrective Action
ex.cannotApproveTimesheet	User cannot approve Timesheet	You may not have the required access grant or Time Management License.	Check the user licenses and privileges (access grants).
ex.timesheetCannotbeApproved	Timesheet is not in pending approval state, it cannot be approved	The time sheet that you try to approve is not pending approval.	Check the state of the time sheet that you are trying to approve.
ex.timesheetNullCannotbeApproved	Timesheet is null, cannot be approved	Time sheet ID is invalid, or database might be corrupt.	Check the validity of the time sheet ID.
ex.timeApproverNotSet	Time Approver for this resource is not set	Time approver for this resource is not set in PPM Center	Set time approver for this resource in PPM Center.
ex.tsIncorrectApprovalStatus	This timesheet can not be approved as timesheet status is unsubmitted	Time Sheet status is incorrect to approve	Perform the getTimeSheet operation to check the time sheet status, and then perform other appropriate operations.

approveTimeSheetLine

Purpose

Approve an existing time sheet line object in PPM Center.

Note: This operation supports the Day in Days option and Period in Days option, which were introduced since PPM Center version 8.00 Service Pack 1. For more information, see the Time Sheet Update Service section in *Project and Portfolio Management Center Version 8.00.01 (Service Pack 1) Release Notes*.

Function

This operation updates the status of a time sheet line to be "approved."

Before this operation, the status of the Line must be "submitted" or "rejected."

The user performing this operation must have the permission to approve this particular time sheet line, meaning that the user must meet the following conditions:

- Have the Time Mgmt: Edit Time Sheets access grant.
- Be one of the following:
 - The time sheet line approver.
 - The delegate of the time sheet line approver.

An additional field this operation updates is the Actual Approver, which is set to the approver's user ID.

Input

The ID of the time sheet and the ID of the time sheet line that you want to approve.

Return

An object of TimeSheetBean with all the fields completed.

Java Interface

```
ApproveTimeSheetLineResponseDocument approveTimeSheetLine  
(ApproveTimeSheetLineDocument in)
```


Parameters	Description
TimeSheetLineID	Long value of the time sheet line ID.
ApproveTimeSheetLineDocument	Wrapper for TimeSheetBean and a long value for a time sheet line ID. See the following example for the construction.
ApproveTimeSheetLineResponseDocument	Wrapper for TimeSheetBean. See the following example on how to retrieve the bean.

Java Examples

Example: approve an existing time sheet line.

```
TimeSheetBean createdTimeSheetBean = null;
public void testApproveTimeSheetNotAllApproved() {
    System.out.println("testApproveTimeSheetNotAllApproved
started ...");
    try {
        if(createdTimeSheetBean == null)
            testSubmitTimeSheet();

        TimeServiceStub stub = new TimeServiceStub(ctx,
WSURL);
        ApproveTimeSheetDocument approveTimeSheetDocument =
ApproveTimeSheetDocument.Factory.newInstance();
        TimeSheetBean approveTimeSheetBean =
approveTimeSheetDocument.addNewApproveTimeSheet().addNewTimeShe
etBean();

approveTimeSheetBean.setTimeSheetId(createdTimeSheetBean.getTim
eSheetId());

approveTimeSheetBean.setResourceId(createdTimeSheetBean.getReso
urceId());

approveTimeSheetBean.setPeriodId(createdTimeSheetBean.getPeriod
Id());
        TimeSheetStatus timeSheetStatus =
approveTimeSheetBean.addNewState();
        timeSheetStatus.setCode(new BigInteger("2"));
        timeSheetStatus.setMeaning("pending-approval");

        TimeSheetLineBean timeSheetLineBean1 =
approveTimeSheetBean.addNewTimeSheetLines();

timeSheetLineBean1.setTimeSheetLineId(createdTimeSheetBean.getT
imeSheetLinesArray(0).getTimeSheetLineId());
```

```
timeSheetLineBean1.setWorkItemId(createdTimeSheetBean.getTimeSheetLinesArray(0).getWorkItemId());

timeSheetLineBean1.setWorkItemSetId(createdTimeSheetBean.getTimeSheetLinesArray(0).getWorkItemSetId());

timeSheetLineBean1.setWorkItemType(createdTimeSheetBean.getTimeSheetLinesArray(0).getWorkItemType());
    TimeSheetLineStatus timeSheetLineStatus1 =
timeSheetLineBean1.addNewState();
    timeSheetLineStatus1.setCode(new BigInteger("4"));
    timeSheetLineStatus1.setMeaning("rejected");
    TimeActualsBean timeActualsBean11 =
timeSheetLineBean1.addNewTimeActualsList();
    timeActualsBean11.setTotalsFlag(true);
    timeActualsBean11.setEffortsListArray(new double[]
{1,2,3,4,5,0,0,1,2,3,4,5,0,0,10});

    TimeActualsBean timeActualsBean21 =
timeSheetLineBean1.addNewTimeActualsList();
    timeActualsBean21.setTotalsFlag(false);
    timeActualsBean21.setEffortsListArray(new double[]
{1,2,3,4,5,0,0,1,2,3,4,5,0,0,10});

    TimeSheetLineBean timeSheetLineBean2 =
approveTimeSheetBean.addNewTimeSheetLines();

timeSheetLineBean2.setTimeSheetLineId(createdTimeSheetBean.getTimeSheetLinesArray(0).getTimeSheetLineId());

timeSheetLineBean2.setWorkItemId(createdTimeSheetBean.getTimeSheetLinesArray(0).getWorkItemId());

timeSheetLineBean2.setWorkItemSetId(createdTimeSheetBean.getTimeSheetLinesArray(0).getWorkItemSetId());

timeSheetLineBean2.setWorkItemType(createdTimeSheetBean.getTimeSheetLinesArray(0).getWorkItemType());
    TimeSheetLineStatus timeSheetLineStatus2 =
timeSheetLineBean2.addNewState();
    timeSheetLineStatus2.setCode(new BigInteger("5"));
    timeSheetLineStatus2.setMeaning("cancelled");
    TimeActualsBean timeActualsBean12 =
timeSheetLineBean2.addNewTimeActualsList();
    timeActualsBean12.setTotalsFlag(true);
    timeActualsBean12.setEffortsListArray(new double[]
{1,2,3,4,5,0,0,1,2,3,4,5,0,0,10});

    TimeActualsBean timeActualsBean22 =
timeSheetLineBean2.addNewTimeActualsList();
    timeActualsBean22.setTotalsFlag(false);
```

```

        timeActualsBean22.setEffortsListArray(new double[]
{1,2,3,4,5,0,0,1,2,3,4,5,0,0,10});

        ApproveTimeSheetResponseDocument response =
stub.approveTimeSheet(approveTimeSheetDocument);
        TimeSheetBean returnTimeSheet =
response.getApproveTimeSheetResponse().getReturn();
        System.out.println("Time Sheet lines were approved "
+ returnTimeSheet.getTimeSheetId());
    }
    catch(Exception e) {
        e.printStackTrace();
    }
    System.out.println("testApproveTimeSheetNotAllApproved
completed");
}
    
```

Errors and Exceptions

When an error occurs on this operation, you will receive a description of the root cause in the log or in the response message.

The server log file content is similar to the following:

Exception in thread "main" org.apache.axis2.AxisFault: [root cause description]

Response message:

```

<exception:exceptionDetails xmlns:exception="http://
www.mercury.com/ppm/ws/exception">
<exception:detail>[root cause description] </exception:detail>
</exception:exceptionDetails>
    
```

Possible root cause descriptions:

Message Code	Message	Cause(s)	Possible Corrective Action
ex.cannotApproveTimesheetLine	User cannot approve Timesheet Line	You may not have the required access grant or Time Management License.	Check the user licenses and privileges (access grants).
ex.timesheetLineCannotbeApproved	Timesheet	You try to	Check the state of the

Message Code	Message	Cause(s)	Possible Corrective Action
	line is not in submitted or rejected state, it cannot be approved	approve a time sheet line that is neither submitted nor rejected.	time sheet line that you are trying to approve.
ex.timesheetLineNotExist	Timesheet line does not exist in this timesheet	The time sheet ID or line ID is invalid, or database might be corrupt.	Check the validity of the time sheet ID and line ID.
ex.timesheetLineNullCannotbeApproved	Timesheet line or timesheet is null, cannot be approved	Time sheet ID or line ID is invalid, or database might be corrupt.	Check the validity of the time sheet ID and line ID.
ex.tsIncorrectApprovalStatus	This timesheet can not be approved as timesheet status is unsubmitted	Time sheet status is incorrect to approve.	Perform the getTimeSheet operation to check the time sheet status, and then perform other appropriate operations.
ex.timeApproverNotSet	Time Approver for this resource is not set	Time approver for this resource is not set in PPM Center.	Set time approver for this resource in PPM Center.

rejectTimeSheet

Purpose

Reject an existing time sheet object in PPM Center.

Note: This operation supports the Day in Days option and Period in Days option, which were

introduced since PPM Center version 8.00 Service Pack 1. For more information, see the Time Sheet Update Service section in *Project and Portfolio Management Center Version 8.00.01 (Service Pack 1) Release Notes*.

