

HP Project and Portfolio Management Center

Software Version: 7.1

Service-Oriented Architecture: Web Services Guide

Document Release Date: December 2007

Software Release Date: December 2007 (SP5)



Legal Notices

This document, and the accompanying software and other documentation, is protected by U.S. and international copyright laws, and may be used only in accordance with the accompanying license agreement. Features of the software, and of other products and services of Mercury Interactive Corporation, may be covered by one or more of the following patents: United States: 5,511,185; 5,657,438; 5,701,139; 5,870,559; 5,958,008; 5,974,572; 6,137,782; 6,138,157; 6,144,962; 6,205,122; 6,237,006; 6,341,310; 6,360,332; 6,449,739; 6,470,383; 6,477,483; 6,549,944; 6,560,564; 6,564,342; 6,587,969; 6,631,408; 6,631,411; 6,633,912; 6,694,288; 6,738,813; 6,738,933; 6,754,701; 6,792,460 and 6,810,494. Australia: 763468 and 762554. Other patents pending. All rights reserved.

U.S. GOVERNMENT RESTRICTED RIGHTS. This Software Documentation is a “commercial item” as defined at 48 C.F.R. 2.101 (October 1995). In accordance with 48 C.F.R. 12.212 (October 1995), 48 C.F.R. 27.401 through 27.404 and 52.227-14 (June 1987, as amended) and 48 C.F.R. 227.7201 through 227.7204 (June 1995), and any similar provisions in the supplements to Title 48 of the C.F.R. (the “Federal Acquisition Regulation”) of other entities of the U.S. Government, as applicable, all U.S. Government users acquire and may use this Documentation only in accordance with the restricted rights set forth in the license agreement applicable to the Computer Software to which this Documentation relates.

Mercury, Mercury Interactive, the Mercury logo, the Mercury Interactive logo, LoadRunner, WinRunner, SiteScope and TestDirector are trademarks of Mercury Interactive Corporation and may be registered in certain jurisdictions. The absence of a trademark from this list does not constitute a waiver of Mercury's intellectual property rights concerning that trademark.

All other company, brand and product names may be trademarks or registered trademarks of their respective holders. Mercury disclaims any responsibility for specifying which marks are owned by which companies or which organizations.

Intel®, Intel® Itanium®, Intel® Xeon™, and Pentium® are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Java™ is a U.S. trademark of Sun Microsystems, Inc.

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

Oracle® is a registered U.S. trademark of Oracle Corporation, Redwood City, California.

UNIX® is a registered trademark of The Open Group.

Mercury provides links to external third-party Web sites to help you find supplemental information. Site content and availability may change without notice. Mercury makes no representations or warranties whatsoever as to site content or availability.

© 1997- 2008 Hewlett-Packard Development Company, L.P.

Documentation Updates

This manual's title page contains the following identifying information:

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

To check for recent updates, or to verify that you are using the most recent edition of a document, go to: http://ovweb.external.hp.com/lpe/doc_serv/.

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

Support

Mercury Product Support

You can obtain support information for products formerly produced by Mercury as follows:

- If you work with an HP Software Services Integrator (SVI) partner (www.hp.com/managementsoftware/svi_partner_list), contact your SVI agent.
- If you have an active HP Software support contract, visit the HP Software Support site and use the Self-Solve Knowledge Search to find answers to technical questions.
- For the latest information about support processes and tools available for products formerly produced by Mercury, we encourage you to visit the HP-Mercury Software Support web site at: support.mercury.com.
- Contact your HP Sales Representative if you have additional questions.

HP Software Support

You can visit the HP Software Support web site at www.hp.com/managementsoftware/services.

HP Software online support provides an efficient way to access interactive technical support tools. As a valued support customer, you can benefit by using the support site to:

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

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract. To find more information about access levels, go to: www.hp.com/managementsoftware/access_level.

To register for an HP Passport ID, go to: www.managementsoftware.hp.com/passport-registration.html.

Table of Contents

List of Tables	xi
Chapter 1: Getting Started with PPM Center Web Services	13
Overview of PPM Center Web Services.....	14
Web Service Special Commands.....	15
What's New in Web Services Version 7.1	15
New Web Services.....	16
Previously Available Web Services	16
Web Service Operations and Special Commands.....	17
Security Enhancements	20
Related Documentation.....	21
Chapter 2: Configuring Web Services on the PPM Server	23
Applicable Standards, Specifications, and Compatibilities.....	24
Overview of Web Service Security	24
Enabling Web Services	25
Optional Debug Logging Configurations.....	26
Enabling Debug Logging for Web Services	26
Disabling Debug Logging for Web Services.....	26
Optional Security Configurations	27
Enabling or Disabling WS-Security.....	27
Server-Based Security	27
Client-Based Security	28
Enabling or Disabling the WS-Security Timestamp	29
Configuring the WS-Security UsernameToken Profile.....	30
Specifying Username and Password for PPM Center Special Commands.....	30
Configuring WS-Security Encryption.....	31
Configuring the Keystore Password.....	32
Configuring HTTP Basic Authentication.....	33
Specifying Basic Authentication for PPM Center Web Services	33
Specifying Basic Authentication for PPM Center Special Commands	34
Configuring HTTPS Authentication	35
Creating the Keystore Using Keytool.....	36
Chapter 3: Understanding the Web Service Operations and Special Commands.....	37
PPM Center Application Modules with Web Service Support	38
HP Project Management Operations.....	39

addTasksToExistingWorkPlan	39
createBlankWorkPlan	39
createProject	39
createWorkPlanFromTemplate	39
executeWorkflowTransition	39
readTasks	39
importWorkPlanTasks	39
searchTasks	40
updateProject	40
updateTaskActuals	40
HP Resource Management Operations	40
createResourcePools	40
getResourceParticipation	40
getResourcePools	40
searchResourcePools	41
setResourceParticipation	41
updateResourcePools	41
HP Financial Management Operations	41
createCostRules	42
deleteCostRules	42
getCostFactors	42
getCostRules	42
searchCostRules	42
setCostFactors	43
updateCostRules	43
HP Financial Management Special Commands	43
HP Demand Management Operations	43
addRequestNotes	44
createRequest	44
deleteRequests	44
getRequests	44
setRequestFields	45
setRequestRemoteReferenceStatus	45
HP Demand Management Special Commands	45
ksc_export_request	46
Description	46
Syntax	46
Parameters	46
ksc_export_fields	48
Description	48
Syntax	48
Parameters	48
ksc_export_status_change	49
Description	49
Syntax	49

Parameters.....	49
HP Time Management Operations.....	50
getActualTime.....	50
HP Time Management Special Commands.....	53
ksc_export_actual_time.....	54
Description.....	54
Syntax.....	54
Parameters.....	54
Chapter 4: Developing Web Service Applications.....	57
Overview of the Development Process.....	58
Version 7.1 WSDL Module Elements.....	61
Web Service Definition Language Specification.....	61
XML Schema Definition Specification.....	62
Version 6.0 and 7.0 WSDL Module Elements.....	63
Web Service Definition Language Specification.....	63
Using PPM Center Special Commands.....	63
Using HP-Provided Special Commands.....	64
Problem.....	64
Solution.....	64
The R&D Workflow.....	66
Export Request.....	67
Export Request Status Change.....	68
Creating Custom Special Commands.....	69
Problem.....	69
Solution.....	69
ksc_local_exec Special Command.....	69
Process Example.....	70
Log File Example.....	73
Chapter 5: Introducing the Web Service Toolkit.....	75
Overview of the Web Service Toolkit.....	76
Toolkit Contents.....	76
java Directory Content.....	76
MicrosoftDotNet Directory Content.....	77
Generating New Stubs.....	78
Sample Java Client.....	78
Phase One: Set Up the Environment.....	79
Set the Environment Variables and Java CLASSPATH.....	79
Phase Two: Write the Client Code.....	80
Establish the Context.....	80
Create a Request Object.....	81
Invoke the Service.....	82

Process the Return Message	83
Set the HTTP Basic Authentication Header	84
Phase Three: Set Up the Client Security	84
Configure Client Security	84
Configure the Password Callback Handler	86
Phase Four: Compile the Client Code	86
Phase Five: Execute the Client Code	87
Sample .NET Client	87
Sample .NET Process	88
Appendix A: HP Project Management Web Service Files	91
Version 7.1 WSDL File	91
Version 7.1 XSD File	96
Appendix B: HP Resource Management Web Service Files	103
Version 7.1 WSDL File	103
Version 7.1 XSD File	107
Appendix C: HP Financial Management Web Service Files	111
Version 7.1 WSDL File	111
Version 7.1 XSD File	116
Versions 6.0 and 7.0 WSDL File	117
Appendix D: HP Demand Management Web Service Files	127
Version 7.1 WSDL File	127
Version 7.1 XSD File	131
Versions 6.0 and 7.0 WSDL File	132
Appendix E: HP Time Management Web Service Files	143
Version 7.1 WSDL File	143
Version 7.1 XSD File	144
Appendix F: Shared Web Service Files	145
Version 7.1 Shared XSD File	145
Appendix G: Outdated Web Services	147
Overview of the Outdated Web Services	148
HP Financial Management Deprecated Operations	149
create	149
read	153

update	155
HP Demand Management Deprecated Operations.....	160
getRequestTypesByFieldGroup.....	160
getRequestTypeFields.....	161
HP Demand Management Re-implemented Operations.....	161
HP Demand Management Re-Implemented Special Commands	162
Index	163

List of Tables

Table 1-1	HP Project Management application module.....	17
Table 1-2	HP Resource Management application module	18
Table 1-3	HP Financial Management application module	18
Table 1-4	HP Demand Management application module	19
Table 1-5	HP Time Management application module	20
Table 3-1	ksc_export_actual_time parameters	54

1 Getting Started with PPM Center Web Services

In This Chapter:

- *Overview of PPM Center Web Services*
 - *Web Service Special Commands*
 - *What's New in Web Services Version 7.1*
 - *New Web Services*
 - *Previously Available Web Services*
 - *Web Service Operations and Special Commands*
 - *Security Enhancements*
 - *Related Documentation*
-

Overview of PPM Center Web Services

Organizations are facing an increasing need to integrate various systems and applications. Web service technology offers a standardized way of achieving such integration. This guide describes the Web services available for you to use in your HP Project and Portfolio Management Center (PPM Center) deployment. Using PPM Center Web services provides significant benefits over the Open Interface and other types of direct PPM Center database integration, and helps you efficiently achieve your strategic integration initiatives.

PPM Center includes Web service support for the following application modules:

- **HP Project Management.** Enables organizations to streamline and standardize the management of project activities using a template-based, collaborative approach.
- **HP Resource Management.** Enables enterprises to effectively manage resource capacity and allocation.
- **HP Financial Management.** Offers automatic real-time calculations of costs and variances, resulting in detailed comparisons of project health.
- **HP Time Management.** Allows resources (users) to use time sheets to report how long they worked on specific work items.
- **HP Demand Management.** Manages requests from creation to implementation. Each request is processed using a workflow that is represented graphically in the user interface.

The supported Web service operations typically include the following categories of services:

- **Create.** Used to create an entity.
- **Get.** Used to return (read) an entity.
- **Set.** Used to revise or remove (update or possibly delete) a field in an entity.
- **Add.** Used to add data to an entity.
- **Search.** Used to find a specified detail in an entity.
- **Import/Export.** Used to import the data into (or out of) PPM Center.

Web Service Special Commands

Selected Web service operations have been incorporated in PPM Center as special commands. These commands are executed through the PPM Workbench. Web service operations can also be executed from workflows in the PPM Workbench.

What's New in Web Services Version 7.1

Initial Version 7.1 Release

PPM Center version 7.1 Web services feature:

- Architectural re-design to use the Axis2 framework
- Security enhancements that conform to the WS-Security standard
- New functionality for several application modules
- Improved implementations for many of previously available Web service operations and special commands



Customers upgrading from Mercury IT Governance Center version 6.0 and PPM Center version 7.0 are encouraged to migrate to the new Axis2 architecture as soon as possible.

Service Pack 5

PPM Center version 7.1 Service Pack 5 includes the following changes:

- Web Service Toolkit enhancements:
 - The Toolkit can be downloaded from your PPM Server (instead of the Download Center).
 - The Toolkit includes .NET examples.
 - The Java™ examples have been expanded.
- The HP Project Management Web services have been updated to correct defect 196268 (SR 1-692995593). This defect is in regards to projects with tasks whose status is Pending Predecessor; the WSDL file was missing the pending-predecessor item in the definition of the task status.

New Web Services

Web service support is now available for the following application modules:

- HP Project Management
- HP Resource Management

New operations are also available for the following application modules:

- HP Demand Management
- HP Time Management
- HP Financial Management

Previously Available Web Services

Web service operations and special commands available in prior versions of PPM Center may have been:

- **Re-implemented.** Many of the version 6.0 and 7.0 Web services have been re-implemented to take advantage of Axis2 security features.

Although you may continue to use these version 6.0 and 7.0 Web services, HP strongly recommends that you convert to the new implementation as soon as practical. Eventually, these versions will be deleted and no longer available for your use.

If you are creating new Web service applications, use the version 7.1 Web services.

- **Deprecated.** Some of the version 6.0 and 7.0 Web services are minimally used by PPM Center customers, or have had their functionality incorporated into version 7.1 Web services.

These Web services have been retained for interim purposes only. Eventually, these Web services will be deleted and no longer available. Therefore, you should not design long-term solutions that are reliant on these Web services.

See [Appendix G, *Outdated Web Services*, on page 147](#) for descriptions and alternatives.

- **Replaced.** Some of the version 6.0 and 7.0 Web services have been replaced with a version 7.1 implementation and the version 6.0 and 7.0 Web service is no longer available for your use.

- **Deleted.** Some of the version 6.0 and 7.0 Web services are no longer used and, consequently, are not available in version 7.1.

For details about the Mercury IT Governance Center version 6.0 and PPM Center version 7.0 Web services, see the version 6.0 *Service-Oriented Architecture: Web Services Guide*.



If you have upgraded from Mercury IT Governance Center version 6.0 or PPM Center version 7.0 and your organization relies on any of the deprecated or deleted Web services, notify HP Software Support.

Web Service Operations and Special Commands

Identification and availability of the operations and special commands available in this and prior versions of PPM Center are shown in *Table 1-1* through *Table 1-5*.

Table 1-1. HP Project Management application module

	Versions 6.0 & 7.0	Version 7.1
Web Service Operations		
addTaskToExistingWorkPlan	Not available	Added
createBlankWorkPlan	Not available	Added
createProject	Not available	Added
createWorkPlanFromTemplate	Not available	Added
executeWorkflowTransition	Not available	Added
importWorkPlanTasks	Not available	Added
readTasks	Not available	Added
searchTasks	Not available	Added
updateProject	Not available	Added
updateTaskActuals	Not available	Added
Special Commands		
	None	None

Table 1-2. HP Resource Management application module

	Versions 6.0 & 7.0	Version 7.1
Web Service Operations		
createResourcePools	Not available	Added
getResourceParticipation	Not available	Added
getResourcePools	Not available	Added
searchResourcePools	Not available	Added
setResourceParticipation	Not available	Added
updateResourcePools	Not available	Added
Special Commands		
	None	None

Table 1-3. HP Financial Management application module

	Versions 6.0 & 7.0	Version 7.1
Web Service Operations		
create	Included	Deprecated
read	Included	Deprecated
update	Included	Deprecated
createCostRules	Not available	Added
deleteCostRules	Not available	Added
getCostFactors	Not available	Added
getCostRules	Not available	Added
searchCostRules	Not available	Added
setCostFactors	Not available	Added
updateCostRules	Not available	Added
Special Commands		
ksc_export_budget	Included	Deleted
ksc_export_budget_as_xml	Included	Deleted

Table 1-4. HP Demand Management application module

	Versions 6.0 & 7.0	Version 7.1
Web Service Operations		
deleteRequests	Included	Re-implemented with the same name
getRequestTypeFields	Included	Deprecated
getRequestTypesByFieldGroup	Included	Deprecated
importRequest	Included	Re-implemented as createRequest
updateRemoteReference	Included	Re-implemented as setRequestRemoteReferenceStatus
addRequestNotes	Not available	Added
createRequest	Not available	Added
getRequests	Not available	Added
setRequestFields	Not available	Added
setRequestRemoteReferenceStatus	Not available	Added
Special Commands		
ksc_export_fields	Included	Included
ksc_export_request	Included	Re-implemented
ksc_export_status_change	Included	Included

Table 1-5. HP Time Management application module

	Versions 6.0 & 7.0	Version 7.1
Web Service Operations		
read	Included	Replaced
getActualTime	Not available	Added
Special Commands		
ksc_export_actual_time	Included	Replaced
ksc_export_actual_time_as_xml	Included	Deleted

Security Enhancements

Security for Web services in version 7.1 of PPM Center have been enhanced as follows:

- Authentication through WS-Security in the SOAP header now supports:
 - Standard UsernameToken with username and plain-text password.
 - UsernameToken with username and password in digest.
 - Encryption of UsernameToken based on Web service security standard.
- Authentication through HTTP headers currently supports:
 - Username and clear-text password in the HTTP headers using the HTTP basic authentication model.
- Authorization of Web services now supports:
 - The standard PPM Center authorization model.