Function

This operation updates the status of all the lines of an existing time sheet to "rejected," and the status of the time sheet to be "in-rework."

Before this operation, the status of the time sheet must be "pending-approval."

The user performing this operation must have the permission to reject this particular time sheet, meaning that the user must meet the following conditions:

- Have the Time Mgmt: Edit Time Sheets access grant.
- Be one of the following:
 - The time sheet line approver.
 - The delegate of the time sheet line approver.

An additional field this operation updates is the Actual Approver, which is set to null.

Input

The ID of the time sheet that you want to reject.

Return

An object of TimeSheetBean, with all the fields completed.

Java Interface

```
RejectTimeSheetResponseDocument rejectTimeSheet(RejectTimeSheetDocument in)
```

Parameters	Description
RejectTimeSheetDocument	Wrapper for TimeSheetBean. See the following example for the construction.
RejectTimeSheetResponseDocument	Wrapper for TimeSheetBean. See the following example on how to retrieve the bean.

Java Examples

Example: reject an existing time sheet.

```
TimeSheetBean createdTimeSheetBean = null;
public void testRejectTimeSheet() {
    System.out.println("testRejectTimeSheet started ...");
    try {
        if(createdTimeSheetBean == null)
            testSubmitTimeSheet();

        TimeServiceStub stub = new TimeServiceStub(ctx,
WSURL);
        RejectTimeSheetDocument rejectTimeSheetDocument =
RejectTimeSheetDocument.Factory.newInstance();
        TimeSheetBean rejectTimeSheetBean =
rejectTimeSheetDocument.addNewRejectTimeSheet().addNewTimeSheet
Bean();

rejectTimeSheetBean.setTimeSheetId(createdTimeSheetBean.getTime
SheetId());

rejectTimeSheetBean.setResourceId(createdTimeSheetBean.getResou
rceId());

rejectTimeSheetBean.setPeriodId(createdTimeSheetBean.getPeriodI
d());
        TimeSheetStatus timeSheetStatus =
rejectTimeSheetBean.addNewState();
        timeSheetStatus.setCode(new BigInteger("2"));
        timeSheetStatus.setMeaning("pending-approval");
        TimeSheetLineBean timeSheetLineBean =
rejectTimeSheetBean.addNewTimeSheetLines();

timeSheetLineBean.setTimeSheetLineId(createdTimeSheetBean.getTi
meSheetLinesArray(0).getTimeSheetLineId());

timeSheetLineBean.setWorkItemId(createdTimeSheetBean.getTimeShe
etLinesArray(0).getWorkItemId());

timeSheetLineBean.setWorkItemSetId(createdTimeSheetBean.getTime
SheetLinesArray(0).getWorkItemSetId());
```

```
timeSheetLineBean.setWorkItemType(createdTimeSheetBean.getTimeS  
heetLinesArray(0).getWorkItemType());  
    TimeActualsBean timeActualsBean1 =  
timeSheetLineBean.addNewTimeActualsList();  
    timeActualsBean1.setTotalsFlag(true);  
    timeActualsBean1.setEffortsListArray(new double[]  
{1,2,3,4,5,0,0,1,2,3,4,5,0,0,10});  
  
    TimeActualsBean timeActualsBean2 =  
timeSheetLineBean.addNewTimeActualsList();  
    timeActualsBean2.setTotalsFlag(false);  
    timeActualsBean2.setEffortsListArray(new double[]  
{1,2,3,4,5,0,0,1,2,3,4,5,0,0,10});  
  
    RejectTimeSheetResponseDocument response =  
stub.rejectTimeSheet(rejectTimeSheetDocument);  
    TimeSheetBean returnTimeSheet =  
response.getRejectTimeSheetResponse().getReturn();  
    System.out.println("Time Sheet was rejected " +  
returnTimeSheet.getTimeSheetId());  
    }  
    catch(Exception e) {  
        e.printStackTrace();  
    }  
    System.out.println("testRejectTimeSheet completed");  
}
```

Errors and Exceptions

When an error occurs on this operation, you will receive a description of the root cause in the log or in the response message.

The server log file content is similar to the following:

```
Exception in thread "main" org.apache.axis2.AxisFault: [root cause description]
```

Response message:

```
<exception:exceptionDetails xmlns:exception="http://  
www.mercury.com/ppm/ws/exception">  
<exception:detail>[root cause description] </exception:detail>  
</exception:exceptionDetails>
```

Possible root cause descriptions:

Message Code	Message	Cause(s)	Possible Corrective Action
ex.cannotRejectTimesheet	User cannot reject Timesheet	You may not have the required access grant or Time Management License.	Check the user licenses and privileges (access grants).
ex.timesheetCannotbeRejected	Timesheet is not in pending approval state, it cannot be rejected	You try to reject a time sheet that is not pending approval.	Check the state of the time sheet that you are trying to reject.
ex.timesheetNullCannotbeRejected	Timesheet is null, cannot be rejected	Time sheet ID is invalid, or database might be corrupt.	Check the validity of the time sheet ID.
ex.tsIncorrectRejectStatus	This timesheet can not be rejected as timesheet status is approved	Time sheet status is incorrect to reject	Perform the getTimeSheet operation to check the time sheet status, and then perform other appropriate operations.

rejectTimeSheetLine

Purpose

Reject an existing time sheet line object in PPM Center.

Note: This operation supports the Day in Days option and Period in Days option, which were introduced since PPM Center version 8.00 Service Pack 1. For more information, see the Time Sheet Update Service section in *Project and Portfolio Management Center Version 8.00.01 (Service Pack 1) Release Notes*.

Function

This operation updates the status of a time sheet line to be "rejected."

Before this operation, the status of the line must be "submitted" or "approved."

The user performing this operation must have the permission to reject this particular time sheet line, meaning that the user must meet the following conditions:

- Have the Time Mgmt: Edit Time Sheets access grant.
- Be one of the following:
 - The time sheet line approver.
 - The delegate of the time sheet line approver.

An additional field this operation updates is the Actual Approver, which is set to null.

Input

A long value of the time sheet line ID.

An object of whole TimeSheetBean where this line exists.

Return

An object of TimeSheetBean, with all the fields completed.

Java Interface

```
RejectTimeSheetLineResponseDocument rejectTimeSheetLine(RejectTimeSheetLineDocument in)
```

Parameters	Description
RejectTimeSheetLineDocument	Wrapper for TimeSheetBean and a long value for a time sheet line ID. See the following example for the

Parameters	Description
	construction.
RejectTimeSheetLineResponseDocument	Wrapper for TimeSheetBean. See the following example on how to retrieve the bean.

Java Examples

Example: reject an existing time sheet line

```
TimeSheetBean createdTimeSheetBean = null;
    public void testRejectTimeSheetLine() {
        System.out.println("testRejectTimeSheetLine started
...");
        try {
            if(createdTimeSheetBean == null)
                testSubmitTimeSheet();

            TimeServiceStub stub = new TimeServiceStub(ctx,
WSURL);
            RejectTimeSheetDocument rejectTimeSheetDocument =
RejectTimeSheetDocument.Factory.newInstance();
            TimeSheetBean rejectTimeSheetBean =
rejectTimeSheetDocument.addNewRejectTimeSheet().addNewTimeSheet
Bean();

rejectTimeSheetBean.setTimeSheetId(createdTimeSheetBean.getTime
SheetId());

rejectTimeSheetBean.setResourceId(createdTimeSheetBean.getResou
rceId());

rejectTimeSheetBean.setPeriodId(createdTimeSheetBean.getPeriodI
d());
            TimeSheetStatus timeSheetStatus =
rejectTimeSheetBean.addNewState();
            timeSheetStatus.setCode(new BigInteger("2"));
            timeSheetStatus.setMeaning("pending-approval");
            TimeSheetLineBean timeSheetLineBean =
rejectTimeSheetBean.addNewTimeSheetLines();

timeSheetLineBean.setTimeSheetLineId(createdTimeSheetBean.getTi
meSheetLinesArray(0).getTimeSheetLineId());

timeSheetLineBean.setWorkItemId(createdTimeSheetBean.getTimeShe
etLinesArray(0).getWorkItemId());

timeSheetLineBean.setWorkItemSetId(createdTimeSheetBean.getTime
SheetLinesArray(0).getWorkItemSetId());
```

```
timeSheetLineBean.setWorkItemType(createdTimeSheetBean.getTimeS  
heetLinesArray(0).getWorkItemType());  
    TimeActualsBean timeActualsBean1 =  
timeSheetLineBean.addNewTimeActualsList();  
    timeActualsBean1.setTotalsFlag(true);  
    timeActualsBean1.setEffortsListArray(new double[]  
{1,2,3,4,5,0,0,1,2,3,4,5,0,0,10});  
  
    TimeActualsBean timeActualsBean2 =  
timeSheetLineBean.addNewTimeActualsList();  
    timeActualsBean2.setTotalsFlag(false);  
    timeActualsBean2.setEffortsListArray(new double[]  
{1,2,3,4,5,0,0,1,2,3,4,5,0,0,10});  
  
    RejectTimeSheetResponseDocument response =  
stub.rejectTimeSheet(rejectTimeSheetDocument);  
    TimeSheetBean returnTimeSheet =  
response.getRejectTimeSheetResponse().getReturn();  
  
    System.out.println("Time Sheet was rejected " +  
returnTimeSheet.getTimeSheetId());  
    }  
    catch(Exception e) {  
        e.printStackTrace();  
    }  
    System.out.println("testRejectTimeSheetLine  
completed");  
}
```

Errors and Exceptions

When an error occurs on this operation, you will receive a description of the root cause in the log or in the response message.

The server log file content is similar to the following:

```
Exception in thread "main" org.apache.axis2.AxisFault: [root cause description]
```

Response message:

```
<exception:exceptionDetails xmlns:exception="http://  
www.mercury.com/ppm/ws/exception">  
<exception:detail>[root cause description] </exception:detail>  
</exception:exceptionDetails>
```

Possible root cause descriptions:

Message Code	Message	Cause(s)	Possible Corrective Action
ex.cannotRejectTimesheetLine	User cannot reject Timesheet Line	You may not have the required access grant or Time Management License.	Check the user licenses and privileges (access grants).
ex.cannotRejectTimesheetLineMore	User may not have access to reject the line or timesheet line does exist.	You may not have the required access grant or Time Management License.Or, the time sheet line does not exist.	Check the user licenses and privileges (access grants). Or, Check the validity of the time sheet ID and line ID.
ex.timesheetLineCannotbeRejected	Timesheet line is not in submitted or approved state, it cannot be rejected	You try to reject a time sheet line that is neither submitted nor approved.	Check the state of the time sheet line that you are trying to reject.
ex.timesheetLineNotExist	Timesheet line does not exist in this timesheet	Time sheet ID or line ID is invalid, or database might be corrupt.	Check the validity of the time sheet ID and line ID.
ex.timesheetLineNullCannotbeRejected	Timesheet line or timesheet is null, cannot be rejected	Time sheet ID or line ID is invalid, or database might be corrupt.	Check the validity of the time sheet ID and line ID.
ex.tsIncorrectRejectStatus	This timesheet can not be rejected as timesheet status is {0}	Time sheet status is incorrect to reject.	Perform the getTimeSheet operation to check the time sheet status, and then perform other appropriate operations.
ex.tsIncorrectRejectByApprover	There is no line item available to	All line items are approved or rejected.	Perform the getTimeSheet operation to check

Message Code	Message	Cause(s)	Possible Corrective Action
	be rejected by this approver		the time sheet status, and then perform other appropriate operations.

reworkTimeSheetLine

Purpose

Rework an existing time sheet line object in PPM Center.

Note: This operation supports the Day in Days option and Period in Days option, which were introduced since PPM Center version 8.00 Service Pack 1. For more information, see the Time Sheet Update Service section in *Project and Portfolio Management Center Version 8.00.01 (Service Pack 1) Release Notes*.

Function

This operation updates the status of a time sheet line to be "unsubmitted" and updates the time sheet to "in-rework."

Before this operation, the status of the line must be "submitted," "approved," or "rejected."

The user performing this operation must have the permission to rework this particular time sheet line, meaning that the user must meet the following conditions:

- Have the Time Mgmt: Edit Time Sheets access grant.
- Be one of the following:
 - The resource of this time sheet.
 - The delegate of the resource.
 - The manager of the resource.

An additional field this operation updates is the Actual Approver, which is set to null.

Input

A long value of the time sheet line ID.

An object of whole TimeSheetBean where this line exists.

Return

An object of TimeSheetBean, complete with all the fields.

Java Interface

```
ReworkTimeSheetLineResponseDocument reworkTimeSheetLine(ReworkTimeSheetLineDocument in)
```

Parameters	Description
ReworkTimeSheetLineDocument	Wrapper for TimeSheetBean and a long value for a time sheet line ID. See the following example for the construction.
ReworkTimeSheetLineResponseDocument	Wrapper for TimeSheetBean. See the following example on how to retrieve the bean.

Java Examples

Example: rework an existing time sheet line.

```
public void testReworkTimeSheetLine() {
    System.out.println("testReworkTimeSheetLine started ...");
    try {
        if(createdTimeSheetBean == null)
            testSubmitTimeSheet();
        TimeServiceStub stub = new TimeServiceStub(ctx, WSURL);
        ReworkTimeSheetLineDocument reworkTimeSheetLineDocument =
        ReworkTimeSheetLineDocument.Factory.newInstance();
        TimeSheetBean reworkTimeSheetBean =
        reworkTimeSheetLineDocument.addNewReworkTimeSheetLine().addNewTimeSheetBean();
```

```
reworkTimeSheetBean.setTimeSheetId(createdTimeSheetBean.getTimeSheetId());
reworkTimeSheetBean.setResourceId(createdTimeSheetBean.getResourceId());

reworkTimeSheetBean.setPeriodId(createdTimeSheetBean.getPeriodId());
    TimeSheetStatus timeSheetStatus = reworkTimeSheetBean.addNewState();
    timeSheetStatus.setCode(new BigInteger("2"));
    timeSheetStatus.setMeaning("pending-approval");
    TimeSheetLineBean timeSheetLineBean =
reworkTimeSheetBean.addNewTimeSheetLines();

timeSheetLineBean.setTimeSheetLineId(createdTimeSheetBean.getTimeSheetLinesArr
ay(0).getTimeSheetLineId());

timeSheetLineBean.setWorkItemId(createdTimeSheetBean.getTimeSheetLinesArray(0)
.getWorkItemId());

timeSheetLineBean.setWorkItemSetId(createdTimeSheetBean.getTimeSheetLinesArray
(0).getWorkItemSetId());

timeSheetLineBean.setWorkItemType(createdTimeSheetBean.getTimeSheetLinesArray(
0).getWorkItemType());
    TimeActualsBean timeActualsBean1 =
timeSheetLineBean.addNewTimeActualsList();
    timeActualsBean1.setTotalsFlag(true);
    timeActualsBean1.setEffortsListArray(new double[]
{1,2,3,4,5,0,0,1,2,3,4,5,0,0,10});

    TimeActualsBean timeActualsBean2 =
timeSheetLineBean.addNewTimeActualsList();
    timeActualsBean2.setTotalsFlag(false);
    timeActualsBean2.setEffortsListArray(new double[]
{1,2,3,4,5,0,0,1,2,3,4,5,0,0,10});

reworkTimeSheetLineDocument.getReworkTimeSheetLine().setTimeSheetLineId(create
dTimeSheetBean.getTimeSheetLinesArray(0).getTimeSheetLineId());
    ReworkTimeSheetLineResponseDocument response =
stub.reworkTimeSheetLine(reworkTimeSheetLineDocument);
    TimeSheetBean returnTimeSheet =
response.getReworkTimeSheetLineResponse().getReturn();

    System.out.println("Time Sheet was reworked " +
returnTimeSheet.getTimeSheetId());
    }
    catch(Exception e) {
        e.printStackTrace();
    }
    System.out.println("testReworkTimeSheetLine completed");
}
```

Errors and Exceptions

When an error occurs on this operation, you will receive a description of the root cause in the log or in the response message.

The server log file content is similar to the following:

```
Exception in thread "main" org.apache.axis2.AxisFault: [root cause description]
```

Response message:

```
<exception:exceptionDetails xmlns:exception="http://www.mercury.com/ppm/ws/exception">
<exception:detail>[root cause description] </exception:detail>
</exception:exceptionDetails>
```

Possible root cause descriptions:

Message Code	Message	Cause(s)	Possible Corrective Action
ex.cannotReworkTimesheetLine	User cannot rework Timesheet Line	You may not have the required access grant or Time Management License.	Check the user licenses and privileges (access grants).
ex.timesheetLineCannotbeReworked	Timesheet line is not in submitted, rejected or approved state, it cannot be reworked	You try to rework a time sheet line that is not submitted, rejected, or approved.	Check the state of the time sheet line that you are trying to rework.
ex.timesheetLineNotExist	Timesheet line does not exist in this timesheet	Time sheet ID or line ID is invalid, database might be corrupt.	Check the validity of the time sheet ID and line ID.
ex.timesheetLineNullCannotbeReworked	Timesheet line or timesheet is null, cannot be	Time sheet ID or line ID is invalid,	Check the validity of the time sheet ID and line ID.