Related Documentation

The following documentation provides additional information about PPM Center:

- *What's New and What's Changed*
- *Getting Started*
- *Open Interface Guide and Reference*
- *HP Program Management User's Guide*
- *HP Portfolio Management User's Guide*
- *HP Project Management User's Guide*
- *HP Resource Management User's Guide*
- *HP Time Management User's Guide*
- *HP Demand Management User's Guide*
- *HP Financial Management User's Guide*
- *Commands, Tokens, and Validations Guide and Reference*

2 Configuring Web Services on the PPM Server

In This Chapter:

- *Applicable Standards, Specifications, and Compatibilities*
 - *Overview of Web Service Security*
 - *Enabling Web Services*
 - *Optional Debug Logging Configurations*
 - *Enabling Debug Logging for Web Services*
 - *Disabling Debug Logging for Web Services*
 - *Optional Security Configurations*
 - *Enabling or Disabling WS-Security*
 - *Enabling or Disabling the WS-Security Timestamp*
 - *Configuring the WS-Security UsernameToken Profile*
 - *Configuring WS-Security Encryption*
 - *Configuring the Keystore Password*
 - *Configuring HTTP Basic Authentication*
 - *Configuring HTTPS Authentication*
-

Applicable Standards, Specifications, and Compatibilities

The following standards and environments apply to the Web services implementation:

- PPM Center version 7.1 uses the Axis2 Web service framework to leverage the following capabilities:
 - Extensible and modular architecture
 - Very easy-to-define custom message routing
 - Easy addition framework for new WS-* frameworks
- PPM Center version 7.1 Web services support the following standards:
 - Document/Literal model
 - SOAP 1.1 and 1.2
 - WSDL 1.1 and 2.0 (with some caveats)
 - WS-Security

See the *System Requirements and Compatibility Matrix* for a complete and current list of PPM Center requirements.

Overview of Web Service Security

Web service security is configured through a set of files that are on the client *and* the server. Whichever style of security is implemented, it must be identical on both the server side and the client side. The security features supported in PPM Center Web services include the following:

- Authentication
 - HTTP basic authentication
 - WS-Security implemented:
 - Through the SOAP message header
 - Using the UsernameToken profile as defined by the OASIS standard (docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0.pdf)

- Authorization
 - Service layer authorization

On the PPM Server, security features are configured in the `server.conf`, `webservices.conf`, `client_axis2.xml`, and `axis2.xml` files. By default, the UsernameToken profile with clear-text password is used and the UsernameToken header is encrypted.

On the client server, configuration is set in the Web service application code that makes the requesting call to PPM Center. The implementation options are through:

- HTTP basic authentication
- SOAP header
- HTTPS

Enabling Web Services

Although automatically installed with PPM Center, Web services must be enabled before they can be used.

Web services are disabled by default.

To enable PPM Center for Web services:

1. (Optional, although highly recommended) Create a backup copy of the `<PPM_Install>/server.conf` file where `<PPM_Install>` represents the location where your PPM Center instance is installed.
2. Open the `server.conf` file for edit.
3. Enable Web services calls by adding (or modifying) the following setting:

```
com.kintana.core.server.ENABLE_WEB_SERVICES=true
```
4. Save and close the `server.conf` file.
5. Stop, then restart the PPM Server.

Optional Debug Logging Configurations

PPM Center Web services debug logging can be enabled and disabled. See the following sections for information on enabling and disabling debug logging.

Enabling Debug Logging for Web Services

Debug logging is only needed to help troubleshoot serious problems.

By default, debug logging is disabled.

To enable Web services logging, complete the following steps:

1. Open the `<PPM_Install>/conf/logging.conf` file for edit.
2. Change the logging threshold by verifying (or modifying) the following setting:

```
com.kintana.core.logging.SYSTEM_THRESHOLD = DEBUG
```

If you change the value, make note of its current setting.

3. Enable the logging level for Web services by adding (or modifying) the following setting.

```
com.kintana.core.logging.PRODUCT_FUNCTION_LOGGING_LEVEL =  
com.mercury.itg.ws, DEBUG
```

4. Save and close the `logging.conf` file.
5. Stop, then restart the PPM Server.

Disabling Debug Logging for Web Services

Once your troubleshooting is complete, Web service debug logging should be disabled.

To disable Web services logging, complete the following steps:

1. Open the `<PPM_Install>/conf/logging.conf` file for edit.
2. If you changed the logging threshold in [step 2](#), re-instate the previous setting. This parameter is typically set as follows:

```
com.kintana.core.logging.SYSTEM_THRESHOLD = ERROR
```

3. Disable the logging level for Web services by commenting out the parameter as follows:

```
# com.kintana.core.logging.PRODUCT_FUNCTION_LOGGING_LEVEL =  
com.mercury.itg.ws, DEBUG
```

4. Save and close the `logging.conf` file.
5. Stop, then restart the PPM Server.

Optional Security Configurations

PPM Center Web services security and their options can be enabled and disabled. To configure PPM Center Web service security options, complete the steps described in the following sections. The configurable security options include:

- *Enabling or Disabling WS-Security*
- *Enabling or Disabling the WS-Security Timestamp*
- *Configuring the WS-Security UsernameToken Profile*
- *Configuring WS-Security Encryption*
- *Configuring the Keystore Password*
- *Configuring HTTP Basic Authentication*
- *Configuring HTTPS Authentication*

Enabling or Disabling WS-Security

By default, WS-Security with encrypted UsernameToken profile is enabled.

Server-Based Security

To enable or disable WS-Security on the PPM Server, complete the following steps:

1. On the PPM Server, open the `<PPM_Install>/server/<PPM_ServerName>/deploy/itg.war/WEB-INF/conf/axis2.xml` file for edit, where

`<PPM_ServerName>` 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 host name of the server.)

2. Locate the `InflowSecurity` parameter, similar to the following:

```
<parameter name="InflowSecurity">
  <action>
    <items>UsernameToken Timestamp Encrypt
    </items>
    <passwordCallbackClass>
      com.mercury.itg.ws.core.
      handlers.security.passwordCallbackHandler
    </passwordCallbackClass>
    <decryptionPropFile>service.properties
    </decryptionPropFile>
  </action>
</parameter>
```

If the text is *not* commented out (as shown in the example), WS-Security is enabled on the server side.

If the text is commented out, WS-Security is disabled on the server side.

3. Add or remove comment characters to specify the desired server-based security setting.
4. Save and close the `axis2.xml` file.

Client-Based Security

To enable or disable *from* the PPM Server, complete the following steps:

1. On the PPM Server, open the `<PPM_Install>/server/<PPM_ServerName>/deploy/itg.war/WEB-INF/conf/client_axis2.xml` file for edit.
2. Locate the `OutflowSecurity` parameter, similar to the following:

```
<parameter name="OutflowSecurity">
  <action>
    <items>UsernameToken Timestamp Encrypt</items>
    <passwordCallbackClass>
      com.mercury.itg.ws.core.handlers.security.
      OutboundpasswordCallbackHandler
    </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>
```

If the text is *not* commented out (as shown in the example), WS-Security is enabled on the client side.

If the text is commented out, WS-Security is disabled on the client side.

3. Add or remove comment characters to specify the desired client-based security setting.
4. Save and close the `client_axis2.xml` file.

Enabling or Disabling the WS-Security Timestamp

When WS-Security is used, the server and client must have the same configuration setting indicating whether or not to include timestamp information. By default, the timestamp setting is enabled.

To enable or disable timestamp use, complete the following steps:

1. On the PPM Server, open the `<PPM_Install>/server/<PPM_ServerName>/deploy/itg.war/WEB-INF/conf/axis2.xml` file for edit.
2. Locate the `InflowSecurity` parameter.
3. To:
 - Enable timestamp use, include the `Timestamp` setting in the list of items (as shown in [step 2 on page 27](#)).
 - Disable timestamp use, remove the timestamp setting in the list of items as follows:

```
<items>UsernameToken Encrypt</items>
```
4. Save and close the `axis2.xml` file.
5. On the PPM Server, open the `<PPM_Install>/server/<PPM_ServerName>/deploy/itg.war/WEB-INF/conf/client_axis2.xml` file for edit.
6. Locate the `OutflowSecurity` parameter.
7. To:
 - Enable timestamp use, include the `Timestamp` setting in the list of items (as shown in [step 2 on page 28](#)).
 - Disable timestamp use, remove the timestamp setting as previously shown
8. Save and close the `client_axis2.xml` file.

Configuring the WS-Security UsernameToken Profile

PPM Center supports the WS-Security UsernameToken profile. The UsernameToken profile (as specified by WS-Security) sends the username and password through SOAP headers. The password has two format options:

- **Clear-text password.** Sends the password to the Web service application and the application is responsible for performing the authentication.
- **Digest password.** Asks the Web service application for user password and then validates the returned “digest.”

Specifying Username and Password for PPM Center Special Commands

To specify the username, password, and Axis framework used to communicate with the PPM Server for the `ksc_export_XXX` commands, complete the following steps:

1. On the PPM Server, open the `<PPM_Install>/conf/webservices.conf` file for edit.
2. In the `remoteServer` tag and specify the following values:

```
<remoteServer baseURL="http://<Hostname>:<Port>/<Path>"
  username="<Username>"
  password="<Password>"
  encUsername="ppmservice"
  axisVersion="<AxisVersion>"
/>
```

where

<code><Hostname></code>	represents the host name of the PPM Server.
<code><Port></code>	represents the port number for Web access.
<code><Path></code>	represents the path where PPM Center special commands are accessed (via the Web).

- `<Username>` represents the username for the PPM Server administrative account.
- `<Password>` represents the password for the PPM Server administrative account.
- `<AxisVersion>` represents the Axis framework version.

Valid values are:

- AXIS1
- AXIS2

The default value is AXIS1 and is used for version 6.0 and 7.0 Web services. Specify AXIS2 for version 7.1 Web services.

For example:

```
<remoteServer baseUrl="http://localhost:8080/itg/
  ppmservices/DemandService"
  username="admin"
  password="admin"
  encUsername="ppmservice"
  axisVersion="AXIS2"
/>
```

3. (Optional) Repeat [step 2](#) if you want to enable special commands for additional application modules or remote systems.
4. Save and close the `webservices.conf` file.

Configuring WS-Security Encryption

Support for username, password, and the SOAP body encryption means that when the SOAP message travels over network, the credential information is encrypted.

To configure WS-Security encryption, complete the following steps:

1. Open the `<PPM_Install>/server/<PPM_ServerName>/deploy/itg.war/WEB-INF/conf/client_axis2.xml` file for edit.
2. In the `encryptionPropFile` tag, specify the name of the encryption property file.

The property file must be in the Java CLASSPATH when executing the client program.

For example:

```
<encryptionPropFile>client.properties</encryptionPropFile>
```

The keystore file that specifies the username and password is specified in the property file.

The keystore file must also be in the Java CLASSPATH when executing the client program.

3. Save and close the `client_axis2.xml` file.
4. Open the `<PPM_Install>/server/<PPM_ServerName>/deploy/itg.war/WEB-INF/conf/axis2.xml` file for edit.
5. In the `dencryptionPropFile` tag, specify the name of the encryption property file.

The property file must be in the Java CLASSPATH when executing the client program.

6. Save and close the `axis2.xml` file.

Configuring the Keystore Password

The WS-Service encryption functionality uses an alias name that is set in the Java keystore file.

For additional information, see the Axis2/rampart documentation at ws.apache.org/axis2/modules/rampart/1_0/security-module.html.

To configure the keystore password, complete the following steps:

1. Open the `<PPM_Install>/server/<PPM_ServerName>/deploy/itg.war/WEB-INF/conf/service.properties` file for edit.
2. For the `org.apache.ws.security.crypto.merlin.keystore.password` parameter, specify the keystore file name.

For example:

```
org.apache.ws.security.crypto.merlin.keystore.password=ppmservice
```

3. Save and close the `service.properties` file.
4. Open the `<PPM_Install>/server/<PPM_ServerName>/deploy/itg.war/WEB-INF/conf/client.properties` file for edit.

5. For the `org.apache.ws.security.crypto.merlin.keystore.password` parameter, specify the keystore file name.

For example:

```
org.apache.ws.security.crypto.merlin.keystore.password=ppmclient
```

6. Save and close the `client.properties` file.

Configuring HTTP Basic Authentication

HTTP basic authentication is another popular way of sending the username and password across the network for authentication.

Whether to accept username and password from HTTP basic authentication is specified on the PPM Server. The PPM Center Web services can be configured to perform authentication using HTTP basic authentication in the SOAP message header.

Specifying Basic Authentication for PPM Center Web Services

To enable HTTP basic authentication, complete the following steps:

1. On the PPM Server, open the `<PPM_Install>/server/<PPM_ServerName>/deploy/itg.war/WEB-INF/conf/axis2.xml` file for edit.

2. Add the following parameter to the file.

```
<parameter name="InFlowBasicAuth">  
  <Enforced>true</Enforced>  
</parameter>
```

3. Specify the desired authentication requirements:

- When this parameter is set to `true`, HTTP basic authentication headers are required by PPM Center Web services.
- When this parameter is set to `false`, any username and password information passed in the SOAP message header is ignored.

4. Save and close the `axis2.xml` file.

Specifying Basic Authentication for PPM Center Special Commands

To enable HTTP basic authentication for the `ksc_export_XXX` special commands, complete the following steps:

1. On the PPM Server, open the `<PPM_Install>/conf/webservice.conf` file for edit.
2. Add the following fields and their values to the `remoteServer` tag. (This parameter may have already had some updates as detailed in *Specifying Username and Password for PPM Center Special Commands* on page 30.)

```
httpAuthEnabled = "true|false"
httpAuthUsername = "<HttpUser>"
httpAuthPassword = "<HttpPassword>"
httpAuthDomain = "<httpDomain>"
```

where

<code><HttpUser></code>	represents the HTTP username. If this field is left empty, the <code>username</code> value is used.
<code><HttpPassword></code>	represents the HTTP password. If this field is left empty, the <code>password</code> value is used.
<code><httpDomain></code>	represents the domain name for HTTP basic credentials.

For example:

```
<remoteServer baseURL="http://localhost:8080/itg/
ppmservices/DemandService"
username="admin"
password="admin"
encUsername="ppmservice"
httpAuthEnabled = "true"
httpAuthUsername = "HttpUser"
httpAuthPassword = "HttpPassword"
httpAuthDomain = "httpDomain"
axisVersion="AXIS2"
/>
```

3. Save and close the `webservice.conf` file.

Configuring HTTPS Authentication

The PPM Center `ksc_export_XXX` special commands support HTTPS through JSSE. The standard trust store retrieval path for JSSE is followed for HTTPS connections.



For more details on JSSE, see java.sun.com/j2se/1.4.2/docs/guide/security/jsse/JSSERefGuide.html.

To enable HTTPS on the PPM Server, see the *System Administration Guide and Reference*.

To send `ksc_export_XXX` special command requests in SSL mode, complete the following steps:

1. On the PPM Server, open the `<PPM_Install>/server.conf` file for edit.
2. Add (or modify) the following three parameters:

```
com.kintana.core.server.WEB_SERVICES_SSL_KEYSTORE=<Keystore>
```

```
com.kintana.core.server.WEB_SERVICES_SSL_KEYSTORE_
PASSWORD=<Password>
```

```
com.kintana.core.server.WEB_SERVICES_SSL_
TRUSTSTORE=<Truststore>
```

where

`<Keystore>` represents the keystore file.

`<Password>` represents the password.

`<Truststore>` represents the truststore.

If no separate trust store is specified, the `WEB_SERVICES_SSL_KEYSTORE` can be used as a trust store.

3. Save and close the `server.conf` file.

Creating the Keystore Using Keytool

There are several methods for creating a keystore, using keytool is only one of them.

If PPM Center is used to call Web services on a remote server using HTTPS, import the certificate that was used to sign the remote server's SSL certificate into the JRE's trusted keystore.

This is required if the certification authority (CA) is not one of the known certificate authorities that ship with the Java Runtime Environment (such as Verisign).

If you use another certification authority, such as an authority internal to your organization:

- Import the certificate into the keystore, using the following command:

```
keytool -import -trustcacerts -alias systemca
-file <CA_certificate>
-keystore <JRE_home>/lib/security/jssecacerts
-storepass <password>
```

where

<CA_certificate>	represents the name of the file containing the certificate (from the certification authority) used to sign the remote server's SSL certificate
<JRE_home>	represents the location of the Java Runtime Environment installation used for the local PPM Center instance
<password>	represents the password used for the trusted certificate keystore

If the `jssecacerts` keystore does not already exist, a new one is created with the password you specified.

- See java.sun.com/j2se/1.4.2/docs/tooldocs/windows/keytool.html for more information about the keytool utility.



When using HTTPS, the base URL specified in the `webservices.conf` and `server.conf` files should use the `https://` prefix instead of `http://`.