Message Code	Message	Cause(s)	Possible Corrective Action
	reworked	database might be corrupt	
ex.tsIncorrectReworkStatus	TimeSheet line can not be reworked as timesheet status is unsubmitted	Time sheet line status is incorrect to rework.	Perform the getTimeSheet operation to check the time sheet line status, and then perform other appropriate operations
ex.tsIncorrectReworkStatus	Rework can not be done on unsubmitted timesheet line	Time sheet line status is incorrect to rework.	Perform the getTimeSheet operation to check the time sheet line status, and then perform other appropriate operations.
ex.tsIncorrectReworkLineStatus	TimeSheet line can not be reworked as timesheet line status is {0}	Time sheet line status is incorrect to rework.	Perform the getTimeSheet operation to check the time sheet line status, and then perform other appropriate operations.
ex.tsReworkLineBeanIdNotMatch	There is no matching timesheet line ({0}) in timesheet bean to rework	Time sheet line ID is invalid.	Check the validity of the time sheet line ID.
ex.tsReworkLineIdNotMatch	There is no matching timesheet line ({0}) in timesheet to rework	Time sheet line ID is invalid.	Check the validity of the time sheet line ID.

freezeTimeSheet

Purpose

Freeze an existing time sheet object in PPM Center.

Note: This operation supports the Day in Days option and Period in Days option, which were introduced since PPM Center version 8.00 Service Pack 1. For more information, see the Time Sheet Update Service section in *Project and Portfolio Management Center Version 8.00.01 (Service Pack 1) Release Notes*.

Function

This operation updates the status of all the lines of an existing time sheet, and the status of the time sheet to "frozen."

Before this operation, the status of the time sheet must be "approved." After this operation, only the closeTimeSheet operation is permitted.

The user performing this operation must have the permission to freeze this particular time sheet, meaning that the user must meet the following conditions:

- Have the Time Mgmt: Edit Time Sheets access grant.
- Be one of the following:
 - The billing approver of the time sheet.
 - The delegate of the billing approver.

Input

An object of whole TimeSheetBean to freeze.

Return

An object of `TimeSheetBean`, with all the fields completed.

Java Interface

```
FreezeTimeSheetResponseDocument freezeTimeSheet(FreezeTimeSheetDocument in)
```

Parameters	Description
<code>FreezeTimeSheetDocument</code>	Wrapper for <code>TimeSheetBean</code> . See the following example for the construction.
<code>FreezeTimeSheetResponseDocument</code>	Wrapper for <code>TimeSheetBean</code> . See the following example on how to retrieve the bean.

Java Examples

Example: freeze an existing time sheet

```
TimeSheetBean createdTimeSheetBean = null;
public void testFreezeTimeSheet() {
    System.out.println("testFreezeTimeSheet started ...");
    try {
        if(createdTimeSheetBean == null)
            testApproveTimeSheetAllApproved();

        TimeServiceStub stub = new TimeServiceStub(ctx,
WSURL);
        FreezeTimeSheetDocument freezeTimeSheetDocument =
FreezeTimeSheetDocument.Factory.newInstance();
        TimeSheetBean freezeTimeSheetBean =
freezeTimeSheetDocument.addNewFreezeTimeSheet().addNewTimeSheet
Bean();

freezeTimeSheetBean.setTimeSheetId(createdTimeSheetBean.getTime
SheetId());

freezeTimeSheetBean.setResourceId(createdTimeSheetBean.getResou
rceId());

freezeTimeSheetBean.setPeriodId(createdTimeSheetBean.getPeriodI
d());
```

```
        TimeSheetStatus timeSheetStatus =
freezeTimeSheetBean.addNewState();
        timeSheetStatus.setCode(new BigInteger("4"));
        timeSheetStatus.setMeaning("approved");
        TimeSheetLineBean timeSheetLineBean =
freezeTimeSheetBean.addNewTimeSheetLines();

timeSheetLineBean.setTimeSheetLineId(createdTimeSheetBean.getTi
meSheetLinesArray(0).getTimeSheetLineId());

timeSheetLineBean.setWorkItemId(createdTimeSheetBean.getTimeShe
etLinesArray(0).getWorkItemId());

timeSheetLineBean.setWorkItemSetId(createdTimeSheetBean.getTime
SheetLinesArray(0).getWorkItemSetId());

timeSheetLineBean.setWorkItemType(createdTimeSheetBean.getTimeS
heetLinesArray(0).getWorkItemType());
        TimeActualsBean timeActualsBean1 =
timeSheetLineBean.addNewTimeActualsList();
        timeActualsBean1.setTotalsFlag(true);
        timeActualsBean1.setEffortsListArray(new double[]
{1,2,3,4,5,0,0,1,2,3,4,5,0,0,10});

        TimeActualsBean timeActualsBean2 =
timeSheetLineBean.addNewTimeActualsList();
        timeActualsBean2.setTotalsFlag(false);
        timeActualsBean2.setEffortsListArray(new double[]
{1,2,3,4,5,0,0,1,2,3,4,5,0,0,10});

        FreezeTimeSheetResponseDocument response =
stub.freezeTimeSheet(freezeTimeSheetDocument);
        TimeSheetBean returnTimeSheet =
response.getFreezeTimeSheetResponse().getReturn();

        System.out.println("Time Sheet was frozen " +
returnTimeSheet.getTimeSheetId());
    }
    catch(Exception e) {
        e.printStackTrace();
    }
    System.out.println("testFreezeTimeSheet completed");
}
```

Errors and Exceptions

When an error occurs on this operation, you will receive a description of the root cause in the log or in the response message.

The server log file content is similar to the following:

Exception in thread "main" org.apache.axis2.AxisFault: [root cause description]

Response message:

```
<exception:exceptionDetails xmlns:exception="http://
www.mercury.com/ppm/ws/exception">
<exception:detail>[root cause description] </exception:detail>
</exception:exceptionDetails>
```

Possible root cause descriptions:

Message Code	Message	Cause(s)	Possible Corrective Action
ex.cannotFreezeTimesheet	User cannot freeze Timesheet	You may not have the required access grant or Time Management License.	Check the user licenses and privileges (access grants).
ex.timesheetCannotbeFrozen	Timesheet is not in approved state, it cannot be frozen	You try to freeze a time sheet that is not approved.	Check the state of the time sheet that you are trying to freeze.
ex.timesheetNullCannotbeFrozen	Timesheet is null, cannot be frozen	Time sheet ID is invalid, database might be corrupt.	Check the validity of the time sheet ID.
ex.tsIncorrectFreezeStatus	This timesheet can not be frozen as timesheet status is {0}	Time sheet line status is incorrect to freeze.	Perform the getTimeSheet operation to check the time sheet line status, and then perform other appropriate operations.

closeTimeSheet

Purpose

Close an existing time sheet object in PPM Center.

Note: This operation supports the Day in Days option and Period in Days option, which were introduced since PPM Center version 8.00 Service Pack 1. For more information, see the Time Sheet Update Service section in *Project and Portfolio Management Center Version 8.00.01 (Service Pack 1) Release Notes*.

Function

This operation updates the status of all the lines of an existing time sheet and the status of the time sheet to "closed."

Before this operation, the status of the time sheet must be "approved" or "frozen."

The user performing this operation must have the permission to close this particular time sheet, meaning that the user must meet the following conditions:

- Have the Time Mgmt: Edit Time Sheets access grant.
- Be one of the following:
 - The billing approver of the time sheet.
 - The delegate of the billing approver.

Input

The ID of the time sheet object that you want to close.

Return

An object of `TimeSheetBean`, complete with all the fields.

Java Interface

```
CloseTimeSheetResponseDocument closeTimeSheet(CloseTimeSheetDocument in)
```

Parameters	Description
<code>CloseTimeSheetDocument</code>	Wrapper for <code>TimeSheetBean</code> . See the following example for the construction.
<code>CloseTimeSheetResponseDocument</code>	Wrapper for <code>TimeSheetBean</code> . See the following example on how to retrieve the bean.

Java Examples

Example: close an existing time sheet

```
TimeSheetBean createdTimeSheetBean = null;
public void testCloseTimeSheet() {
    System.out.println("testCloseTimeSheet started ...");
    try {
        if(createdTimeSheetBean == null)
            testFreezeTimeSheet();

        TimeServiceStub stub = new TimeServiceStub(ctx,
WSURL);
        CloseTimeSheetDocument closeTimeSheetDocument =
CloseTimeSheetDocument.Factory.newInstance();
        TimeSheetBean closeTimeSheetBean =
closeTimeSheetDocument.addNewCloseTimeSheet().addNewTimeSheetBe
an();

closeTimeSheetBean.setTimeSheetId(createdTimeSheetBean.getTimeS
heetId());

closeTimeSheetBean.setResourceId(createdTimeSheetBean.getResourc
eId());

closeTimeSheetBean.setPeriodId(createdTimeSheetBean.getPeriodId
());
```

```
        TimeSheetStatus timeSheetStatus =
closeTimeSheetBean.addNewState();
        timeSheetStatus.setCode(new BigInteger("6"));
        timeSheetStatus.setMeaning("frozen");
        TimeSheetLineBean timeSheetLineBean =
closeTimeSheetBean.addNewTimeSheetLines();

timeSheetLineBean.setTimeSheetLineId(createdTimeSheetBean.getTi
meSheetLinesArray(0).getTimeSheetLineId());

timeSheetLineBean.setWorkItemId(createdTimeSheetBean.getTimeShe
etLinesArray(0).getWorkItemId());

timeSheetLineBean.setWorkItemSetId(createdTimeSheetBean.getTime
SheetLinesArray(0).getWorkItemSetId());

timeSheetLineBean.setWorkItemType(createdTimeSheetBean.getTimeS
heetLinesArray(0).getWorkItemType());
        TimeActualsBean timeActualsBean1 =
timeSheetLineBean.addNewTimeActualsList();
        timeActualsBean1.setTotalsFlag(true);
        timeActualsBean1.setEffortsListArray(new double[]
{1,2,3,4,5,0,0,1,2,3,4,5,0,0,10});

        TimeActualsBean timeActualsBean2 =
timeSheetLineBean.addNewTimeActualsList();
        timeActualsBean2.setTotalsFlag(false);
        timeActualsBean2.setEffortsListArray(new double[]
{1,2,3,4,5,0,0,1,2,3,4,5,0,0,10});

        CloseTimeSheetResponseDocument response =
stub.closeTimeSheet(closeTimeSheetDocument);
        TimeSheetBean returnTimeSheet =
response.getCloseTimeSheetResponse().getReturn();

        System.out.println("Time Sheet was closed " +
returnTimeSheet.getTimeSheetId());
    }
    catch(Exception e) {
        e.printStackTrace();
    }
    System.out.println("testCloseTimeSheet completed");
}
```