3 Understanding the Web Service Operations and Special Commands

In This Chapter:

- *PPM Center Application Modules with Web Service Support*
- *HP Project Management Operations*
 - *addTasksToExistingWorkPlan*
 - *createBlankWorkPlan*
 - *createProject*
 - *createWorkPlanFromTemplate*
 - *executeWorkflowTransition*
 - *readTasks*
 - *importWorkPlanTasks*
 - *searchTasks*
 - *updateProject*
 - *updateTaskActuals*
- *HP Resource Management Operations*
 - *createResourcePools*
 - *getResourceParticipation*
 - *getResourcePools*
 - *searchResourcePools*
 - *setResourceParticipation*
 - *updateResourcePools*
- *HP Financial Management Operations*
 - *createCostRules*
 - *deleteCostRules*
 - *getCostFactors*
 - *getCostRules*
 - *searchCostRules*
 - *setCostFactors*
 - *updateCostRules*
- *HP Financial Management Special Commands*
- *HP Demand Management Operations*
 - *addRequestNotes*
 - *createRequest*
 - *deleteRequests*
 - *getRequests*
 - *setRequestFields*
 - *setRequestRemoteReferenceStatus*
- *HP Demand Management Special Commands*

- *ksc_export_request*
 - *ksc_export_fields*
 - *ksc_export_status_change*
 - *HP Time Management Operations*
 - *getActualTime*
 - *HP Time Management Special Commands*
 - *ksc_export_actual_time*
-

PPM Center Application Modules with Web Service Support

Currently the PPM Center Web service infrastructure supports the following PPM Center application modules:

- **HP Project Management.** Enables organizations to streamline and standardize the management of project activities using a template-based, collaborative approach. It enables collaborative HP Project Management, from concept to completion for both repetitive projects and one-time projects. It integrates project and process control, using task-level workflows.
- **HP Resource Management.** Enables you to effectively manage resource capacity and allocation. It balances your resource supply, including both staffing levels and skill base, with incoming demand, giving you full visibility and control over project demand as well as day-to-day demand.
- **HP Financial Management.** Sets cost rules and resource pools to programmatically perform automatic real-time calculations of costs and variances, giving you detailed comparisons of project health. It supports bulk operations, such as the bulk creation of cost rules. Use HP Financial Management to import the data into PPM Center at regular intervals to keep the PPM Center data up to date. It also can be used to export PPM Center data into another system or even to create a different interface for manipulating PPM Center data.
- **HP Demand Management.** Manages requests from creation to implementation. Each request is processed using a workflow. It can follow complex business rules by using different approval methods. Email and pager notifications can be generated as the request passes through the various workflow steps. Prioritization and delegation features allow requests to efficiently advance through the workflows applied to them, routing them to the relevant department, group, or individual.

- **HP Time Management.** Allows resources (people) to use time sheets to report how long they worked on specific work items, which include tasks, requests, packages, or other miscellaneous types of work. Planners or operations managers assign work items to specific resources. It bridges the gap between work management and HP Time Management across a range of items, including projects, tasks, requests, and deployments.

HP Project Management Operations

HP Project Management Web service operations provide access to the following project planning and work plan activities.

This functionality is new to PPM Center version 7.1.

`addTasksToExistingWorkPlan`

Add a list of new tasks to an existing work plan.

`createBlankWorkPlan`

Create an entirely new work plan with only the root task.

`createProject`

Create a new PPM Center project on PPM Center instance.

`createWorkPlanFromTemplate`

Create an entirely new work plan from a PPM Center template

`executeWorkflowTransition`

Execute a workflow transition. Note that this transition must be visible in the project details view.

`readTasks`

Get the detail information for a list of tasks.

`importWorkPlanTasks`

Load the specified work plan into PPM Center.

searchTasks

Get task IDs related to the given search preference. They may be from different work plans.

updateProject

Update the following details for a specified project:

- Project details
- Linked staffing profile, budget, or benefit

You cannot update the details of the individual staffing profile, budget, or benefit. However, you can use this operation to associate the project with a different a staffing profile, budget, or benefit.

updateTaskActuals

Update task actuals for a list of tasks.

HP Resource Management Operations

HP Resource Management Web service operations provide access to the following resource management activities.

This functionality is new to PPM Center version 7.1

createResourcePools

Create a new resource pool with specified properties. The input is an ordered list of resource pool items and the output is a correspondingly ordered list of the primary keys of the newly created entities.

getResourceParticipation

Search for the participation in one or more resource pools of a set of resources. Add and remove resources from resource pools as well as change resources' participation.

getResourcePools

Return the distribution among resource pools of a set of resources identified by the ID. The returned collection contains data for every resource. This includes

identifying the resource and a reference to the resource pool, the start and end dates of the resource's association with the pool and the resource's participation as a percentage.

searchResourcePools

Search the system for resource pools that meet the specified criteria. The set of filters for a resource pool search are: name, description, parent, resource pool managers, region and org unit. All filter fields are single-valued.

setResourceParticipation

Manage the participation in one or more resource pools of a set of resources. Add and remove resources from resource pools as well as change resources' participation.

updateResourcePools

Updates existing resource pools based on the specified properties. The primary key is used to locate the resource pool to update. Every specified property is written to the resource pool. The only exception to this rule are the non-nullable properties of resource pools. These include resource pool name, notes (notes may only be added, never deleted), access control list and region. Attempts to set these to null causes an error.

HP Financial Management Operations

HP Financial Management Web service operations provide access to costing activities.

Versions 6.0 and 7.0

Mercury IT Governance Center version 6.0 introduced Web service operations that allowed external systems to manipulate budget data for HP Financial Management.

These version 6.0 and 7.0 operations (`create`, `read`, and `update`) are available in version 7.1. See [Appendix G, *Outdated Web Services*, on page 147](#) for details.



These Web service operations have been deprecated; however, HP plans to re-implement these in a future release.

Version 7.1

PPM Center version 7.1 added Web service support for cost rules and cost factors in HP Financial Management. Changes to cost rules and cost factors trigger the recalculation of costs of affected entities in the system. Changes are allowed only if no conflicts exist among the resulting set of cost rules.

This new functionality relies on the Axis2 framework and is described in the following sections.

createCostRules

Create new cost rules. The input is an ordered list of cost rule items and the output is a correspondingly ordered list of the primary keys of the newly created entities. All properties are validated before written. Selected properties are non-nullable.

deleteCostRules

Delete a set of cost rules identified by primary key. Return the number of rules actually deleted. Attempts to delete rules that do not exist are ignored and do not trigger an error.

getCostFactors

Return the list of cost factors selected for use ordered by precedence from highest to lowest.

getCostRules

Return a set of cost rules that match a certain criteria. The filter specifies values for cost factors and also includes an effective range filter. All filter fields are single-valued. It is an error to specify a filter value for a cost factor that is not among the selected cost factors.

searchCostRules

Return the list of cost rules selected for use ordered by precedence from highest to lowest.

setCostFactors

Set the cost factors to be used by PPM Center. The order of factors in the list determines their precedence. By manipulating the content of the list you can remove existing cost factors or add new ones. Cost factors can be removed.

updateCostRules

Update a set of cost rules. The changed rules are subject to the same validation that occurs during rule creation. The primary key is used to locate the cost rule to update. Every property of the item is written to the cost rule. Some properties are non-nullable and attempts to set them to null causes an error. The rule number cannot be changed once a rule is created.

HP Financial Management Special Commands

Mercury IT Governance Center version 6.0 introduced special commands that allowed interactions using the PPM Workbench interface.

All of these special commands (`ksc_export_budget` and `ksc_export_budget_as_xml`) have been deleted in version 7.1.



If your organization currently relies on any of these deleted special commands, notify HP Software Support.

HP Demand Management Operations

HP Demand Management Web service operations provide access to workflow and services activities.

Versions 6.0 and 7.0

Mercury IT Governance Center version 6.0 introduced Web service operations that allowed external systems to manipulate various aspects of PPM Center requests.

These version 6.0 and 7.0 operations (`getRequestTypeFields` and `getRequestTypeByFieldGroup`) are available in version 7.1. See [Appendix G, *Outdated Web Services*, on page 147](#) for details.



These operations have been retained for interim purposes only. Eventually, these operations will be deleted and no longer available for your use.

If your organization currently relies on any of these deprecated Web service operations, notify HP Software Support.

The remaining operations (`deleteRequests`, `importRequest`, and `updateRemoteReference`) have been re-implemented using the Axis 2

framework. See `deleteRequests`, `createRequest`, and `setRequestRemoteReferenceStatus` in the following sections.



As an interim solution, you may continue to use Web service applications using these version 6.0 and 7.0 Web service operations. However, you should convert these applications to use the re-implemented operations at your earliest convenience.

Version 7.1

With the release of PPM Center version 7.1, HP Demand Management Web service support was expanded and relies on the Axis2 framework, as described in the following sections.

`addRequestNotes`

Allows an external system to add a list of notes to the request in HP Demand Management, identified by the ID.

`createRequest`

Allows an external system to create a single request in HP Demand Management from the input request object. It returns an object that points to the request created.

`deleteRequests`

Allows an external system to delete one or more requests in HP Demand Management. Each of the specified request IDs is deleted, if it exists. Requests that do not exist on the instance are ignored and no error is generated. The number of requests that are deleted is returned.

`getRequests`

Allows an external system to read one or more requests in HP Demand Management. The method returns a list of request objects that match the specified request IDs. Requests that do not exist on the instance are ignored and no error is generated.

`setRequestFields`

Allows an external system to update the fields of the request in HP Demand Management, identified by the ID.

setRequestRemoteReferenceStatus

Allows an external system to update the status of the request's remote reference in HP Demand Management, identified by the ID. If the workflow is at a step that has an outbound transition that matches the new status, the workflow transition is followed. Functions include:

- Update existing remote reference
- Update request fields
- Force workflow transition using reference status

HP Demand Management Special Commands

Mercury IT Governance Center version 6.0 introduced special commands that allowed interactions using the PPM Workbench interface. This functionality has been retained in PPM Center version 7.1 and is described in the following sections.

No new special commands were introduced in version 7.1.

ksc_export_request

Description

`ksc_export_request` causes the PPM Center workflow engine to invoke the `importRequest` or `createRequest` Web service operation, based on the `axisVersion` value of the `remoteServer` setting in the `webservices.conf` file.

- `importRequest` is invoked if the value is `AXIS1`.
- `createRequest` is invoked if the value is `AXIS2`.

Manually fill in the necessary parameters in the execution step editing window.



Although the Command Builder (in the PPM Workbench) shows `ksc_export_request` in the list of commands, do *not* use the Command Builder to construct complete commands.

Syntax

```
ksc_export_request <remoteServerURL>, <remoteRequestType>,  
<exportNotes>, <exportReferences>, <exportRemoteReference>,  
<createLocalReference>
```

Parameters

Name	Type	Description
<code>remoteServerURL</code>	String	Provides the base URL of the import request server that is called by this special command. For example, <code><hostname>:<port>/itg/</code>
<code>remoteRequestType</code>	String	Specifies the name of a valid request type. This is particularly useful for integrations between PPM Center instances.
<code>exportNotes</code>	Boolean	Indicates whether you wish to include the current request's notes in the export.

Name	Type	Description
exportReferences	Boolean	Indicates whether you wish to include the current request references in the export. References are exported as URL links.
exportRemoteReference	Boolean	Indicates whether you wish to include a remote references back to the source request. If set to <code>TRUE</code> , then the remote server that is importing the request creates a remote reference back to the source request.
createLocalReference	Boolean	Indicates whether you wish to create a remote reference on the source request to the new request that was imported into the subordinate.

ksc_export_fields

Description

`ksc_export_fields` causes the PPM Center workflow engine to invoke the `updateRemoteReference` Web service operation and update the data of the request on all the remote references associated with the request.

Manually fill in the necessary parameters in the execution step editing window.



Although the Command Builder (in the Workbench) shows `ksc_export_fields` in the list of commands, do *not* use the Command Builder to construct complete commands.

Syntax

```
ksc_export_fields
<token1>="<value1>"
<token2>="<value2>"
...
<tokenN>="<valueN>"
ksc_end_of_parameters
```

Parameters

Name	Type	Description
token	String	Specifies the fully qualified token.
value	String	Specifies the new data value.

Do not export multiple selection fields or tables using `ksc_export_fields`.

Do not include square brackets around the token.

ksc_export_status_change

Description

`ksc_export_status_change` causes the PPM Center Workflow Engine to invoke the `updateRemoteReference` Web service operation and update the current status of the request on all the remote references associated with the request, as well as update the specified data.

Manually fill in the necessary parameters in the execution step editing window.



Although the Command Builder (in the Workbench) shows `ksc_export_status_change` in the list of commands, do *not* use the Command Builder to construct complete commands

Syntax

```
ksc_export_status_change <newStatus>  
<token1>="<value1>"  
<token2>="<value2>"  
...  
<tokenN>="<valueN>"  
ksc_end_of_parameters
```

Parameters

Name	Type	Description
newStatus	String	Specifies the new status.
token	String	Specifies the fully qualified token.
value	String	Specifies the new data value.

Do not export multiple selection fields or tables using `ksc_export_status_change`.

Do not include square brackets around the token.

HP Time Management Operations

HP Time Management Web service operations provide access to retrieve the time-related data.

Versions 6.0 and 7.0

Mercury IT Governance Center version 6.0 introduced a Web service operation that allowed external systems to retrieve PPM Center time data for a specified period.

This version 6.0 and 7.0 operation (`read`) has been deleted and is no longer available for your use. This has been replaced by the version 7.1 `getActualTime` Web service operation.

Version 7.1

With the release of PPM Center version 7.1, HP Time Management Web service support was altered and relies on the Axis2 framework, as described in the following section.

`getActualTime`

The `getActualTime` Web service operation provides the capability to retrieve the number of hours logged against work items. These numbers are normally used to calculate cost in a financial system.

Note that the hours returned are grouped by activities and charge codes, where appropriate.

The input parameters to `getActualTime` are as follows:

Name	Type	Description
<code>startDate</code>	Date	The (inclusive) start date for the period to get actual time information.
<code>endDate</code>	Date	The (inclusive) end date for the period to get actual time information.
Period Type Name	String	The period type name of the period to get actual time information. This should correspond to the name of a specific period type defined in the <code>KTMG_PERIOD_TYPES</code> table.

Name	Type	Description
Work Item Type	String	(Optional) Returns time logged only against this particular work item type. Valid values are: <ul style="list-style-type: none"> ■ TASK ■ REQUEST ■ PACKAGE ■ MISC
Work Item Name	String	(Optional) Returns time logged only against the named work item. Only relevant for Tasks and Misc items.
Work Item ID	String	(Optional) Returns time logged only against the specific work item ID.
Resource	Array of Strings	(Optional) Returns time logged only for the named resource. This should be the username of a PPM Center user.

Name	Type	Description
Time Sheet Statuses	Array of Strings	(Optional) Returns time only from time sheets currently having one of the included statuses. Valid values are: <ul style="list-style-type: none"> ■ UNSUBMITTED ■ PENDING_APPROVAL ■ IN_REWORK ■ APPROVED ■ CANCELLED ■ CLOSED ■ FROZEN
Time Sheet Line Statuses	Array of Strings	(Optional) Returns time only from time sheet lines currently having one of the included statuses. Valid values are: <ul style="list-style-type: none"> ■ UNSUBMITTED ■ SUBMITTED ■ APPROVED ■ REJECTED ■ CANCELLED ■ FROZEN ■ CLOSED
Include Actual Cost	Boolean	If set to <code>TRUE</code> , returns the actual cost associated with the work item. Default is <code>FALSE</code> .

HP Time Management Special Commands

Mercury IT Governance Center version 6.0 introduced special commands that allowed interactions using the PPM Workbench interface.

Versions 6.0 and 7.0

One of the version 6.0 and 7.0 special commands (`ksc_export_actual_time`) has been replaced with a new implementation—the version 6.0 implementation is no longer available for your use.

To export the time sheet data to the remote system, the `ksc_export_actual_time` special command retrieves actual time data from PPM Server and calls `updateActualTime` in the remote `TimeExportService`.

Note that the `updateActualTime` Web service operation is not implemented on the PPM Server. Therefore, if you want to take advantage of this operation, you need to implement this operation on the PPM Server.



Contact your PPM Center administrator for a copy of the `TimeExportService.wsdl` and `Time.xsd` files that complements this functionality. These files are located in their respective `wsdl` and `xsd` directories on the PPM Server at `<PPM_Install>/integration/webservices/timeservice`.

The remaining special command (`ksc_export_actual_time_as_xml`) has been deleted and is no longer available for your use.



If your organization currently relies on this deleted special command, notify HP Software Support.

Version 7.1

No new special commands were introduced in version 7.1. The only change was in the implementation of the `ksc_export_actual_time` special command.

ksc_export_actual_time

Description

`ksc_export_actual_time` allows PPM Center to update the time data for the specified period on the remote system with the actual data from the PPM Center instance.

Manually fill in the necessary parameters in the execution step editing window.



Although the Command Builder (in the Workbench) shows `ksc_export_actual_time` in the list of commands, do *not* use the Command Builder to construct complete commands.

Syntax

```
ksc_export_actual_time <remoteServerURL> <workItemType>  
<workItemID> <startDate> <endDate>
```

Parameters

Table 3-1. `ksc_export_actual_time` parameters

Name	Type	Description
remoteServerURL	String	Provides the base URL of the update request that is called by this special command. For example, <code><hostname>:<port>/itg/</code>
workItemType	String	Specifies the type of work item for which the time data is to be updated. These include the following: <ul style="list-style-type: none">■ Projects■ Tasks■ Requests■ Packages■ Misc

Table 3-1. ksc_export_actual_time parameters

Name	Type	Description
workItemID	String	<p>Specifies the work item for which the time data is to be updated.</p> <p>For requests or packages, specify the corresponding ID.</p> <p>For projects or tasks, specify the master project ID.</p> <p>For Misc, specify the lookup code.</p>
startDate	Date	<p>Specifies a start date for which the time data is to be collected.</p> <p>Use date format: YYYY:MM:DD HH:MM:SS where HH uses 24-hour notation.</p>
endDate	Date	<p>Specifies the end date for which the time data is to be collected.</p> <p>Use date format: YYYY:MM:DD HH:MM:SS where HH uses 24-hour notation.</p>

4 Developing Web Service Applications

In This Chapter:

- *Overview of the Development Process*
 - *Version 7.1 WSDL Module Elements*
 - *Web Service Definition Language Specification*
 - *XML Schema Definition Specification*
 - *Version 6.0 and 7.0 WSDL Module Elements*
 - *Web Service Definition Language Specification*
 - *Using PPM Center Special Commands*
 - *Using HP-Provided Special Commands*
 - *Creating Custom Special Commands*
-

Overview of the Development Process

The process for designing your Web service integration with PPM Center is as follows (and summarized in *Figure 4-1*).

Steps Three and Four of this development process may be streamlined using the Web Service Toolkit detailed in *Chapter 5, Introducing the Web Service Toolkit*, on page 75.

- Step One: Identify the PPM Center application module that matches your goals. See *Chapter 3, Understanding the Web Service Operations and Special Commands*, on page 37 for a description of the Web service-enabled application modules.
- Step Two: Select the available Web service operations that you want to use with PPM Center. See the appendix corresponding to your selected application module for a copy of its WSDL file, which contains all the possible operations for that module.
- Step Three: Using the appropriate Web service WSDL file, generate the classes or stubs required to create the scripts.
 - In a Java environment, run the selected WSDL file through a Java tool to create the Java stubs you can then use to construct a target object needed to invoke the Web service operation. See *Sample Java Client on page 78* for an example.
 - In a .NET environment, run the selected WSDL through a Microsoft® tool (such as Microsoft Visual Studio) to create the classes needed to call a Web services operation. See *Sample .NET Client on page 87* for an example.
- Step Four: Write your client application using the generated stubs for each of the desired Web service operations you want to implement.
- Step Five: Execute the client application to send the SOAP request and receive the response.

Figure 4-1. Constructing a Web service application

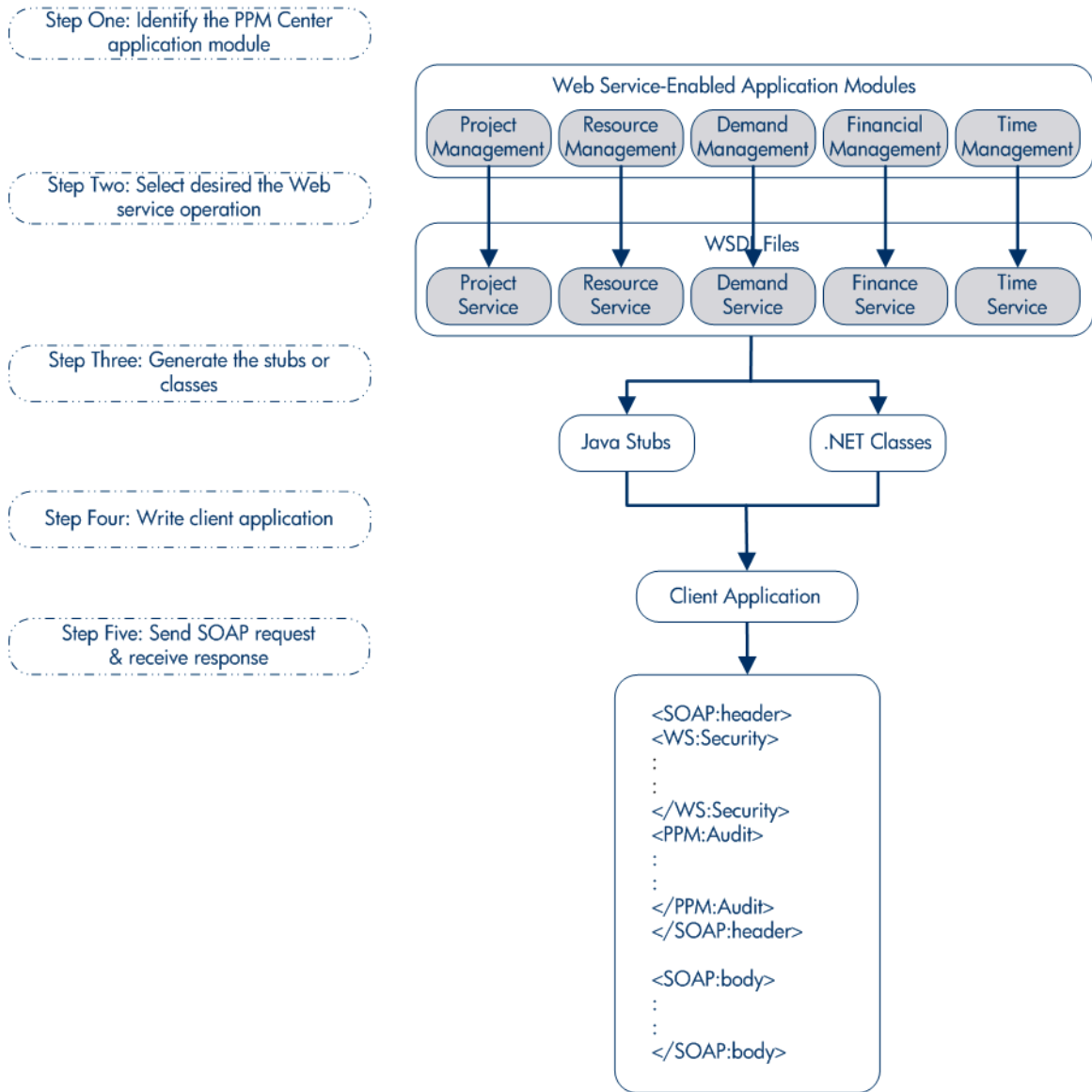
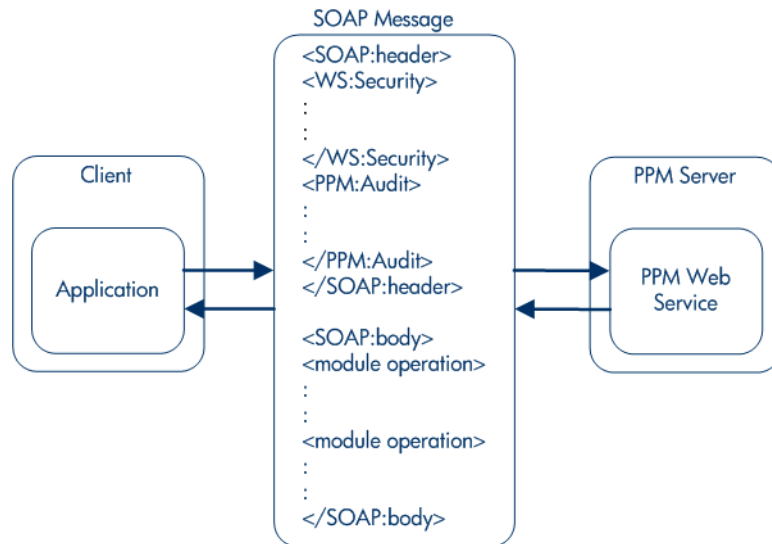


Figure 4-2 shows what happens when the Web service application is executing and an action in that application triggers the PPM Center Web service request. In this example, the application is started from a system that is not a PPM Center instance.

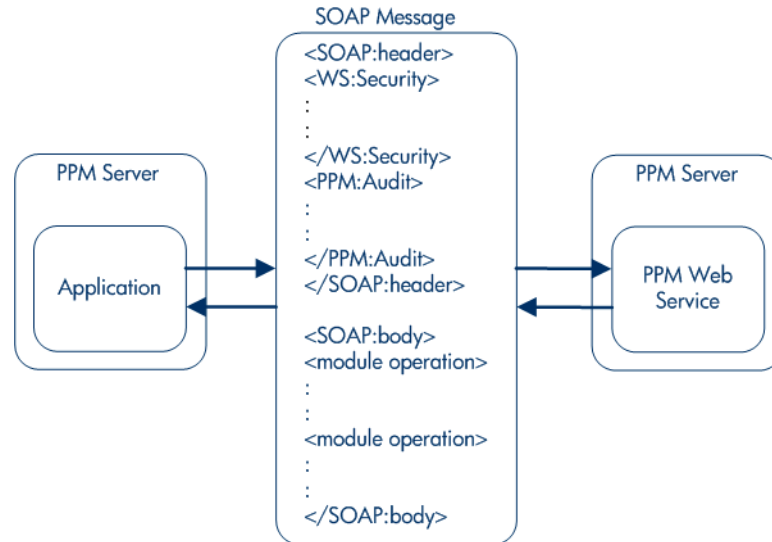
1. The application invokes a PPM Center application module operation by creating the SOAP request message.
2. The SOAP request message is sent to the PPM Center Web service through HTTP.
3. The PPM Center Web service sends the request to PPM Center.
4. PPM Center processes the request.
5. The PPM Center responds to the PPM Center Web service by returning the requested object.
6. The PPM Center Web service returns the requested objects by wrapping them in a SOAP response message.
7. The application receives the response SOAP response message and continues with its actions.

Figure 4-2. Executing using a non-PPM Center-based client



The application requesting the action can be another instance of PPM Center as shown in the following figure.

Figure 4-3. Executing using a PPM Center-based client



Version 7.1 WSDL Module Elements

Web Service Definition Language Specification

The Web Service Definition Language (WSDL) specifications used in conjunction with PPM Center Web services can be found at the following locations on your PPM Server:

- HP Project Management

`<PPM_Home>/ppmservices/ProjectService?wsdl`

- HP Demand Management

`<PPM_Home>/ppmservices/DemandService?wsdl`

- HP Resource Management

`<PPM_Home>/ppmservices/ResourceService?wsdl`

- HP Time Management

`<PPM_Home>/ppmservices/TimeService?wsdl`

- HP Financial Management

`<PPM_Home>/ppmservices/FinanceService?wsdl`

where `<PPM_Home>` represents the base URL used to access PPM Center.

Additionally, the WSDL files are stored in Axis2 archives (.aar files) located on the PPM Server in the `<PPM_Install>/server/<PPM_ServerName>/deploy/itg.war/WEB-INF/ppmservices` directory. The XSD files and schema are also included in the archive files.

XML Schema Definition Specification

XML Schema Definition (XSD) specifications that augment the WSDL can be found at the following locations on your PPM Server:

- HP Project Management

`<PPM_Home>/ppmservices/ProjectService?xsd=xsd0`

- HP Demand Management

`<PPM_Home>/ppmservices/DemandService?xsd=xsd0`

- HP Resource Management

`<PPM_Home>/ppmservices/ResourceService?xsd=xsd0`

- HP Time Management

`<PPM_Home>/ppmservices/TimeService?xsd=xsd0`

- HP Financial Management

`<PPM_Home>/ppmservices/FinanceService?xsd=xsd0`

In addition to these XSD files, the following application modules also rely on a “common” XSD file detailed in [Appendix F, *Shared Web Service Files*](#), on page 145.

- HP Project Management

`<PPM_Home>/ppmservices/ProjectService?xsd=xsd1`

- HP Demand Management

`<PPM_Home>/ppmservices/DemandService?xsd=xsd1`

- HP Resource Management

`<PPM_Home>/ppmservices/ResourceService?xsd=xsd1`

Version 6.0 and 7.0 WSDL Module Elements

Web Service Definition Language Specification

The Web Service Definition Language (WSDL) specifications used in conjunction with PPM Center Web services version 6.0 can be found at the following locations on your PPM Server:

- HP Demand Management

`<PPM_Home>/services/Request?wsdl`

- HP Financial Management

`<PPM_Home>/services/Finance?wsdl`

where `<PPM_Home>` represents the base URL used to access PPM Center.

The version 6.0 and 7.0 WSDL files do not rely on any XSD files.

Using PPM Center Special Commands

Special commands are reusable PPM Center routines that can be used in object types, request types, report types, validations, and workflow steps to perform automated functions like establishing connections to remote environments and copying files between machines.

Command instructions are interpreted by the execution engine and translated into operating system commands to be dynamically executed. Command sequences allow the automation of an entire sequence of commands that would otherwise need to be run manually one at a time. For example, command sequences could automate source code compilation, check files into version control, or run a report.

When used with Web services, special commands are most powerful when used in PPM Center workflows. In PPM Center, selected Web service operations have been incorporated in PPM Center as special commands. Additionally, user-defined Web service operations can also be executed from the PPM Center workflows.

The following sections describe some possible scenarios for using PPM Center Web services and special commands.

Using HP-Provided Special Commands

Problem

Consider the following scenario:

- The research and development (R&D) sustaining engineering team implements code to correct programming errors (defects).
- Once corrective action is completed, the defect is tested by the quality assurance (QA) team.
- After testing is complete, R&D is notified whether the test passed or failed.
- The R&D process is tracked on a PPM Center instance.
- The QA activities are monitored on a different (remote) PPM Center instance.

Solution

Use the “out-of-the-box” Web service special commands.

In this case, the HP-provided Web services are used by the two teams to request defect validation and obtain the status of the testing

Figure 4-4 shows the following Web services interactions between the two teams.

- **Software defect.** The R&D workflow step [QA handoff] creates a remote request to the QA server for testing.

The R&D workflow awaits the results of the testing [Wait for QA].

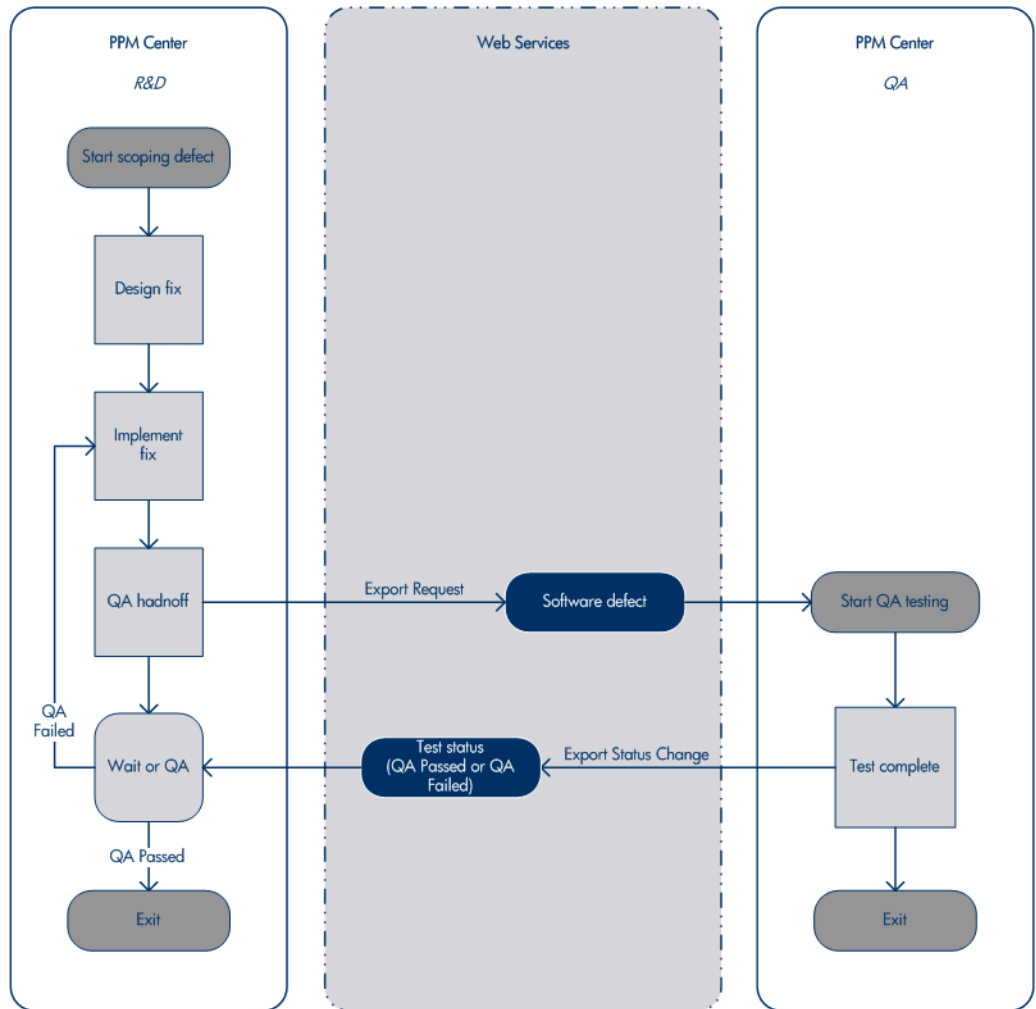
The remote QA workflow [Start QA testing] is initiated on the QA server.

- **Test status.** Once the testing is complete [Test complete], a status update (either *QA Passed* or *QA Failed*) is sent from the QA workflow to the R&D server.

The R&D workflow now takes the appropriate path based on the status of the testing.

The QA workflow is completed and exited.