Errors and Exceptions

When an error occurs on this operation, you will receive a description of the root cause in the log or in the response message.

The server log file content is similar to the following:

Exception in thread "main" org.apache.axis2.AxisFault: [root cause description]

Response message:

```
<exception:exceptionDetails xmlns:exception="http://
www.mercury.com/ppm/ws/exception">
<exception:detail>[root cause description] </exception:detail>
</exception:exceptionDetails>
```

Possible root cause descriptions:

Message Code	Message	Cause(s)	Possible Corrective Action
ex.cannotCloseTimesheet	User cannot close Timesheet	You may not have the required access grant or Time Management License.	Check the user licenses and privileges (access grants).
ex.timesheetCannotbeClosed	Timesheet is not in approved or frozen state, it cannot be closed	You try to close a time sheet that is neither approved, nor frozen.	Check the state of the time sheet that you are trying to close.
ex.timesheetNullCannotbeClosed	Timesheet is null, cannot be closed	Time sheet ID is invalid, database might be corrupt.	Check the validity of the time sheet ID.
ex.tsIncorrectCloseStatus	This timesheet can not be closed as timesheet status is {0}	Time sheet line status is incorrect to close.	Perform the getTimeSheet operation to check the time sheet line status, and then perform other appropriate operations.

cancelTimeSheet

Purpose

Cancel an existing Time Sheet object in PPM Center.

Note: This operation supports the Day in Days option and Period in Days option, which were introduced since PPM Center version 8.00 Service Pack 1. For more information, see the Time Sheet Update Service section in *Project and Portfolio Management Center Version 8.00.01 (Service Pack 1) Release Notes*.

Function

This operation updates the status of all the lines of an existing time sheet and the status of the time sheet to "cancelled." Additionally, this operation removes the corresponding time actuals from the project.

Before this operation, the status of the time sheet must be "unsubmitted," or "in-rework."

The user performing this operation must have the permission to cancel this particular time sheet, meaning that the user must meet the following conditions:

- Have the Time Mgmt: Edit Time Sheets access grant.
- Be one of the following:
 - The resource of this time sheet.
 - The delegate of the resource.
 - The manager of the resource.

Input

The ID of the time sheet object that you want to cancel.

Return

An object of TimeSheetBean, with all the fields completed

Java Interface

```
CancelTimeSheetResponseDocument cancelTimeSheet(CancelTimeSheetDocument in)
```

Parameters	Description
CancelTimeSheetDocument	Wrapper for TimeSheetBean. See the following example for the construction.
CancelTimeSheetResponseDocument	Wrapper for TimeSheetBean. See the following example on how to retrieve the bean.

Java Examples

Example: cancel an existing time sheet.

```
TimeSheetBean createdTimeSheetBean = null;
public void testCloseTimeSheet() {
    System.out.println("testCloseTimeSheet started ...");
    try {
        if(createdTimeSheetBean == null)
            testFreezeTimeSheet();

        TimeServiceStub stub = new TimeServiceStub(ctx,
WSURL);
        CloseTimeSheetDocument closeTimeSheetDocument =
CloseTimeSheetDocument.Factory.newInstance();
        TimeSheetBean closeTimeSheetBean =
closeTimeSheetDocument.addNewCloseTimeSheet().addNewTimeSheetBe
an();

closeTimeSheetBean.setTimeSheetId(createdTimeSheetBean.getTimeS
heetId());

closeTimeSheetBean.setResourceId(createdTimeSheetBean.getResourc
eId());

closeTimeSheetBean.setPeriodId(createdTimeSheetBean.getPeriodId
());
        TimeSheetStatus timeSheetStatus =
closeTimeSheetBean.addNewState();
        timeSheetStatus.setCode(new BigInteger("6"));
        timeSheetStatus.setMeaning("frozen");
        TimeSheetLineBean timeSheetLineBean =
closeTimeSheetBean.addNewTimeSheetLines();

timeSheetLineBean.setTimeSheetLineId(createdTimeSheetBean.getTi
meSheetLinesArray(0).getTimeSheetLineId());

timeSheetLineBean.setWorkItemId(createdTimeSheetBean.getTimeShe
etLinesArray(0).getWorkItemId());

timeSheetLineBean.setWorkItemSetId(createdTimeSheetBean.getTime
SheetLinesArray(0).getWorkItemSetId());
```

```
timeSheetLineBean.setWorkItemType(createdTimeSheetBean.getTimeS  
heetLinesArray(0).getWorkItemType());  
    TimeActualsBean timeActualsBean1 =  
timeSheetLineBean.addNewTimeActualsList();  
    timeActualsBean1.setTotalsFlag(true);  
    timeActualsBean1.setEffortsListArray(new double[]  
{1,2,3,4,5,0,0,1,2,3,4,5,0,0,10});  
  
    TimeActualsBean timeActualsBean2 =  
timeSheetLineBean.addNewTimeActualsList();  
    timeActualsBean2.setTotalsFlag(false);  
    timeActualsBean2.setEffortsListArray(new double[]  
{1,2,3,4,5,0,0,1,2,3,4,5,0,0,10});  
  
    CloseTimeSheetResponseDocument response =  
stub.closeTimeSheet(closeTimeSheetDocument);  
    TimeSheetBean returnTimeSheet =  
response.getCloseTimeSheetResponse().getReturn();  
  
    System.out.println("Time Sheet was closed " +  
returnTimeSheet.getTimeSheetId());  
    }  
    catch(Exception e) {  
        e.printStackTrace();  
    }  
    System.out.println("testCloseTimeSheet completed");  
}
```

Errors and Exceptions

When an error occurs on this operation, you will receive a description of the root cause in the log or in the response message.

The server log file content is similar to the following:

```
Exception in thread "main" org.apache.axis2.AxisFault: [root cause description]
```

Response message:

```
<exception:exceptionDetails xmlns:exception="http://  
www.mercury.com/ppm/ws/exception">  
<exception:detail>[root cause description] </exception:detail>  
</exception:exceptionDetails>
```

Possible root cause descriptions:

Message Code	Message	Cause(s)	Possible Corrective Action
ex.cannotEditTimesheet	User cannot edit Timesheet	You may not have the required access grant or Time Management License.	Check the user licenses and privileges (access grants).
ex.timesheetCannotbeCancelled	Timesheet is not in unsubmitted state, it cannot be cancelled	You try to cancel a Time sheet that is not unsubmitted.	Check the state of the time sheet that you are trying to cancel.
ex.timesheetNullCannotbeCancelled	Timesheet is null, cannot be cancelled	Time sheet ID is invalid, database might be corrupt.	Check the validity of the time sheet ID.
ex.tsIncorrectCancelStatus	This timesheet can not be cancelled as timesheet status is {0}.	Time sheet line status is incorrect to cancel.	Perform the getTimeSheet operation to check the time sheet line status, and then perform other appropriate operations.

Chapter 10: Multilingual User Interface Support in Web Services

- ["Overview" below](#)
- ["Preferred Language Setting" below](#)

Overview

PPM Center supports a multilingual user interface (MLU) to facilitate the needs of global enterprises. PPM Center Web services is also enhanced to support MLU so that users can specify their preferred languages when invoking a Web service operation.

Users can use their preferred language even if that language is different from the specified system language or session language. When calling PPM Center through Web services, you can specify the preferred language in the Simple Object Access Protocol (SOAP) header.

In the PPM Center standard interface, a user can choose a session language during log-in and a Web service caller can specify a preferred language in the SOAP header that will override the session language that the user specified.

Preferred Language Setting

Specifying the Session Language

The following example specifies the session language in the SOAP header.

```
<soap:Header>
  <common:UserLocaleHeader xmlns:common="http://mercury.com/ppm/common/1.0">
    <common:LanguageLocale>de</common:LanguageLocale>
  </common:UserLocaleHeader>

  <wssecurity xmlns:wss="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
  soap:mustUnderstand="1">
```

```

    <wsu:Timestamp xmlns:wsu="http://docs.oasis-open.org/
wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
wsu:Id="Timestamp-6557466">
        <wsu:Created>2009-08-21T18:29:15.687Z</wsu:Created>
        <wsu:Expires>2009-08-21T18:34:15.687Z</wsu:Expires>
    </wsu:Timestamp>
    <wsse:UsernameToken xmlns:wsu="http://docs.oasisopen.
org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd" wsu:Id="UsernameToken-7433399">
        <wsse:Username>admin</wsse:Username>
        <wsse:password Type="http://docs.oasis-open.org/wss/
2004/01/oasis-200401-wss-username-token-profile-
1.0#PasswordText">admin</wsse:password>
    </wsse:UsernameToken>
</wsse:security>
</soap:Header>

```

When invoking a Web service operation, a caller can specify the preferred language in the UserLocaleHeader SOAP header element. To make it easier for the caller to set this SOAP header element, the Web service toolkit has enhanced the SOAPHeaderCreator class by adding the setUserLocaleHeader method.

SOAPHeaderCreator

Purpose

This class provides interfaces to set SOAP header elements for PPM Center Web services.

Function

setAuditHeader

This method is used to add some auditing information through an Audit header.

setUserLocaleHeader

This method is used to set a preferred language when a caller invokes a Web service operation. For example, if you want to create requests in a language other than the system language, you can specify the preferred locale by using this method.

Limitations

Data in Multiple Languages

In a single web service operation, there is no support for working with data in multiple languages. To work with data in multiple languages, you must perform multiple, separate Web service transactions, each specifying the desired session language.

Supported Languages

The language code you set in the method must be supported (and enabled) in PPM Center. If the translation of a certain entity does not exist in the specified language, the result of a Get operation will be returned in the Definition Language of the entity.

Java Interface

Add the Audit header for auditing information:

```
SOAPHeaderCreator.setAuditHeader(stub, "Submitted By: TEST", "PPM on " +  
InetAddress.getLocalHost().getHostAddress(), "createRequest");
```

Parameters	Description
stub	Generated by another Web service operation. For example: DemandServiceStub stub = new DemandServiceStub(ctx, serviceURL);

Add UserLocaleHeader to set the language locale to a specified locale:

```
SOAPHeaderCreator.setUserLocaleHeader(stub, locale);
```

Parameters	Description
stub	Generated by another Web service operation. For example: DemandServiceStub stub = new DemandServiceStub(ctx, serviceURL);
locale	Language code of the language you want to set the session to, for example, de (for German) or ko (for Korean).

Java Example

This is an example for DemandService that demonstrates how to get the MLU functionality from DemandService in a PPM Center instance that supports MLU.

```
import examples.dm.DemandServiceClient;
import examples.util.SOAPHeaderCreator;

public class DemandServiceMLUClient {

    protected ConfigurationContext ctx = null;

    public DemandServiceMLUClient() {
        String repositoryPath =
System.getProperty("client.repository.dir");
        String axis2 = repositoryPath + "/conf/clientaxis2.
xml";
        File file = new File(axis2);
        if (file.exists()) {
            try {
                ctx =
ConfigurationContextFactory.createConfigurationContextFromFileS
ystem(repositoryPath, axis2);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }

    /**
     * The main program
     *
     * Parameter: args[0] - service URL. e.g.
     * http://server:port/itg/ppmservices/DemandService
     * args[1] - language locale e.g. de or en, ko etc.
     *
     */
    public static void main(String[] args) throws Exception {
        // check parameter
        if (args.length < 1) {
            System.out.println("Usage: java DemandServiceClient
<service URL> <language locale>");
            System.exit(1);
        }

        System.out.println("Starting Demand Service MLU
tests...");
    }
}
```

```

    // get server URL
    String serviceURL = args[0];
    // get preferred language
    String locale = null;
    if (args.length > 1) locale = args[1];

    // Test Create Request
    DemandServiceMLUClient dm = new
DemandServiceMLUClient();
    String requestId = dm.createRequest(serviceURL);

    // Test Get Request
    dm.getRequests(serviceURL, locale, requestId);

    System.out.println("Demand Service MLU tests
complete.");
}

/**
 * This method creates a request and it shows the
instruction on how to
 * create the request in a language other than the system
language
 * @param serviceURL
 * @return
 * @throws Exception
 */
private String createRequest(String serviceURL) throws
Exception {

    // Get web service
    DemandServiceStub stub = new DemandServiceStub(ctx,
serviceURL);
    // Add the Audit header for auditing information
    SOAPHeaderCreator.setAuditHeader(stub, "Submitted By:
TEST", "PPM on " +
InetAddress.getLocalHost().getHostAddress(), "createRequest");

/
*****
*
* Note: If you want to create a request in a language other than
the System language, you must specify the preferred locale by
uncommenting the following line:
SOAPHeaderCreator.setUserLocaleHeader(stub, locale);

Then, make sure to pass the token values in the translated
language instead of the System language. This is because when
you set the LanguageLocale field in UserLocaleHeader to a
specific locale, PPM web service expects the token values in the
corresponding language if the translation for that value in the
preferred language exists in the system. If the translation

```

doesn't exist in that language, you must pass the values in the System language.

Take the following scenario as an example:

- o You set the preferred locale to "de" and you want to set the value for the REQ.DEPARTMENT_NAME token to 'Manufacturing' (English).

- o The German translation for 'Manufacturing' exists in the system.

In this scenario, you must specify it as 'Herstellung', which is the German translated value of 'Manufacturing' and so forth for other token values as well.

```

*****/

        // Add UserLocaleHeader to set the language locale to
the specified locale
        // SOAPHeaderCreator.setUserLocaleHeader(stub, locale);

        // Construct a request object
Request oRequest = Request.Factory.newInstance();
oRequest.setRequestType("Bug");
SimpleField[] fields = new SimpleField[2];

        // Set values for the fields of the request object

        // Set field 'Description'
SimpleField field_A = SimpleField.Factory.newInstance();
field_A.setToken("REQ.DESCRPTION");
field_A.setStringValue1Array(new String[] { "WebService
Test" });
        fields[0] = field_A;

        // Set field 'Department'
SimpleField field_B = SimpleField.Factory.newInstance();
field_B.setToken("REQ.DEPARTMENT_NAME");
field_B.setStringValue1Array(new String[]{"Finance"});
        fields[1] = field_B;

        // Add all the fields to request object
oRequest.setSimpleFieldsArray(fields);
    }
/**
 * This method invokes getRequest operation with language
locale set to specified locale
 * @param serviceURL
 * @param language
 * @param requestId
 * @throws Exception
 */
private void getRequests(String serviceURL, String

```

```
language, String requestId)
    throws Exception {

    // Set Identifier
    Identifier[] ids = new Identifier[1];
    Identifier reqId = Identifier.Factory.newInstance();
    reqId.setId(requestId);
    reqId.setServerURL(serviceURL);
    ids[0] = reqId;

    // Get web service
    DemandServiceStub stub = new DemandServiceStub(ctx,
serviceURL);
    // Add the UserLocaleHeader SOAP header
    SOAPHeaderCreator.setUserLocaleHeader(stub, language);

    // Construct message to send
    GetRequestsDocument inDoc =
GetRequestsDocument.Factory.newInstance();
    GetRequestsDocument.GetRequests getRequests =
inDoc.addNewGetRequests();
    getRequests.setRequestIdsArray(ids);

    // Invoke web service
    GetRequestsResponseDocument outDoc =
stub.getRequests(inDoc);

    // Process return message
    Request[] requests =
outDoc.getGetRequestsResponse().getReturnArray();
    System.out.println("getRequests Succeeded");
    System.out.println("Returned Request: " +
requests[0].getId());
    }
}
```

Errors and Exceptions

There are no special exceptions for the SOAPHeaderCreator class.

Chapter 11: Web Service Security

- "Overview" below
- "Web Service Security on PPM Server" on the next page
- "Web Service Authentication for Web Service Toolkit" on page 458

Overview

Authentication

PPM Center Web services uses the Web services Security specification (WS-Security) to secure SOAP message exchanges. PPM Center Web services relies on a Rampart module integrated with Axis2 Web service engine to provide WS-Security support.

For more information about WS-Security specification, go to the following site:

<http://www.oasis-open.org/specs/index.php#wssv1.1>

The WS-Security specification defines a set of standard SOAP headers to provide quality of protection through the following mechanisms:

- Message integrity (XML signature)
- Message confidentiality (XML encryption)
- Single message authentication (User name token authentication, Kerberos authentication, X509 certificate authentication, and so forth.)

These mechanisms can be used to accommodate a wide variety of security models. The WS-Security specification is considered a message-level authentication protocol because all security information is carried within the SOAP message.

Out of the box, PPM Center supports WS-Security user name token authentication, timestamp validation, and encryption of WS-Security headers. PPM Center also supports HTTP basic authentication (HTTP transport-level authentication protocol), as well as HTTPS (secure) authentication.