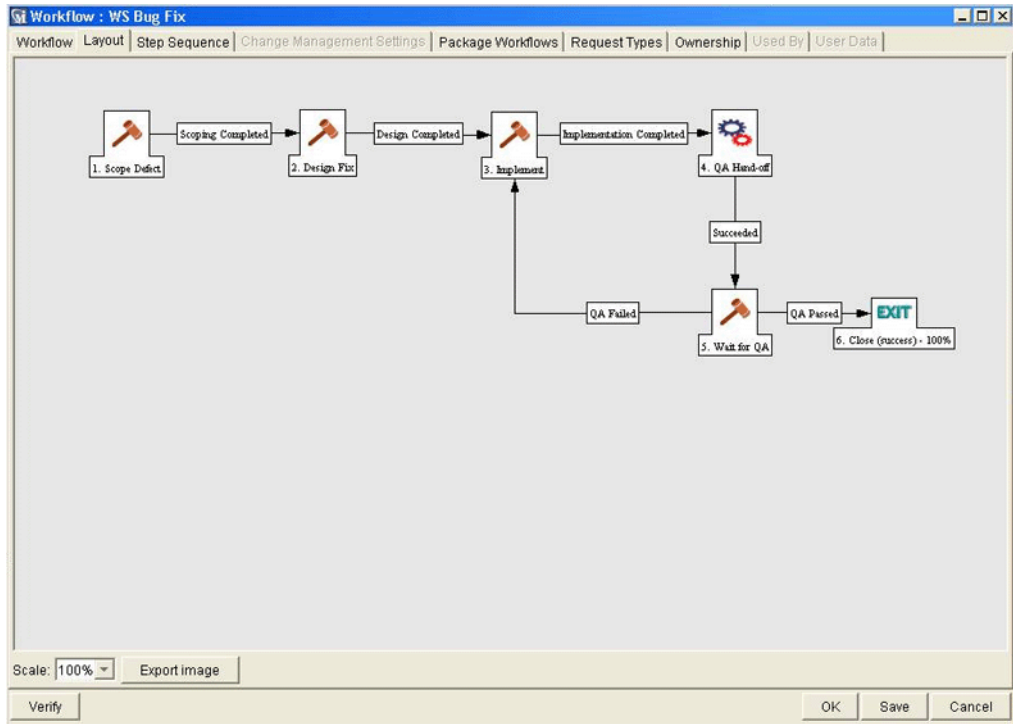
Figure 4-4. Sample scenario



The R&D Workflow

The workflow, used on the R&D PPM Center instance, is shown in the following figure.

Figure 4-5. R&D workflow



Export Request

For this scenario, the following special command is used by the R&D workflow to request testing of the corrective action.

```
ksc_export_request "http://qa/itg" "Software Defect" true true true true
```

Figure 4-6. Export request

The screenshot shows the 'Execution' dialog box with the following configuration:

- Name:** export request
- Workflow Scope:** Requests
- Description:** (empty)
- Execution Type:** Workflow Step Commands
- Workflow Event:** None
- Validation:** WF - Standard Execution Results
- Timeout:** (empty) Days
- Icon:** (empty)
- Processing Type:** Manual
- Enabled:** Yes No
- Page Response:** Finish Execution before displaying the request page to the user Display the request page immediately while execution is still running

Execution:

Command	Condition
export request	

Command Steps:

- ksc_comment "Exporting request to QA system"
- ksc_export_request "http://qa/itg" "Software Defect" true true true true

Buttons: +All, -All, New Cmd, Edit Cmd, Copy Cmd, Remove, Up, Down, Verify, OK, Save, Cancel

Ready

Export Request Status Change

Once the testing is complete, the following special command is used by the QA workflow to update the status of the R&D request.

```
ksc_export_status_change "QA Passed"  
REQD.VP.COMPLETION_DATE="[REQD.VP.COMPLETION_DATE]"  
REQD.VP.COMPLETED_BY="[REQD.VP.COMPLETED_BY]"  
ksc_end_parameters
```

Figure 4-7. Export request status change

The screenshot shows the 'Execution' dialog box with the following configuration:

- Name:** export status change
- Workflow Scope:** Requests
- Description:** (empty)
- Execution Type:** Workflow Step Commands
- Workflow Event:** None
- Validation:** WF - Standard Execution Results
- Processing Type:** Manual
- Enabled:** Yes
- Page Response:** Finish Execution before displaying the request page to the user

The 'Execution' section contains the following tables:

Command	Condition
export status change	

Command	Description
ksc_export_status_change "QA Passed" REQD.VP.COMPLETION_DATE="[REQD.VP.COMPLETION_DATE]" REQD.VP.COMPLETED_BY="[REQD.VP.COMPLETED_BY]" ksc_end_parameters	

Creating Custom Special Commands

Problem

Consider the following scenario:

- You want to create a request, insert a task in a work plan, or create a budget using a PPM Center Web service operation at specific workflow step.

Solution

Under this circumstance, use the `ksc_local_exec` special command to create a custom implementation.

This command invokes a local process on the machine running the PPM Server. It can be used to run any program that does not require interactive input. Each call using `ksc_local_exec` is an independent process and it does not execute in the context of other commands that precede it. The starting directory for the processes generated using `ksc_local_exec` is the home directory of the PPM Server. Therefore, complete path information for the executable being called is necessary if the PPM Server does not have the correct system path information.

`ksc_local_exec` Special Command

Format

The command format is as follows:

```
ksc_local_exec <command> [<parameters>]
```

The command can be batch file (on Windows®) or shell script file (on UNIX®). The parameters will be passed to the scripts as if though they were entered on the command line.

System commands do not invoke either UNIX shells or DOS shells. This means that the following `ksc_local_exec` code segment is not valid because it cannot use the pipe (|) or redirect commands (>):



```
ksc_local_exec cat names.txt | grep address > file.out
```

Return Code

The command will have an exit code which will be used as the return result of the special command. Exit code 0 is treated as success, while other values considered a failure.

Execution Output

All the output of the command execution will be written into the workflow log file for the request and can be accessed using the status section of the request.

Process Example

This example assumes the PPM Center Web Service Toolkit is used for creating a Java Web service client program. If your client program is created using different approach, simply replace the script with proper command.

1. Create a batch file or shell script to encapsulate the invocation of your Web service client program.

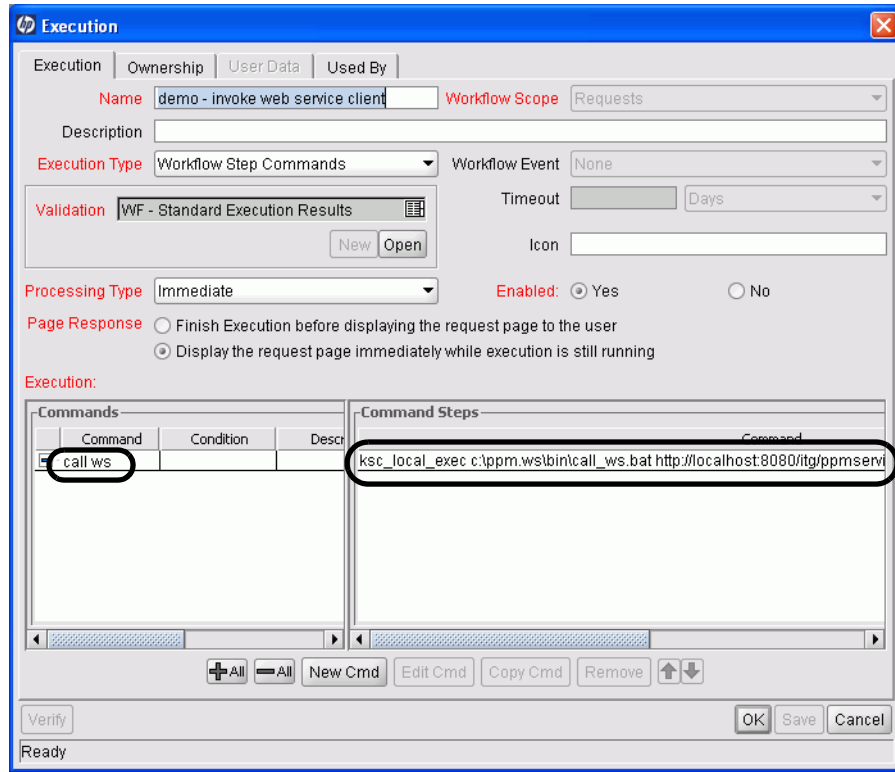
For this example, the `C:\ppm.ws\bin\call_ws.bat` file includes the following commands:

```
rem Set up environment
call C:\ppm.ws\bin\setEnv.bat

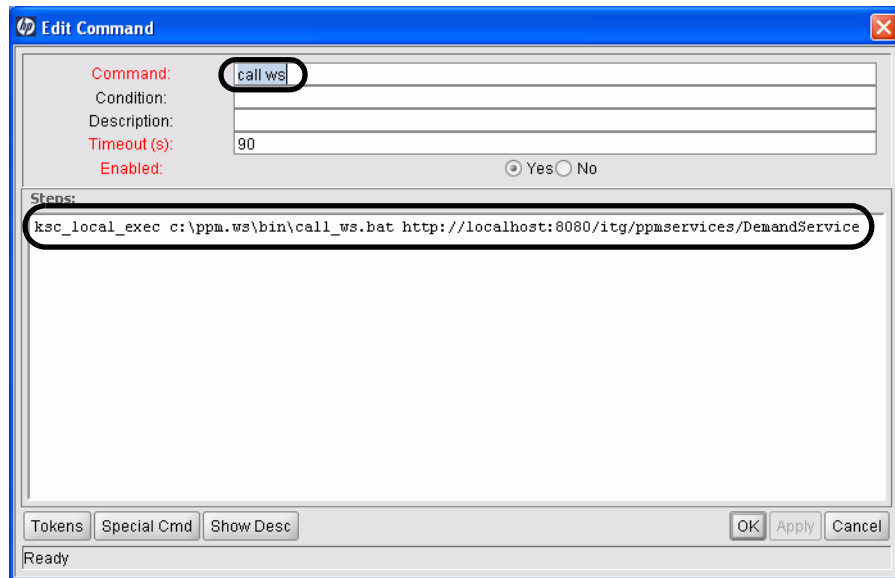
rem Invoke the Web service
java -Dclient.repository.dir=C:/ppm.ws -classpath %CPATH%
      examples.dm.DemandServiceClient %1
```

In the example above, the script will set up the correct environment and invoke the HP Demand Management Web service by reading the target Web service URL from the first command parameter.

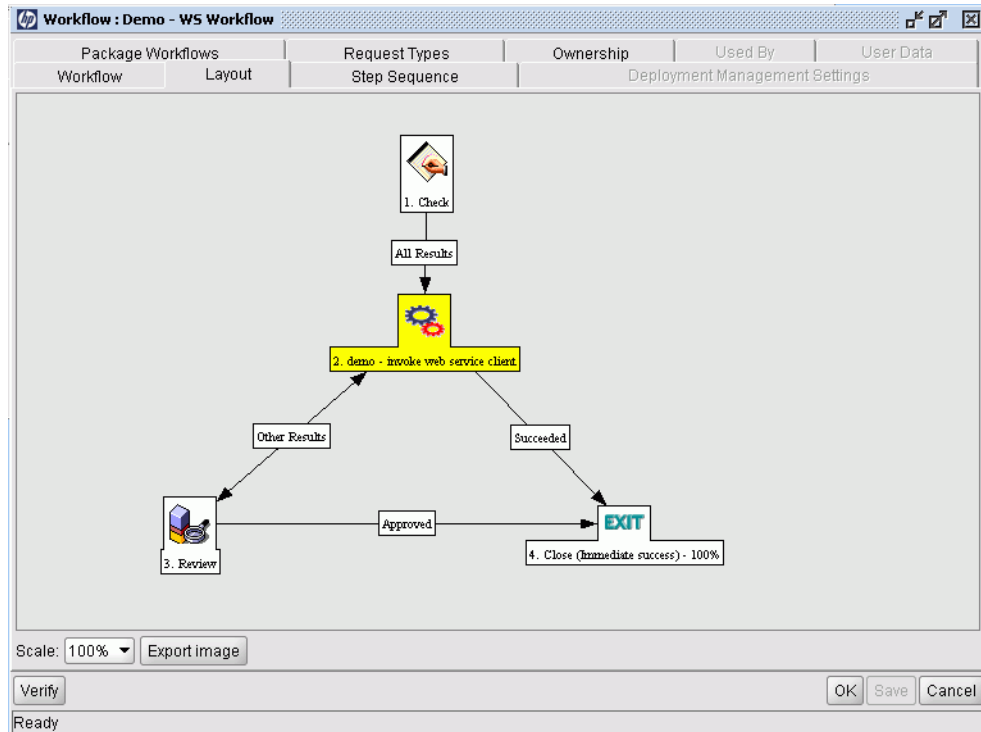
2. Create the execution step, using the `call ws` command.



In this step, the command (and its sole step) is defined as follows:



3. Add the step to the workflow.

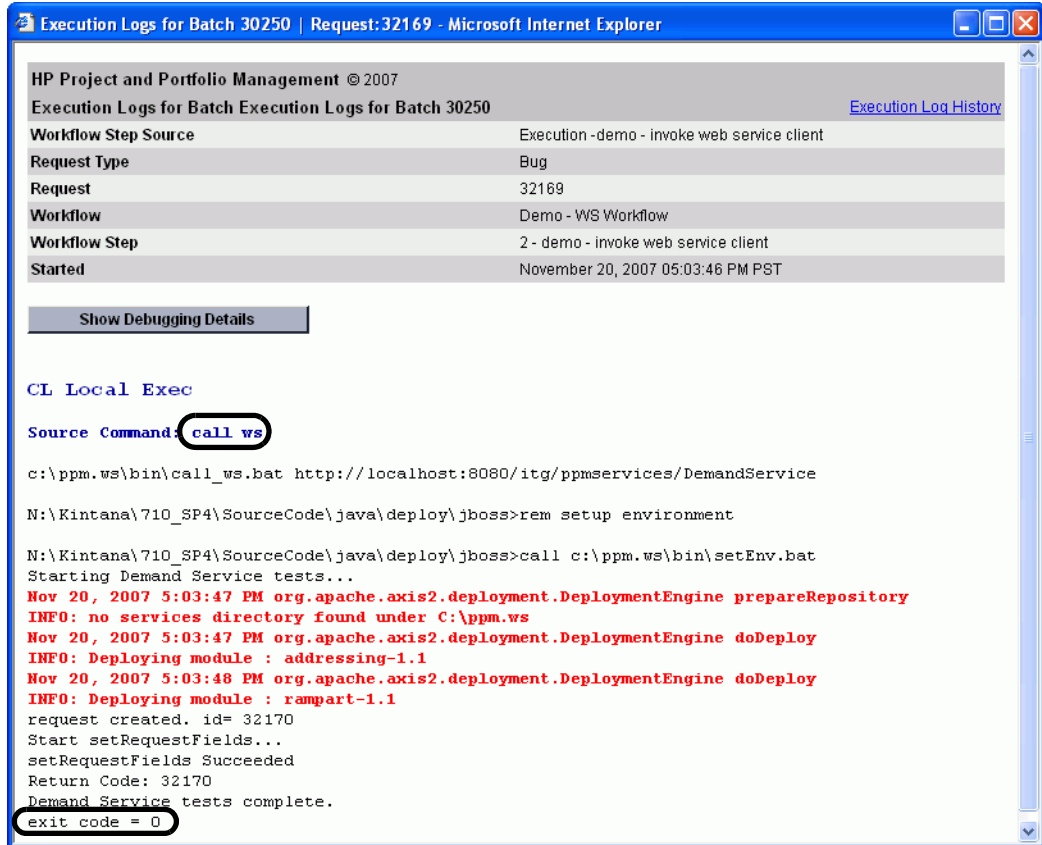


From the workflow, you can branch based on the result of the execution, allowing you to process errors or take alternative actions.

4. Apply the workflow to your request, as you normally would.

Log File Example

The following sample log file shows a successful execution of the `call ws` command, as indicated by the exit code of 0 (shown in the last line).



The screenshot shows a web browser window titled "Execution Logs for Batch 30250 | Request: 32169 - Microsoft Internet Explorer". The page content includes a header for "HP Project and Portfolio Management © 2007" and a table of execution details. Below the table is a "Show Debugging Details" button. The main log content shows a command execution: "CL Local Exec" followed by "Source Command: call ws". The log then shows the execution of a batch file "call_ws.bat" which runs a series of commands including setting up the environment and calling a web service. The log concludes with "Demand Service tests complete." and "exit code = 0".

```
HP Project and Portfolio Management © 2007
Execution Logs for Batch Execution Logs for Batch 30250
Workflow Step Source Execution - demo - invoke web service client
Request Type Bug
Request 32169
Workflow Demo - WS Workflow
Workflow Step 2 - demo - invoke web service client
Started November 20, 2007 05:03:46 PM PST

Show Debugging Details

CL Local Exec
Source Command: call ws
c:\ppm.ws\bin\call_ws.bat http://localhost:8080/itg/ppmservices/DemandService
N:\Kintana\710_SP4\SourceCode\java\deploy\jboss>rem setup environment
N:\Kintana\710_SP4\SourceCode\java\deploy\jboss>call c:\ppm.ws\bin\setEnv.bat
Starting Demand Service tests...
Nov 20, 2007 5:03:47 PM org.apache.axis2.deployment.DeploymentEngine prepareRepository
INFO: no services directory found under C:\ppm.ws
Nov 20, 2007 5:03:47 PM org.apache.axis2.deployment.DeploymentEngine doDeploy
INFO: Deploying module : addressing-1.1
Nov 20, 2007 5:03:48 PM org.apache.axis2.deployment.DeploymentEngine doDeploy
INFO: Deploying module : rampart-1.1
request created. id= 32170
Start setRequestFields...
setRequestFields Succeeded
Return Code: 32170
Demand Service tests complete.
exit code = 0
```

5 Introducing the Web Service Toolkit

In This Chapter:

- *Overview of the Web Service Toolkit*
 - *Toolkit Contents*
 - *Generating New Stubs*
 - *Sample Java Client*
 - *Phase One: Set Up the Environment*
 - *Phase Two: Write the Client Code*
 - *Phase Three: Set Up the Client Security*
 - *Phase Four: Compile the Client Code*
 - *Phase Five: Execute the Client Code*
 - *Sample .NET Client*
 - *Sample .NET Process*
-

Overview of the Web Service Toolkit

For your convenience, HP provides the Web Service Toolkit to decrease Web service application development time.

Starting with Service Pack 5, this software is available from your PPM Center instance. You may need to contact your PPM Center administrator to obtain a copy of the Toolkit, which is located on the PPM Server at:

```
<PPM_Home>/utilities/webservice/webservice-toolkit.zip
```

where *<PPM_Home>* represents the base URL used to access PPM Center.