PPM Center Web services can also be integrated with third-party single sign-on software such as SiteMinder.

Authorization

PPM Center Web services follows the same authorization model as Web applications. Refer to the *Security Model Guide and Reference* for details on specific functional areas. This section focuses only on authentication.

Web Service Security on PPM Server

This section describes the "[WS-Security Authentication](#)" below and the "[HTTP Basic Authentication](#)" on page 457.

WS-Security Authentication

WS-Security authentication includes user name token authentication, timestamp validation, and encryption of WS-Security headers. This section provides you with several examples of authentication configuration.

Enable/Disable WS-Security Authentication

WS-Security user name token configuration can be found in:

```
<PPM_Home>/server/<PPM_Server_Name>/deploy/itg.war/WEB-INF/conf/axis2.xml.
```

Variable	Meaning
<PPM_Home>	Represents the path where your PPM Center instance is installed. For example: xyzserver/E/PPMServer.
<PPM_Server_Name>	Represents the name assigned to your PPM Server during installation. For example: xyzProduction. This corresponds to the KINTANA_SERVER_NAME server.conf parameter value and does not necessarily reflect the actual host name of the server.

By default, WS-Security authentication is enabled.

axis2.xml

In the axis2.xml file, the following XML configuration enables WS-Security authentication. To disable WS-Security authentication, comment out all of this XML.

```
<module ref="rampart" />
<parameter name="InflowSecurity">
  <action>
    <items>
      UsernameToken Timestamp Encrypt
    </items>
    <passwordCallbackClass>
com.mercury.itg.ws.core.handlers.security.PasswordCallbackHandl
er
    </passwordCallbackClass>
    <decryptionPropFile>
      service.properties
    </decryptionPropFile>
  </action>
</parameter>
```

In the InflowSecurity section, the following three action items are defined:

- UsernameToken: Specifies that the UsernameToken security credentials should be expected in received SOAP messages.

The UsernameToken profile defines a set of SOAP headers to carry the username/password from the client to the server.

- Timestamp: Specifies that the Timestamp element should be validated.

The default clock skew tolerance is five minutes.

- Encrypt: Specifies that encrypted messages should be decrypted.

SOAP Header Without Timestamp Data or Encryption

The following example shows a SOAP header that does not include timestamp data or encryption:

```
<soap:Header>
  <wsse:Security xmlns:wsse="http://docs.oasis-open.org/
wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
  soap:mustUnderstand="1">
    <wsse:UsernameToken
      xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-wssecurity-utility-1.0.xsd"
      wsu:Id="UsernameToken-25699763">
      <wsse:Username>admin</wsse:Username>
```

```
<wsse:Password
  Type="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-username-token-profile-
1.0#PasswordText">admin</wsse:Password>
  </wsse:UsernameToken>
</wsse:Security>
</soap:Header>
```

Refer to the Axis2 Rampart module's documentation for more information:

http://ws.apache.org/axis2/modules/rampart/1_0/security-module.html

WS-Security Timestamp

By default, PPM Center enables Timestamp validation.

To disable Timestamp validation on the server side, remove "Timestamp" from the action items list as shown in the following example:

```
<module ref="rampart" />
<parameter name="InflowSecurity">
  <action>
    <items>
      UsernameToken Encrypt
    </items>
    <passwordCallbackClass>
com.mercury.itg.ws.core.handlers.security.PasswordCallbackHandl
er
    </passwordCallbackClass>
    <decryptionPropFile>
      service.properties
    </decryptionPropFile>
  </action>
</parameter>
```

WS-Security Encryption

By default, PPM Center encrypts WS-Security user name token headers.

To disable encryption, remove "Encrypt" from the action items list as shown in the following example:

```
<module ref="rampart" />
<parameter name="InflowSecurity">
  <action>
    <items>
```



```
        UsernameToken Timestamp
    </items>
    <passwordCallbackClass>
com.mercury.itg.ws.core.handlers.security.PasswordCallbackHandl
er
    </passwordCallbackClass>
    <decryptionPropFile>
        service.properties
    </decryptionPropFile>
    </action>
</parameter>
```

The following content shows an example of the service.properties file:

```
org.apache.ws.security.crypto.provider=org.apache.ws.security.c
omponents.crypto.Merlin
org.apache.ws.security.crypto.merlin.keystore.type=jks
org.apache.ws.security.crypto.merlin.keystore.password=ppmservi
ce
org.apache.ws.security.crypto.merlin.file=service.jks
```

In this example, the java key store file is defined as the security properties file, and the password to the key store file is ppmservice.

HTTP Basic Authentication

In a scenario where it is not convenient to use WS-Security, it is possible to configure PPM Center to accept user credentials passed by using HTTP basic authentication headers.

1. Open the axis2.xml file.
2. Locate "InflowBasicAuth" section.
3. Change the value of "Enforced" to true:

```
<parameter name="InFlowBasicAuth">
    <Enforced>true</Enforced>
</parameter>
```

4. Save and close the axis2.xml file.

When HTTP basic authentication is enabled on the PPM Server, the credential carried in HTTP authentication header is authenticated against PPM Center users' credentials.

The following example shows an http header using the HTTP basic authentication:

```
POST /itg/ppmservices/ProjectService HTTP/1.1
User-Agent: Crosscheck Networks SOAPSonar
Content-Type: text/xml; charset=utf-8
SOAPAction: "urn:GetProjectDetails"
Authorization: Basic YWRtaW46YWRtaW4=
Host: localhost:8088
Content-Length: 542
Expect: 100-continue
Connection: Keep-Alive
```

Web Service Authentication for Web Service Toolkit

In the Web service toolkit, the WS-Security configuration is defined in

```
<webservice_toolkit>/java/conf/client-axis2.xml.
```

Specify a User Through Configuration File

By default, as a PPM Center server the PPM Center Web service toolkit enables UsernameToken, Timestamp, and Encrypt.

Set Headers

In the client-axis2.xml file, the following "OutflowSecurity" section defines how WS-Security headers are set on an outgoing SOAP message generated by the toolkit.

```
<module ref="rampart" />
<parameter name="OutflowSecurity">
  <action>
    <items>UsernameToken Encrypt Timestamp</items>
    <user>admin</user>
    <passwordCallbackClass>examples.security.PasswordCallbackHandler
    </passwordCallbackClass>
    <passwordType>PasswordText
    </passwordType>
    <encryptionParts>{Element}{http://docs.oasis-open.org/
      wss/2004/01/oasis-200401-wss-wssecurity-secext-
      1.0.xsd}UsernameToken</encryptionParts>
    <encryptionUser>ppmservice</encryptionUser>
```

```
        <encryptionPropFile>client.properties
    </encryptionPropFile>
</action>
</parameter>
```

To specify a fixed user name for the web service call, simply change the value of the <user> element in the client-axis2.xml file.

Set Password

To set the password for this particular user, following these steps:

1. Open <webservice_toolkit>/java/
client/src/examples/security/**PasswordCallbackHandler.java**
2. Change the default password "admin" to the desired password.

Example:

```
public class PasswordCallbackHandler implements CallbackHandler
{
String username = null;

    public void handle(Callback[] callbacks) throws
IOException,
        UnsupportedCallbackException {
    for (int i = 0; i < callbacks.length; i++) {
        WSPasswordCallback callback =
            (WSPasswordCallback)callbacks[i];

        // obtain password. This can be customized to obtain
        // password from any desire source and apply any
        // necessary algorithm.
        //
        // if your logic requires the username, you can get
the
        // user name by:
        // String username = callback.getIdentifier();
        //
        String password = "admin";

        // set the obtained password
        callback.setPassword(password);
    }
}
}
```

Specify User

The following code describes how to specify the user through a configuration file.

```
String repositoryPath =  
System.getProperty("client.repository.dir");  
    String axis2 = repositoryPath + "/conf/clientaxis2.  
xml";  
    File file = new File(axis2);  
    if (file.exists()) {  
        try {  
            ctx = ConfigurationContextFactory  
.createConfigurationContextFromFileSystem(  
                repositoryPath, axis2);  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

HTTP Basic Authentication

The PPM Center Web service client can be programmed so that instead of WS-Security headers, you can use HTTP basic authentication headers to pass the user credential.

The following examples illustrate how to pass the user credential by using HTTP basic authentication headers:

```
public static void setHttpBasicAuthHeader(Stub stub, String  
username, String password) {  
    Options options = stub._getServiceClient().getOptions();  
    if (options == null) return;  
    HttpTransportProperties.Authenticator auth = new  
HttpTransportProperties.Authenticator();  
    auth.setUsername(username);  
    auth.setPassword(password);  
    auth.setPreemptiveAuthentication(true);  
    options.setProperty(HTTPConstants.AUTHENTICATE, auth);  
    stub._getServiceClient().setOptions(options);  
}
```

NTLM Authentication

The PPM Center Web service client can also be programmed to pass NTLM credentials.

The following example illustrates how to pass NTLM credentials:

```
public static void setNTLMAuthHeader(Stub stub,
    String domain,
    String username,
    String password,
    String host) {
    Options options = stub._getServiceClient().getOptions();
    if (options == null) return;
    HttpTransportProperties.Authenticator auth = new
    HttpTransportProperties.Authenticator();
    List schemes = new ArrayList();

    schemes.add(HttpTransportProperties.Authenticator.NTLM);
    auth.setAuthSchemes(schemes);
    auth.setPreemptiveAuthentication(true);
    auth.setUsername(username);
    auth.setPassword(password);
    auth.setDomain(domain);
    auth.setHost(host);
    options.setProperty(HTTPConstants.AUTHENTICATE, auth);
    stub._getServiceClient().setOptions(options);
}
```

Configure Web Service Client for HTTPS

To enable a Web service client for HTTPS, do the following:

1. Get the certification authority (CA) root certificate used by the Web server with which PPM Center is integrated.
2. Perform one of the following options to import the certificate into the key store.
 - Create a custom key store at the client side to import the CA root certificate into it as trusted CA certificate, and specify the key store for the Web service client.
 - Or, directly import the CA certificate into the JDK's default key store. In this option, you do not need to specify the key store for the Web service client.

Custom Key Store

After you obtain the CA root certificate used by the web server with which PPM Center is integrated, you can run the following command to import the certificate into your new key store.

```
keytool -import -keystore {keystore file} -alias {entryAlias} -file {certfile}
```

Notify the Web service client of the location of the certificate by using system property "javax.net.ssl.trustStore" as shown in the following example:

```
java -Dclient.repository.dir=%WSCLIENT_HOME% -classpath %CPATH%  
-Djavax.net.ssl.trustStore="C:/toolkit/java/conf/client.jks"  
examples.pm.ProjectServiceClient https://localhost:8443/itg/  
ppmservices/ProjectService "kevin8"
```

Or, you can use code inside your program as shown in the following example:

```
system.setProperty("javax.net.ssl.trustStore", "full-path-of-keystore-file" );
```

SSL-Client Authentication

If Secure Sockets Layer (SSL)-client authentication is required by the Web server, the `javax.net.ssl.keyStore` system property should be set to a key store file that contains the client's personal certificate.

Example:

```
java -Dclient.repository.dir=%WSCLIENT_HOME% -classpath %CPATH%  
-Djavax.net.ssl.trustStore="C:/toolkit/java/conf/client.jks"  
-Djavax.net.ssl.keyStore="C:/toolkit/java/conf/client.jks"  
examples.pm.ProjectServiceClient https://localhost:8443/itg/  
ppmservices/ProjectService "kevin8"
```

The personal key store and the trust key store can point to the same key store file or a different one.

JDK Default Key Store

Another place to keep the trusted CA certificate is in the JDK's default key store, which can be found at `$JRE_HOME/lib/security/cacerts`. The default password to the JDK key store is 'changeit'.

Use the following command to import the certificate into the JDK default key store:

```
keytool -import -trustcacerts -keystore {$JRE_HOME/lib/  
security/cacerts} -alias {entryAlias} -file {certfile}
```

Make sure you are updating the right version of JDK on your machine if multiple JDKs are installed.

Java Virtual Machine (JVM) will load this key store when the program starts, so no additional Web service client configuration is needed if the CA certificate is loaded into the key store.

Web Service Single Sign-On

As with many other PPM Center Web components, PPM Center Web services is able to integrate with most industry-standard single sign-on (SSO) systems such as CA SiteMinder, Oracle Identity Management, RSA Sign-On Manager, and IBM Tivoli Access Manager through pluggable authentication frameworks. PPM Center provides a log-in module for SiteMinder. For other SSO systems, additional customization may be required.

You can integrate with SiteMinder using the PPM Center SiteMinder Log-in Module. When this authentication mode is used, PPM Center authenticates users to SiteMinder, and does not store user passwords in the PPM Center database.

PPM Center Server Configuration

From the PPM Center server side, you can add the following parameters into the server.conf file:

- To allow SiteMinder Login Module to be invoked for Web service user authentication, set the following parameter:

```
com.kintana.core.server.ENABLE_WEBSERVICE_SSO=true
```

- To choose SiteMinder to be the authentication mode, set the following parameter:

```
com.kintana.core.server.authentication_mode=SiteMinder
```

Note: If SiteMinder is chosen as the only authentication mode, any individual user's authentication mode that was set through the workbench user page would be overwritten by this mode.

- To make PPM Center Web application use single sign-on mode, set the following parameter:

```
com.kintana.core.server.SINGLE_SIGN_ON_  
PLUGIN=com.kintana.sc.security.auth.SiteMinderSingleSignOn.
```

Integration with a Client-Side Log-In Module

To complement the integration with client-side log-in module, follow these steps:

1. Develop a Java Authentication and Authorization Service (JAAS) log-in module that authenticates with the SSO system and receives an SSO token. The token could be set as a private credential in the Subject class.

PPM Center has already provided such a module:

```
com.kintana.sc.security.auth.SiteMinderLoginModule
```

2. Create the JAAS configuration file under the \$WebServiceToolkit/java/conf directory

Example:

```
#authentication.conf
SiteMinder {
  com.kintana.sc.security.auth.SiteMinderLoginModule required
  debug=true;}

```

3. Specify the JAAS login configure system property in the command lines used to invoke the Web services in the compile_client.bat file

Example:

```
java -Dclient.repository.dir=%WSCLIENT_HOME% -classpath
%CPATH%
-Djava.security.auth.login.config==%WSCLIENT_HOME%/conf/
authentication.conf
examples.pm.ProjectServiceClient https://localhost:8443/itg/
ppmservices/ProjectService "kevin8"
```

4. Add logic in the Web service client to invoke JAAS login

Examples:

```
public Subject login() {
    LoginContext lc = null;
    lc = new LoginContext(
        " MyCustomModule ",
        myCallbackHandler
    );
    lc.login();
    return lc.getSubject();
}
```

5. Add the SSO token as cookie in the Web service client.

Examples: Set a HTTP cookie in axis2 Web service client.

```
public void setSSOCookie(Stub stub, String ssoToken) {
    List headers = new ArrayList();
```



```
//Set the required session variable for SSO system
Header header = new Header(
    "Cookie",
    "SMSESSION=" + ssoToken
);
headers.add(header);

ServiceClient client = stub._getServiceClient();
Options option = client.getOptions();
option.setProperty(HTTPConstants.HTTP_HEADERS,
headers);
}
```

6. Call the corresponding method to set the SSO cookie after a stub is created.
7. Make the desired Web service request with the SSO cookie you set.

Working with Proxy Servers and Proxy Authentication

In some cases, the Web service client must go through a proxy server to reach the PPM Center server. In this case, a proxy server must be specified when you establish a connection. Otherwise, a connection time out error or no connection exception will occur.

There are two solutions available (depending on whether you want all or some calls to go through the proxy):

- Configure proxy by using client-axis2.xml.
- Configure proxy through java code.

All Web Service Calls Go Through Proxy: Configure Proxy By Using client-axis2.xml

This configuration forces all Web service calls from the Web service toolkit to go through the proxy. This file locates in the following directory:

```
<webservice_toolkit>/java/conf
```

Add following XML configuration:

```
<transportSender
name=""
class="org.apache.axis2.transport.http.CommonsHTTPTransportSend
er">
    <parameter name="PROTOCOL" locked="false">HTTP/1.1</
parameter>
<parameter
    name="PROXY"
    proxy_host="proxy_host_name"
    proxy_port="proxy_host_port"
    locked="true">
        userName:domain:password
    </parameter>
</transportSender>
```

If authentication is not available, fill

"userName:domain:password" as "anonymous:anonymous:anonymous."

Some Web Service Calls Go Through Proxy: Configure Proxy By Using Java Code

If only a selected number of operations in the toolkit should go through the proxy, choose this solution.

Add the following method in the client code:

```
public void setProxy(Stub stub, String proxyHost, int
proxyPort) {
    // get options
    Options options = stub._getServiceClient().getOptions();
    if (options == null) {
        options = new Options();
        stub._getServiceClient().setOptions(options);
    }

    HttpTransportProperties.ProxyProperties proxyProperties = new
HttpTransportProperties.ProxyProperties();
    proxyProperties.setProxyName(proxyHost);
    proxyProperties.setProxyPort(proxyPort);
    options.setProperty(HTTPConstants.PROXY,
proxyProperties);
}
```

Call this method before you invoke a Web service operation. For example:

```
DemandServiceStub stub = new DemandServiceStub(ctx,
serviceURL);
```

```
setProxy(stub, "proxy.hp.com", 8888);  
...
```

Send documentation feedback

If you have comments about this document, you can [contact the documentation team](#) by email. If an email client is configured on this system, click the link above and an email window opens with the following information in the subject line:

Feedback on Web Services Programmer's Guide (Project and Portfolio Management Center 9.40)

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to your `_IE_team_PDL@hpe.com`.

We appreciate your feedback!