The files are archived in Zip format and must be extracted before they can be used. The files can be copied to your local PC or workstation—they do not have to be installed on a PPM Server.

There are many ways to create client applications. The Toolkit provides copies of the WSDL and XSD files, along with several libraries, so that you can concentrate on your applications' logic rather than worry about the framework.

Toolkit Contents

Although Web services can be implemented using numerous languages, the Toolkit currently provides only Java and .NET examples. These are included in the Toolkit in the following directories:

- java
- MicrosoftDotNet

java Directory Content

The Java-specific directory in the Toolkit includes the following directories and files:

- `bin` contains the scripts to set up the compiling and execution environment.
- `client`
 - `src`
 - `examples` contains subdirectories with code samples which can be used as starting point for developing custom client program logic.
 - `dm` contains an example for HP Demand Management.
 - `fm` contains an example for HP Financial Management.
 - `pm` contains an example for HP Project Management.

- `rm` contains an example for HP Resource Management.
 - `tm` contains an example for HP Time Management.
 - `security` contains an example of the callback handler.
 - `fm60` contains an example for HP Financial Management using the version 6.0 and 7.0 Web service implementation.
- `conf` contains example configuration files which are required for authentication—these typically require modification.
- `docs`
 - `javadoc_7.1.zip` contains all the PPM Center version 7.1 javadocs.
 - `javadoc_6.0.zip` contains the javadocs for the Mercury IT Governance Center version 6.0 and PPM Center version 7.0 implementations.
- `lib` subdirectories include all the required libraries for compiling and executing the client code, including the `webservice_client.jar`, which is the stubs generated from WSDL.
- `modules` contains the rampart and addressing libraries.

MicrosoftDotNet Directory Content

The .NET-specific directory of the Toolkit includes the following directories and files:

- `DemandServiceTest` contains an example for HP Demand Management
 - `Properties` contains Visual Studio and .NET configuration files.
 - `Web References`
 - `localhostDemandService` contains WSDL, XSD, and data source files.
- `ProjectServiceTest` contains an example for HP Project Management
 - `Properties` contains Visual Studio and .NET configuration files.
 - `Web References`
 - `localhostDemandService` contains WSDL, XSD, and data source files.

Generating New Stubs

The Toolkit contains the Web service stubs. However, there are situations where you may want to generate new (or additional) stubs from the WSDL. For example, you may want to:

- Change the package name of the stubs
- Understand the stub sources to make it easier to program the client program
- Make changes to stubs to meet the special needs for advanced customers
- The `bin` directory of the Toolkit includes example scripts for generating stubs.
 - For Windows, use `wSDL2java.bat <ServiceName>`
 - For UNIX, use `wSDL2java.sh <ServiceName>`

where `<ServiceName>` represents the name of the application module Web service. For example, for HP Time Management, the `<ServiceName>` would be `TimeService`.

Sample Java Client

The following sections detail how to create a Java client for PPM Center Web service for an HP Demand Management operation. The Demand Service is defined in the WSDL file `DemandService.wsdl`.

The examples provided in this Toolkit rely on the Java SDK, version 1.4.2 (or later).

Using this scenario, development of a Java client requires the following phases:

- *Phase One: Set Up the Environment*
- *Phase Two: Write the Client Code*
- *Phase Three: Set Up the Client Security*
- *Phase Four: Compile the Client Code*
- *Phase Five: Execute the Client Code*

Phase One: Set Up the Environment

Set the Environment Variables and Java CLASSPATH

In order to compile and execute the client code, the following libraries must be in the Java CLASSPATH (in addition to any libraries needed by your customization):

- **Axis2 libraries.** These are located in `<Java_DevKit_Install>/lib/axis2.`

where `<Java_DevKit_Install>` represents the location where the Java files from the Toolkit files were extracted.

- **Axis1.1 libraries.** These are located in `<Java_DevKit_Install>/lib/axis1.1.`
- **Rampart libraries.** These are located in `<Java_DevKit_Install>/lib/rampart.`
- **PPM Center libraries.** These are located in `<Java_DevKit_Install>/lib/ppm.`

- **Web service stubs.** If you use the stubs that are shipped with the Toolkit, they are found in `<Java_DevKit_Install>/lib/ppm.`

If you use the stubs you generated yourself, the Toolkit searches for them in `<Java_DevKit_Install>/client/classes.`

The Toolkit includes scripts to help establish the proper Java CLASSPATH and any Toolkit-related environment variables.

- For Windows, use `<Java_DevKit_Install>\bin\setEnv.bat`
- For UNIX, use `<Java_DevKit_Install>/bin/setEnv.sh`

Phase Two: Write the Client Code

The sample detailed in this section is for an HP Demand Management Web service. The full content of the example code can be found in the Toolkit at

`<Java_DevKit_Install>/client/src/examples/dm/
DemandServiceClient.java.`

Line numbers in the following sections are provided for clarity and identification purposes.

Establish the Context

In the Toolkit, the Axis2 framework is used to simplify the client code. The first thing the client program needs to do is to establish the context in its constructor:

```
32     protected ConfigurationContext ctx = null;
33
34     public DemandServiceClient() {
35         String repositoryPath =
36             System.getProperty("client.repository.dir");
37         String axis2 = repositoryPath + "/conf/client-axis2.xml";
38         File file = new File(axis2);
39         if (file.exists()) {
40             try {
41                 ctx = ConfigurationContextFactory
42                     .createConfigurationContextFromFileSystem(
43                         repositoryPath, axis2);
44             } catch (Exception e) {
45                 e.printStackTrace();
46             }
47         }
48     }
```

- Line 35 reads in the system property, `client.repository.dir`. It uses this information to locate the Axis2 client configuration file. It is also used by Axis2 to locate other required resource such as modules. Make sure this line of code is pointing to the root directory of the Toolkit, which contains subdirectories for each of the application modules.

This property is required when the client program executes.

- Lines 39 to 45 create the context using the setting of repository directory and the name of the Axis2 client configuration file.

Create a Request Object

The following portion of the sample creates a request using a Web service.

- You need to create a similar request object for your application module operation.
- The logic can be customized to construct the request object using your own logic, such as reading from another system.

```
84     private String createRequest(String serviceURL) throws
      Exception {
85
86         // Construct a request object
87         Request oRequest = Request.Factory.newInstance();
88         oRequest.setRequestType("Bug");
89         SimpleField[] fields = new SimpleField[6];
90
91         // Set values for the fields of the request object
92
93         // Set field 'Description'
94         SimpleField field_A = SimpleField.Factory.newInstance();
95         field_A.setToken("REQ.DESCRPTION");
96         field_A.setStringValue1Array(new String[] { "WebService
          Test" });
97         fields[0] = field_A;
98
99         // Set field 'Department'
100        SimpleField field_B = SimpleField.Factory.newInstance();
101        field_B.setToken("REQ.DEPARTMENT_NAME");
102        field_B.setStringValue1Array(new String[] { "Finance" });
103        fields[1] = field_B;
104
105        // Set field 'Module'
106        SimpleField field_C = SimpleField.Factory.newInstance();
107        field_C.setToken("REQD.VP.MODULE");
108        field_C.setStringValue1Array(new String[] { "Module A"
          });
109        fields[2] = field_C;
110
111        // Set field 'Platform'
112        SimpleField field_D = SimpleField.Factory.newInstance();
113        field_D.setToken("REQD.VP.PLATFORM");
114        field_D.setStringValue1Array(new String[] { "Unix" });
115        fields[3] = field_D;
116
117        // Set field 'Impact'
118        SimpleField field_E = SimpleField.Factory.newInstance();
119        field_E.setToken("REQD.VP.IMPACT");
120        field_E.setStringValue1Array(new String[] { "Warning" });
121        fields[4] = field_E;
122
123        // Set field 'Reproducible'
124        SimpleField field_F = SimpleField.Factory.newInstance();
125        field_F.setToken("REQD.VP.REPRO");
126        field_F.setStringValue1Array(new String[] { "Y" });
127        fields[5] = field_F;
128
129        // Add all the fields to request object
```

```

130 oRequest.setSimpleFieldsArray(fields);
131
132 // Set Notes
133 Note[] notes = new Note[1];
134 Note note = Note.Factory.newInstance();
135 note.setAuthor("admin");
136 note.setContent("WebService Test Note");
137 Calendar calendar = Calendar.getInstance();
138 note.setCreationDate(calendar);
139 notes[0] = note;
140 oRequest.setNotesArray(notes);

```

- Line 87 instantiates a new request object.
- Line 88 sets the request type to `Bug`.

The request type is a mandatory field for each request object.

- Line 89 creates an empty field list, with a size of six for the six to-be-populated fields.
- Lines 91 to 130 set the values for the request fields.

Each field has a token as its identifier and a single-element string array as the value.

- Lines 132 to line 140 set the values for the request notes.

Invoke the Service

After the request object constructed, the Web service can be invoked.

```

142 // Get Web services
143 DemandServiceStub stub = new DemandServiceStub(ctx,
serviceURL);
144 setHttpBasicAuthHeader(stub, serviceURL);
145 ServiceClient sc = stub._getServiceClient();
146
147 // Construct message to send
148 CreateRequestDocument inDoc =
CreateRequestDocument.Factory
149 .newInstance();
150 CreateRequestDocument.CreateRequest createRequest = inDoc
151 .addNewCreateRequest();
152 createRequest.setRequestObj(oRequest);
153
154 // Invoke Web services
155 CreateRequestResponseDocument outDoc =
stub.createRequest(inDoc);

```

- Line 142 creates the service stub. It uses the context and URL of the Web service.

In this example, the Web service URL is obtained from the command line. Customize this code, as desired, to read the URL from an alternate configuration file.

- Line 144 sets the request header for HTTP basic authentication.

This coding is only required when the server demands HTTP basic authentication. See [Chapter 2, *Configuring Web Services on the PPM Server*, on page 23](#) for additional configuration information.

When HTTP basic authentication is not required on the server, the client program can still call the function `setHttpBasicAuthHeader()` without any affect—it is simply ignored by the server. However, if HTTP basic authentication is required and the client program does not call the function, the Web service call will fail with error message similar to the following:

```
org.apache.axis2.AxisFault: com.mercury.itg.ws.core.utils.WSRuntimeException: HTTP basic authentication failed: HTTP basic authentication is configured as enforced for web service, but authorization header is not received.
at org.apache.axis2.description.OutInAxisOperationClient.send(OutInAxisOperation.java:271)
at org.apache.axis2.description.OutInAxisOperationClient.execute(OutInAxisOperation.java:202)
at com.test.DemandServiceStub.createRequest(DemandServiceStub.java:553)
at com.mercury.itg.ws.dm.DemandServiceClient.createRequest(DemandServiceClient.java:155)
at com.mrcury.itg.ws.dm.DemandServiceClient.main(DemandServiceClient.java:70)
```

- Line 145 gets a reference to the Web service from the service stub.
- Lines 147 to line 152 construct the request message using the request object constructed in [Create a Request Object on page 81](#).
- Line 155 invokes the Web service.

Process the Return Message

Based on the Web service definition, each operation has different return type.

```
157 // Process return message
158 RemoteReference ref =
    outDoc.getCreateRequestResponse().getReturn();
159 return ref.getIdentifier().getId();
```

- Line 158 gets the return message.
- Line 159 extracts the request ID from the return message.

Set the HTTP Basic Authentication Header

In this sample, the user ID and password are hard-coded.

Customize this code, as desired, to read the user name and password from an alternate source, such as a database or configuration file. Alternatively, replace the user ID and password strings in lines 255 and 256 with the user ID and password of a PPM Center user who has rights to create requests.

```
247  /**
248  * Set http basic auth headers
249  * @param stub
250  */
251  public void setHttpBasicAuthHeader(Stub stub, String
serviceURL) {
252  Options options = new Options();
253  HttpTransportProperties.Authenticator auth = new
HttpTransportProperties.Authenticator();
254  auth.setUsername("admin");
255  auth.setPassword("admin");
256  auth.setPreemptiveAuthentication(true);
257  options.setProperty(HTTPConstants.AUTHENTICATE, auth);
258  options.setTo(new EndpointReference(serviceURL));
259  stub._getServiceClient().setOptions(options);
260
261  }
```

Phase Three: Set Up the Client Security

Configure Client Security

PPM Center Web services supports two types of authentication:

- HTTP basic authentication
- WS-Security

If WS-Security is desired, you need to configure the client (see [Chapter 2, *Configuring Web Services on the PPM Server*, on page 23](#) for additional details about configuring WS-Security).

The Toolkit includes a sample client configuration file, `<Java_DevKit_Install>/conf/client-axis2.xml`. If you change the name of the configuration file, update the client program accordingly when creating the context (as detailed in the section [Establish the Context on page 80](#)).

The configuration file contains the user name (admin in this excerpt) and for password, it provides a password callback class (examples.security.PasswordCallbackHandler) which implements the Java CallbackHandler. This class supplies the password and you should program your own password callback class as detailed in the [Configure the Password Callback Handler](#) section.

```
<parameter name="OutflowSecurity">
  <action>
    <items>UsernameToken Timestamp Encrypt</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>
```

You also have the option to encrypt the password so that when the SOAP message travels over a network, the credentials are encrypted. Configuration for this option would be similar to the following:

```
<module ref="rampart" />
<parameter name="OutflowSecurity">
  <action>
    <items>UsernameToken Timestamp Encrypt</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>
```

The `<encryptionPropFile>` tag specifies the name of the encryption property file. The file must be in the Java CLASSPATH when executing the client program.

The Toolkit includes a sample `<Java_DevKit_Install>/conf/client.properties` file. Note that this path should be included in the Java CLASSPATH.

The Toolkit also includes a sample keystore file, `<Java_DevKit_Install>/conf/client.jks`. The name of the keystore file is specified in the `client.properties` property file. This path should also be included in the Java CLASSPATH.

In the event the password that has been provided is incorrect, an error message similar to the following will be displayed:

```
org.apache.axis2.AxisFault: Authentication failed for user admin.
at org.apache.axis2.description.OutInAxisOperationClient.send(OutInAxisOperation.java:271)
at org.apache.axis2.description.OutInAxisOperationClient.execute(OutInAxisOperation.java:202)
at com.test.DemandServiceStub.createRequest(DemandServiceStub.java:553)
at com.HP.itg.ws.dm.DemandServiceClient.createRequest(DemandServiceClient.java:150)
at com.HP.itg.ws.dm.DemandServiceClient.main(DemandServiceClient.java:65)
```

Additionally, when WS-Security is used, the server and client need to have the same configuration setting for `Timestamp` (indicating whether or not to include timestamp information). An error message similar to the following will be displayed if the `Timestamp` settings differ:

```
Exception in thread "main" org.apache.axis2.AxisFault: WSDoAllReceiver: security processing failed
(actions mismatch)
    at org.apache.axis2.description.OutInAxisOperationClient.send(OutInAxisOperation.java:271)
    at
org.apache.axis2.description.OutInAxisOperationClient.execute(OutInAxisOperation.java:202)
    at com.test.DemandServiceStub.createRequest(DemandServiceStub.java:553)
    at com.HP.itg.ws.dm.DemandServiceClient.createRequest(DemandServiceClient.java:150)
    at com.HP.itg.ws.dm.DemandServiceClient.main(DemandServiceClient.java:65)
```

Configure the Password Callback Handler

As previously described, the `client-axis2.xml` configuration file specifies the class that is called by Axis2 to get the password.

The Toolkit includes a very simple password callback sample. The source is found in `<Java_DevKit_Install>/client/src/examples/security/PasswordCallbackHandler.java`.

Normally you need to change this class to include the actual logic for obtaining the password, such as reading from database or LDAP.

Phase Four: Compile the Client Code

The Toolkit includes sample scripts to compile the client code:

- For Windows, use `<Java_DevKit_Install>\bin\compile_client.bat`
- For UNIX, use `<Java_DevKit_Install>/bin/compile_client.sh`

Phase Five: Execute the Client Code

The same Java CLASSPATH used to compile the client code is needed to execute the program.

In addition, the `<Java_DevKit_Install>/conf` directory must be included in the Java CLASSPATH if encryption is used. This is required to enable Axis2 to find the encryption properties file. If you set up the environment as detailed in *Set the Environment Variables and Java CLASSPATH on page 79*, this has already been included.

For Windows, use:

```
run_client.bat examples.dm.DemandServiceClient http://<PPM_Home>/ppmservices/DemandService
```

For UNIX, use:

```
run_client.sh examples.dm.DemandServiceClient http://<PPM_Home>/ppmservices/DemandService
```

Sample .NET Client

The following section details how to create a .NET client for PPM Center Web service for an HP Demand Management operation. The Demand Service is defined in the WSDL file `DemandService.wsdl`.

The examples provided in this Toolkit rely on the following:

- Visual Studio 2005 with C# installed
- .NET framework 2.0
- Microsoft Web Services Enhancements (WSE) 3.0 installed in “Visual Studio Developer” mode



Note that, although similar, this sample does not match the content in the Toolkit.

Sample .NET Process

1. Open Visual Studio 2005.
2. Create a new Console Application using C#. Name the new console application `PPMWebServiceSample`.
3. Right-click on the project in Solution Explorer and select **WSE Settings 3.0**.
4. Enable the **Enable this project for Web Services Enhancements** option and click **OK**.
5. Add a Web reference to your project.

Name the new Web reference `PPMDemandService` and specify the following URL as the reference:

```
<PPM_Home>/ppmservices/DemandService?wsdl
```

6. Add the following statements to the beginning of the main file (`Program.cs`):

```
using PPMWebServiceSample.PPMDemandService;  
using Microsoft.Web.Services3;  
using Microsoft.Web.Services3.Design;  
using Microsoft.Web.Services3.Security;  
using Microsoft.Web.Services3.Security.Tokens;
```

7. Insert the following code into the `Main` function of your main (`Program`) class:

```
// create service proxy  
DemandServiceWse serviceProxy = new DemandServiceWse();  
UsernameOverTransportAssertion policyAssertion = new  
UsernameOverTransportAssertion();  
  
// setup WS-Security credentials  
policyAssertion.UsernameTokenProvider = new  
UsernameTokenProvider("username", "password");  
Policy policy = new Policy(policyAssertion);  
serviceProxy.SetPolicy(policy);
```

8. Replace the user ID and password strings in the `UsernameTokenProvider` statement with the user ID and password of a PPM Center user who has rights to create requests

The sample code listed here creates a Web service proxy to access the HP Demand Management Web service, sets the WSE authentication headers, and specifies the PPM Center user account under which to perform the operations on the PPM Center instance.

9. Continue to edit the `Main` function and append the following code after the code you added in [step 7](#).


```

//Instantiate Request object
Request request = new Request();
request.requestType = "Bug";

//Set Required Fields
SimpleField[] fields = new SimpleField[6];
//Description
fields[0] = new SimpleField();
fields[0].token = "REQ.DESCRPTION";
fields[0].stringValue = new String[] { "WebService Sample"
};

//Department
fields[1] = new SimpleField();
fields[1].token = "REQ.DEPARTMENT_NAME";
fields[1].stringValue = new String[] { "Finance" };

//Module
fields[2] = new SimpleField();
fields[2].token = "REQ.VP.MODULE";
fields[2].stringValue = new String[] { "Module A" };

//Platform
fields[3] = new SimpleField();
fields[3].token = "REQ.VP.PLATFORM";
fields[3].stringValue = new String[] { "Unix" };

//Impact
fields[4] = new SimpleField();
fields[4].token = "REQ.VP.IMPACT";
fields[4].stringValue = new String[] { "Warning" };

//Reproducible
fields[5] = new SimpleField();
fields[5].token = "REQ.VP.REPRO";
fields[5].stringValue = new String[] { "Y" };

//Add fields to request
request.simpleFields = fields;

//Set Notes
Note[] notes = new Note[1];
notes[0] = new Note();
notes[0].author = "username";
notes[0].content = "WebService Sample Note";
notes[0].creationDate = DateTime.Now;

//Add notes to request
request.notes = notes;

//Call Webservice
createRequest cr = new createRequest();
cr.requestObj = request;
createRequestResponse response =
serviceProxy.createRequest(cr);

Console.WriteLine("Request: " +
response.@return.identifier.id + " Status: " +
response.@return.status);

```

10. In the `notes[0].author = "username";` statement, replace the username string with the user ID you specified for the authentication in [step 7](#).

This code calls the `createRequest` operation to create a PPM Center request using the sample data (included in the code).

11. Run the project.

The program connects to the HP Demand Management Web service, creates a request, then displays the request ID.

A HP Project Management Web Service Files

Version 7.1 WSDL File

```
<wsdl:definitions xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:ns="http://www.mercury.com/ppm/pm/ws" xmlns:pm="http://www.mercury.com/ppm/pm/types" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:axis2="http://pm.ppm.mercury.com" targetNamespace="http://pm.ppm.mercury.com">
  <wsdl:types>

    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" attributeFormDefault="qualified" elementFormDefault="qualified" xmlns:ns="http://www.mercury.com/ppm/pm/ws" targetNamespace="http://www.mercury.com/ppm/pm/ws">
      <xs:import namespace="http://www.mercury.com/ppm/pm/types" schemaLocation="../xsd/projectManagement.xsd"/>

      <xs:element name="createProject">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="projectBean" nillable="false" maxOccurs="unbounded" type="pm:projectCreationInput"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="createProjectResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="return" nillable="true" maxOccurs="unbounded" type="pm:createProjectResultType"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="updateProject">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="projectUpdateInput" nillable="false" maxOccurs="unbounded" type="pm:projectUpdateType"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="updateProjectStatus">
        <xs:complexType>
          <xs:sequence>
            <xs:element maxOccurs="1" name="projectInput" nillable="false" type="pm:projectInputType"/>
            <xs:element name="newStatus" type="xs:string" minOccurs="1" maxOccurs="1"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="createBlankWorkPlan">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="workPlanInput" nillable="false" maxOccurs="unbounded" type="pm:workPlanInputType"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="createWorkPlanFromTemplate">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="workPlanInput" nillable="false" maxOccurs="1" type="pm:workPlanInputType"/>
            <xs:element name="templateName" nillable="false" maxOccurs="1" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:schema>
  </wsdl:types>

```

```

        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="addTaskToExistingWorkPlan">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="workPlanInput" nillable="false" type="pm:workPlanInputType"/>
            <xs:element name="tasks" nillable="false" maxOccurs="unbounded" type="pm:taskType"/>
            <xs:element name="topAnchor" nillable="false" type="pm:anchorType"/>
            <xs:element name="bottomAnchor" nillable="true" minOccurs="0" type="pm:anchorType"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="addTaskToExistingWorkPlanResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="return" nillable="true" maxOccurs="unbounded"
type="pm:addTaskResultType"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="importWorkPlanTasks">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="workPlanInput" nillable="false" type="pm:workPlanInputType"/>
            <xs:element name="tasks" nillable="false" maxOccurs="unbounded" type="pm:taskType"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="importWorkPlanTasksResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="return" nillable="true" maxOccurs="unbounded"
type="pm:addTaskResultType"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="searchTasks">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="searchPreferences" nillable="false"
type="pm:searchTaskPreferenceDetail"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="searchTasksResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="taskId" nillable="false" maxOccurs="unbounded" type="xs:long"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="readTasks">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="taskId" nillable="false" maxOccurs="unbounded" type="xs:long"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="readTasksResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="return" nillable="false" maxOccurs="unbounded" type="pm:taskType"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="updateTaskActuals">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="actualDetail" nillable="false" maxOccurs="unbounded"
type="pm:updateActualsInput"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:schema>

```

```

</wsdl:types>
<wsdl:message name="createProjectMessage">
  <wsdl:part name="input" element="ns:createProject"/>
</wsdl:message>
<wsdl:message name="createProjectResponseMessage">
  <wsdl:part name="output" element="ns:createProjectResponse"/>
</wsdl:message>
<wsdl:message name="updateProjectMessage">
  <wsdl:part name="input" element="ns:updateProject"/>
</wsdl:message>
<wsdl:message name="updateProjectStatusMessage">
  <wsdl:part name="input" element="ns:updateProjectStatus"/>
</wsdl:message>
<wsdl:message name="createBlankWorkPlanMessage">
  <wsdl:part name="input" element="ns:createBlankWorkPlan"/>
</wsdl:message>
<wsdl:message name="createWorkPlanFromTemplateMessage">
  <wsdl:part name="input" element="ns:createWorkPlanFromTemplate"/>
</wsdl:message>
<wsdl:message name="addTaskToExistingWorkPlanMessage">
  <wsdl:part name="input" element="ns:addTaskToExistingWorkPlan"/>
</wsdl:message>
<wsdl:message name="addTaskToExistingWorkPlanResponseMessage">
  <wsdl:part name="output" element="ns:addTaskToExistingWorkPlanResponse"/>
</wsdl:message>
<wsdl:message name="importWorkPlanTasksMessage">
  <wsdl:part name="input" element="ns:importWorkPlanTasks"/>
</wsdl:message>
<wsdl:message name="importWorkPlanTasksResponseMessage">
  <wsdl:part name="output" element="ns:importWorkPlanTasksResponse"/>
</wsdl:message>
<wsdl:message name="searchTasksMessage">
  <wsdl:part name="input" element="ns:searchTasks"/>
</wsdl:message>
<wsdl:message name="searchTasksResponseMessage">
  <wsdl:part name="output" element="ns:searchTasksResponse"/>
</wsdl:message>
<wsdl:message name="readTasksMessage">
  <wsdl:part name="input" element="ns:readTasks"/>
</wsdl:message>
<wsdl:message name="readTasksResponseMessage">
  <wsdl:part name="output" element="ns:readTasksResponse"/>
</wsdl:message>
<wsdl:message name="updateTaskActualsMessage">
  <wsdl:part name="input" element="ns:updateTaskActuals"/>
</wsdl:message>
<wsdl:portType name="ProjectServicePortType">
  <wsdl:operation name="createProject">
    <wsdl:input message="axis2:createProjectMessage"/>
    <wsdl:output message="axis2:createProjectResponseMessage"/>
  </wsdl:operation>
  <wsdl:operation name="updateProject">
    <wsdl:input message="axis2:updateProjectMessage"/>
  </wsdl:operation>
  <wsdl:operation name="updateProjectStatus">
    <wsdl:input message="axis2:updateProjectStatusMessage"/>
  </wsdl:operation>
  <wsdl:operation name="createBlankWorkPlan">
    <wsdl:input message="axis2:createBlankWorkPlanMessage"/>
  </wsdl:operation>
  <wsdl:operation name="createWorkPlanFromTemplate">
    <wsdl:input message="axis2:createWorkPlanFromTemplateMessage"/>
  </wsdl:operation>
  <wsdl:operation name="addTaskToExistingWorkPlan">
    <wsdl:input message="axis2:addTaskToExistingWorkPlanMessage"/>
    <wsdl:output message="axis2:addTaskToExistingWorkPlanResponseMessage"/>
  </wsdl:operation>
  <wsdl:operation name="importWorkPlanTasks">
    <wsdl:input message="axis2:importWorkPlanTasksMessage"/>
    <wsdl:output message="axis2:importWorkPlanTasksResponseMessage"/>
  </wsdl:operation>
  <wsdl:operation name="searchTasks">
    <wsdl:input message="axis2:searchTasksMessage"/>
    <wsdl:output message="axis2:searchTasksResponseMessage"/>
  </wsdl:operation>
  <wsdl:operation name="readTasks">

```

```

        <wsdl:input message="axis2:readTasksMessage"/>
        <wsdl:output message="axis2:readTasksResponseMessage"/>
    </wsdl:operation>
    <wsdl:operation name="updateTaskActuals">
        <wsdl:input message="axis2:updateTaskActualsMessage"/>
    </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="ProjectServiceSOAP11Binding" type="axis2:ProjectServicePortType">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="createProject">
        <soap:operation soapAction="axis2:createProject" style="document"/>
        <wsdl:input>
            <soap:body use="literal" namespace="http://pm.ppm.mercury.com"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" namespace="http://pm.ppm.mercury.com"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="updateProject">
        <soap:operation soapAction="axis2:updateProject" style="document"/>
        <wsdl:input>
            <soap:body use="literal" namespace="http://pm.ppm.mercury.com"/>
        </wsdl:input>
    </wsdl:operation>
    <wsdl:operation name="updateProjectStatus">
        <soap:operation soapAction="axis2:updateProjectStatus" style="document"/>
        <wsdl:input>
            <soap:body use="literal" namespace="http://pm.ppm.mercury.com"/>
        </wsdl:input>
    </wsdl:operation>
    <wsdl:operation name="createBlankWorkPlan">
        <soap:operation soapAction="axis2:createBlankWorkPlan" style="document"/>
        <wsdl:input>
            <soap:body use="literal" namespace="http://pm.ppm.mercury.com"/>
        </wsdl:input>
    </wsdl:operation>
    <wsdl:operation name="createWorkPlanFromTemplate">
        <soap:operation soapAction="axis2:createWorkPlanFromTemplate" style="document"/>
        <wsdl:input>
            <soap:body use="literal" namespace="http://pm.ppm.mercury.com"/>
        </wsdl:input>
    </wsdl:operation>
    <wsdl:operation name="addTaskToExistingWorkPlan">
        <soap:operation soapAction="axis2:addTaskToExistingWorkPlan" style="document"/>
        <wsdl:input>
            <soap:body use="literal" namespace="http://pm.ppm.mercury.com"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" namespace="http://pm.ppm.mercury.com"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="importWorkPlanTasks">
        <soap:operation soapAction="axis2:importWorkPlanTasks" style="document"/>
        <wsdl:input>
            <soap:body use="literal" namespace="http://pm.ppm.mercury.com"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" namespace="http://pm.ppm.mercury.com"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="searchTasks">
        <soap:operation soapAction="axis2:searchTasks" style="document"/>
        <wsdl:input>
            <soap:body use="literal" namespace="http://pm.ppm.mercury.com"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" namespace="http://pm.ppm.mercury.com"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="readTasks">
        <soap:operation soapAction="axis2:readTasks" style="document"/>
        <wsdl:input>
            <soap:body use="literal" namespace="http://pm.ppm.mercury.com"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" namespace="http://pm.ppm.mercury.com"/>
        </wsdl:output>
    </wsdl:operation>

```

```

    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="updateTaskActuals">
    <soap:operation soapAction="axis2:importWorkPlanTasks" style="document"/>
    <wsdl:input>
      <soap:body use="literal" namespace="http://pm.ppm.mercury.com"/>
    </wsdl:input>
  </wsdl:operation>
</wsdl:binding>
<wsdl:binding name="ProjectServiceSOAP12Binding" type="axis2:ProjectServicePortType">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="createProject">
    <soap:operation soapAction="axis2:createProject" style="document"/>
    <wsdl:input>
      <soap:body use="literal" namespace="http://pm.ppm.mercury.com"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" namespace="http://pm.ppm.mercury.com"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="updateProject">
    <soap:operation soapAction="axis2:updateProject" style="document"/>
    <wsdl:input>
      <soap:body use="literal" namespace="http://pm.ppm.mercury.com"/>
    </wsdl:input>
  </wsdl:operation>
  <wsdl:operation name="updateProjectStatus">
    <soap:operation soapAction="axis2:updateProjectStatus" style="document"/>
    <wsdl:input>
      <soap:body use="literal" namespace="http://pm.ppm.mercury.com"/>
    </wsdl:input>
  </wsdl:operation>
  <wsdl:operation name="createBlankWorkPlan">
    <soap:operation soapAction="axis2:createBlankWorkPlan" style="document"/>
    <wsdl:input>
      <soap:body use="literal" namespace="http://pm.ppm.mercury.com"/>
    </wsdl:input>
  </wsdl:operation>
  <wsdl:operation name="createWorkPlanFromTemplate">
    <soap:operation soapAction="axis2:createWorkPlanFromTemplate" style="document"/>
    <wsdl:input>
      <soap:body use="literal" namespace="http://pm.ppm.mercury.com"/>
    </wsdl:input>
  </wsdl:operation>
  <wsdl:operation name="addTaskToExistingWorkPlan">
    <soap:operation soapAction="axis2:addTaskToExistingWorkPlan" style="document"/>
    <wsdl:input>
      <soap:body use="literal" namespace="http://pm.ppm.mercury.com"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" namespace="http://pm.ppm.mercury.com"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="importWorkPlanTasks">
    <soap:operation soapAction="axis2:importWorkPlanTasks" style="document"/>
    <wsdl:input>
      <soap:body use="literal" namespace="http://pm.ppm.mercury.com"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" namespace="http://pm.ppm.mercury.com"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="searchTasks">
    <soap:operation soapAction="axis2:searchTasks" style="document"/>
    <wsdl:input>
      <soap:body use="literal" namespace="http://pm.ppm.mercury.com"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" namespace="http://pm.ppm.mercury.com"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="readTasks">
    <soap:operation soapAction="axis2:readTasks" style="document"/>
    <wsdl:input>
      <soap:body use="literal" namespace="http://pm.ppm.mercury.com"/>
    </wsdl:input>
  </wsdl:operation>

```

```

        <wsdl:output>
            <soap:body use="literal" namespace="http://pm.ppm.mercury.com"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="updateTaskActuals">
        <soap:operation soapAction="axis2:importWorkPlanTasks" style="document"/>
        <wsdl:input>
            <soap:body use="literal" namespace="http://pm.ppm.mercury.com"/>
        </wsdl:input>
    </wsdl:operation>
</wsdl:binding>
<wsdl:service name="ProjectService">
    <wsdl:port name="ProjectServiceSOAP11port" binding="axis2:ProjectServiceSOAP11Binding">
        <soap:address location="http://localhost:8080/itg/ppmservices/ProjectService"/>
    </wsdl:port>
    <wsdl:port name="ProjectServiceSOAP12port" binding="axis2:ProjectServiceSOAP12Binding">
        <soap12:address location="http://localhost:8080/itg/ppmservices/ProjectService"/>
    </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

Version 7.1 XSD File

```

<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://mercury.com/ppm/pm/1.0" xmlns:xsd="http://www.w3.org/2001/
XMLSchema" xmlns:pm="http://mercury.com/ppm/pm/1.0" xmlns:common="http://mercury.com/ppm/common/
1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="qualified">
    <import namespace="http://mercury.com/ppm/common/1.0" schemaLocation="../xsd/Common.xsd"/>
    <complexType name="projectType">
        <sequence>
            <element name="projectName" type="xsd:string" minOccurs="0"/>
            <choice minOccurs="0">
                <element name="projectTypeName" type="xsd:string" minOccurs="0"/>
                <element name="projectTypeId" type="xsd:long" minOccurs="0"/>
            </choice>
            <element name="projectManagerUserName" type="xsd:string" minOccurs="0"
maxOccurs="unbounded"/>
            <element name="plannedStartPeriodFullName" type="xsd:string" minOccurs="0"/>
            <element name="plannedFinishPeriodFullName" type="xsd:string" minOccurs="0"/>
            <element name="regionName" type="xsd:string" minOccurs="0"/>
            <element name="staffingProfileName" type="xsd:string" minOccurs="0"/>
            <element name="budgetName" type="xsd:string" minOccurs="0"/>
            <element name="financialBenefitName" type="xsd:string" minOccurs="0"/>
            <choice minOccurs="0" maxOccurs="unbounded">
                <element name="simpleFields" type="pm:simpleFields" minOccurs="0" maxOccurs="unbounded"/
>
            </choice>
            <element name="tables" type="common:Table" minOccurs="0" maxOccurs="unbounded"/>
            </choice>
            <element name="pmReferences" type="pm:pmReference" minOccurs="0" maxOccurs="unbounded"/>
            <element name="notes" type="common:Note" minOccurs="0" maxOccurs="unbounded"/>
            <element name="fieldChangeNotes" type="pm:fieldChangeNote" minOccurs="0"
maxOccurs="unbounded"/>
        </sequence>
    </complexType>
    <complexType name="createProjectResultType">
        <sequence>
            <element name="projectId" type="xsd:long"/>
            <element name="requestId" type="xsd:integer"/>
        </sequence>
    </complexType>
    <complexType name="projectInputType">
        <sequence>
            <choice minOccurs="1">
                <element name="projectName" type="xsd:string" minOccurs="0"/>
                <element name="projectId" type="xsd:long" minOccurs="0"/>
                <element name="requestId" type="xsd:integer" minOccurs="0"/>
            </choice>
        </sequence>
    </complexType>

```



```

<complexType name="workPlanInputType">
  <sequence>
    <choice minOccurs="1">
      <element name="projectName" type="xsd:string"/>
      <element name="projectId" type="xsd:long"/>
    </choice>
  </sequence>
</complexType>

<complexType name="taskAnchors">
  <sequence>
    <element name="topAnchor" type="pm:anchorType"/>
    <element name="bottomAnchor" type="pm:anchorType" minOccurs="0" />
  </sequence>
</complexType>

<complexType name="addTaskResultType">
  <sequence>
    <element name="taskSequenceNumber" type="xsd:long"/>
    <element name="taskId" type="xsd:long"/>
  </sequence>
</complexType>

<complexType name="searchTaskPreferenceType">
  <sequence>
    <element name="startSearchPosition">
      <simpleType>
        <restriction base="xsd:int">
          <minInclusive value="1"/>
        </restriction>
      </simpleType>
    </element>
    <element name="maximumTasksToShow">
      <simpleType>
        <restriction base="xsd:int">
          <minInclusive value="1"/>
        </restriction>
      </simpleType>
    </element>
    <element name="taskNamePrefix" type="xsd:string" minOccurs="0"/>
    <element name="projectNames" type="xsd:string" minOccurs="0" maxOccurs="unbounded"/>
    <element name="resourceUserNames" type="xsd:string" minOccurs="0" maxOccurs="unbounded"/>
    <element name="includeFinishedTask" type="xsd:boolean" minOccurs="0"/>
    <element name="showOnlyExceptionTask" type="xsd:boolean" minOccurs="0"/>
    <element name="showOnlyMilestone" type="xsd:boolean" minOccurs="0"/>
    <element name="scheduledStartFrom" type="xsd:dateTime" minOccurs="0"/>
    <element name="scheduledStartTo" type="xsd:dateTime" minOccurs="0"/>
    <element name="scheduledFinishFrom" type="xsd:dateTime" minOccurs="0"/>
    <element name="scheduledFinishTo" type="xsd:dateTime" minOccurs="0"/>
  </sequence>
</complexType>

<complexType name="updateActualsInput">
  <sequence>
    <element name="taskId" type="xsd:long"/>
    <element name="assignment" type="pm:assignmentType"/>
  </sequence>
</complexType>

<complexType name="taskType">
  <sequence>
    <element name="taskId" type="xsd:long" minOccurs="0"/>
    <element name="taskName" type="xsd:string"/>
    <element name="taskSequence" type="xsd:long"/>
    <element name="outlineLevel" type="xsd:long"/>
    <element name="taskStatus" minOccurs="0">
      <simpleType>
        <restriction base="xsd:string">
          <enumeration value="in-progress"/>
          <enumeration value="completed"/>
          <enumeration value="in-planning"/>
          <enumeration value="ready"/>
          <enumeration value="on-hold"/>
          <enumeration value="active"/>
          <enumeration value="cancelled"/>
          <enumeration value="pending-predecessor"/>
          <enumeration value="completed-pending-predecessor"/>
          <enumeration value="completed-pending-request"/>
          <enumeration value="pending-request"/>
        </restriction>
      </simpleType>
    </element>
  </sequence>
</complexType>

```

```

        </restriction>
    </simpleType>
</element>
<element name="description" type="xsd:string" minOccurs="0"/>
<element name="priority" type="xsd:long" minOccurs="0"/>
<element name="isMilestone" type="xsd:boolean" minOccurs="0"/>
<element name="isMajorMilestone" type="xsd:boolean" minOccurs="0"/>
<element name="isMilestoneAutomaticallyCompletes" type="xsd:boolean" minOccurs="0"/>
<element name="ismilestoneManualConversion" type="xsd:boolean" minOccurs="0"/>
<element name="activity" type="pm:activityType" minOccurs="0"/>
<element name="owners" type="pm:resourceType" minOccurs="0" maxOccurs="unbounded"/>
<element name="workUnits" type="pm:assignmentType" minOccurs="0" maxOccurs="unbounded"/>
<element name="actuals" type="pm:taskActualType" minOccurs="0"/>
<element name="roleBean" type="pm:roleInfo" minOccurs="0"/>
<element name="schedulingBean" type="pm:scheduleInfo"/>
<element name="costBean" type="pm:costInfo" minOccurs="0"/>
<element name="predecessors" type="pm:dependencyInfo" minOccurs="0" maxOccurs="unbounded"/>
<element name="notificationSetupBean" type="pm:notificationSetupInfo" minOccurs="0"/>
<element name="userData" type="pm:userDataInfo" minOccurs="0"/>
<element name="skillProficiencies" type="pm:skillProficiencyInfo" minOccurs="0"
maxOccurs="unbounded"/>
<element name="notes" type="pm:taskNoteInfo" minOccurs="0" maxOccurs="unbounded"/>
<element name="pmReferences" type="pm:pmReference" minOccurs="0" maxOccurs="unbounded"/>
</sequence>
</complexType>
<complexType name="taskProperty">
<sequence/>
</complexType>
<complexType name="activityType">
<sequence>
<element name="activityId" type="xsd:long"/>
<element name="activityName" type="xsd:string" minOccurs="0"/>
</sequence>
</complexType>
<complexType name="resourceType">
<sequence>
<element name="resourceId" type="xsd:long"/>
<element name="firstName" type="xsd:string" minOccurs="0"/>
<element name="lastName" type="xsd:string" minOccurs="0"/>
</sequence>
</complexType>
<element name="schedulingBean" type="pm:scheduleInfo"/>
<complexType name="scheduleInfo">
<sequence>
<element name="scheduledDuration" type="xsd:double" minOccurs="0"/>
<element name="scheduledEffort" type="xsd:double"/>
<element name="scheduledStart" type="xsd:dateTime"/>
<element name="scheduledFinish" type="xsd:dateTime"/>
<element name="constraintType" minOccurs="0">
<simpleType>
<restriction base="xsd:string">
<enumeration value="as-soon-as-possible"/>
<enumeration value="as-late-as-possible"/>
<enumeration value="start-no-earlier-than"/>
<enumeration value="start-no-later-than"/>
<enumeration value="finish-no-earlier-than"/>
<enumeration value="finish-no-later-than"/>
<enumeration value="must-start-on"/>
<enumeration value="must-finish-on"/>
</restriction>
</simpleType>
</element>
<element name="constraintDate" type="xsd:dateTime" minOccurs="0"/>
</sequence>
</complexType>
<complexType name="taskActualType">
<sequence>
<element name="actualDuration" type="xsd:double" minOccurs="0"/>
<element name="percentComplete" type="xsd:double" minOccurs="0"/>
<element name="actualEffort" type="xsd:double" minOccurs="0"/>
<element name="estimatedRemainingEffort" type="xsd:double" minOccurs="0"/>
<element name="actualStart" type="xsd:dateTime" minOccurs="0"/>
<element name="estimatedFinish" type="xsd:dateTime" minOccurs="0"/>
<element name="actualFinish" type="xsd:dateTime" minOccurs="0"/>
</sequence>
</complexType>

```

```

<complexType name="moneyInfo">
  <sequence>
    <element name="localValue" type="xsd:decimal"/>
    <element name="localCurrencyCode" type="xsd:string"/>
  </sequence>
</complexType>
<element name="costBean" type="pm:costInfo"/>
<complexType name="costInfo">
  <sequence>
    <element name="actualCapLaborMoney" type="pm:moneyInfo" minOccurs="0"/>
    <element name="actualCapNonLaborMoney" type="pm:moneyInfo" minOccurs="0"/>
    <element name="actualOpLaborMoney" type="pm:moneyInfo" minOccurs="0"/>
    <element name="actualOpNonLaborMoney" type="pm:moneyInfo" minOccurs="0"/>
    <element name="plannedCapLaborMoney" type="pm:moneyInfo" minOccurs="0"/>
    <element name="plannedCapNonLaborMoney" type="pm:moneyInfo" minOccurs="0"/>
    <element name="plannedOpLaborMoney" type="pm:moneyInfo" minOccurs="0"/>
    <element name="plannedOpNonLaborMoney" type="pm:moneyInfo" minOccurs="0"/>
  </sequence>
</complexType>
<complexType name="roleInfo">
  <sequence>
    <element name="roleId" type="xsd:long"/>
    <element name="roleName" type="xsd:string" minOccurs="0"/>
  </sequence>
</complexType>
<element name="predecessor" type="pm:dependencyInfo"/>
<complexType name="dependencyInfo">
  <sequence>
    <element name="predTaskSeq" type="xsd:long"/>
    <element name="predRelationType">
      <simpleType>
        <restriction base="xsd:string">
          <enumeration value="start-to-finish"/>
          <enumeration value="finish-to-finish"/>
          <enumeration value="start-to-start"/>
          <enumeration value="finish-to-start"/>
        </restriction>
      </simpleType>
    </element>
    <element name="lagInDays" type="xsd:double" minOccurs="0"/>
  </sequence>
</complexType>
<complexType name="assignmentType">
  <sequence>
    <element name="taskId" type="xsd:long" minOccurs="0"/>
    <element name="scheduledEffort" type="xsd:double"/>
    <element name="resource" type="pm:resourceType" minOccurs="0"/>
    <element name="actualDuration" type="xsd:double" minOccurs="0"/>
    <element name="percentComplete" minOccurs="0">
      <simpleType>
        <restriction base="xsd:double">
          <minInclusive value="0"/>
          <maxInclusive value="100"/>
        </restriction>
      </simpleType>
    </element>
    <element name="actualEffort" type="xsd:double" minOccurs="0"/>
    <element name="estimatedRemainingEffort" type="xsd:double" minOccurs="0"/>
    <element name="actualStart" type="xsd:dateTime" minOccurs="0"/>
    <element name="estimatedFinish" type="xsd:dateTime" minOccurs="0"/>
    <element name="actualFinish" type="xsd:dateTime" minOccurs="0"/>
  </sequence>
</complexType>
<element name="notificationSetupBean" type="pm:notificationSetupInfo"/>
<complexType name="notificationSetupInfo">
  <sequence>
    <element name="isNotifyResourcesBeforeStart" type="xsd:boolean" minOccurs="0"/>
    <element name="isNotifyResourcesIfLate" type="xsd:boolean" minOccurs="0"/>
    <element name="isNotifyIfLate" type="xsd:boolean" minOccurs="0"/>
    <element name="isNotifyOnMilestone" type="xsd:boolean" minOccurs="0"/>
    <element name="isNotifyOnException" type="xsd:boolean" minOccurs="0"/>
    <element name="notifyResourcesDaysBeforeStart" type="xsd:int" minOccurs="0"/>
    <element name="notifyResourcesIfLateFrequency" type="pm:frequencyInfo" minOccurs="0"/>
    <element name="notifyIfLateByDays" type="xsd:int" minOccurs="0"/>
    <element name="notifyIfLateFrequency" type="pm:frequencyInfo" minOccurs="0"/>
  </sequence>
</complexType>

```

```

        <element name="notifyIfLateResources" type="pm:resourceType" minOccurs="0"
maxOccurs="unbounded"/>
        <element name="notifyIfLateSecurityGroups" type="pm:securityGroupInfo" minOccurs="0"
maxOccurs="unbounded"/>
        <element name="notifyOnMilestoneResources" type="pm:resourceType" minOccurs="0"
maxOccurs="unbounded"/>
        <element name="notifyOnMilestoneSecurityGroups" type="pm:securityGroupInfo" minOccurs="0"
maxOccurs="unbounded"/>
        <element name="notifyOnExceptionResources" type="pm:resourceType" minOccurs="0"
maxOccurs="unbounded"/>
        <element name="notifyOnExceptionSecurityGroups" type="pm:securityGroupInfo" minOccurs="0"
maxOccurs="unbounded"/>
        <element name="notifyIfLateParticipants" type="pm:participantInfo" />
        <element name="notifyOnMilestoneParticipants" type="pm:participantInfo" />
        <element name="notifyOnExceptionParticipants" type="pm:participantInfo" />
    </sequence>
</complexType>
<element name="frequency" type="pm:frequencyInfo"/>
<simpleType name="frequencyInfo">
    <restriction base="xsd:string">
        <enumeration value="Notify daily"/>
        <enumeration value="Notify hourly"/>
        <enumeration value="Notify immediately"/>
    </restriction>
</simpleType>

<element name="participants" type="pm:participantInfo"/>
    <complexType name="participantInfo">
        <sequence>
            <element name="participantType" type="pm:participantType" nillable="true"
minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
    </complexType>

<element name="participantType" type="pm:participantType"/>
<simpleType name="participantType">
    <restriction base="xsd:string">
        <enumeration value="StakeHolders"/>
        <enumeration value="ProjectManagers"/>
        <enumeration value="SummaryTaskOwners"/>
    </restriction>
</simpleType>
<complexType name="securityGroupInfo">
    <sequence>
        <element name="groupId" type="xsd:long"/>
        <element name="groupName" type="xsd:string"/>
    </sequence>
</complexType>
<element name="userData" type="pm:userDataInfo"/>
<complexType name="userDataInfo">
    <sequence>
        <element name="userData1" type="xsd:string" minOccurs="0"/>
        <element name="userData2" type="xsd:string" minOccurs="0"/>
        <element name="userData3" type="xsd:string" minOccurs="0"/>
        <element name="userData4" type="xsd:string" minOccurs="0"/>
        <element name="userData5" type="xsd:string" minOccurs="0"/>
        <element name="userData6" type="xsd:string" minOccurs="0"/>
        <element name="userData7" type="xsd:string" minOccurs="0"/>
        <element name="userData8" type="xsd:string" minOccurs="0"/>
        <element name="userData9" type="xsd:string" minOccurs="0"/>
        <element name="userData10" type="xsd:string" minOccurs="0"/>
        <element name="userData11" type="xsd:string" minOccurs="0"/>
        <element name="userData12" type="xsd:string" minOccurs="0"/>
        <element name="userData13" type="xsd:string" minOccurs="0"/>
        <element name="userData14" type="xsd:string" minOccurs="0"/>
        <element name="userData15" type="xsd:string" minOccurs="0"/>
        <element name="userData16" type="xsd:string" minOccurs="0"/>
        <element name="userData17" type="xsd:string" minOccurs="0"/>
        <element name="userData18" type="xsd:string" minOccurs="0"/>
        <element name="userData19" type="xsd:string" minOccurs="0"/>
        <element name="userData20" type="xsd:string" minOccurs="0"/>
        <element name="visUserData1" type="xsd:string" minOccurs="0"/>
        <element name="visUserData2" type="xsd:string" minOccurs="0"/>
        <element name="visUserData3" type="xsd:string" minOccurs="0"/>
        <element name="visUserData4" type="xsd:string" minOccurs="0"/>
    </sequence>

```

```

<element name="visUserData5" type="xsd:string" minOccurs="0"/>
<element name="visUserData6" type="xsd:string" minOccurs="0"/>
<element name="visUserData7" type="xsd:string" minOccurs="0"/>
<element name="visUserData8" type="xsd:string" minOccurs="0"/>
<element name="visUserData9" type="xsd:string" minOccurs="0"/>
<element name="visUserData10" type="xsd:string" minOccurs="0"/>
<element name="visUserData11" type="xsd:string" minOccurs="0"/>
<element name="visUserData12" type="xsd:string" minOccurs="0"/>
<element name="visUserData13" type="xsd:string" minOccurs="0"/>
<element name="visUserData14" type="xsd:string" minOccurs="0"/>
<element name="visUserData15" type="xsd:string" minOccurs="0"/>
<element name="visUserData16" type="xsd:string" minOccurs="0"/>
<element name="visUserData17" type="xsd:string" minOccurs="0"/>
<element name="visUserData18" type="xsd:string" minOccurs="0"/>
<element name="visUserData19" type="xsd:string" minOccurs="0"/>
<element name="visUserData20" type="xsd:string" minOccurs="0"/>
</sequence>
</complexType>
<complexType name="skillProficiencyInfo">
  <sequence>
    <element name="proficiencyLevel" type="xsd:int"/>
    <element name="skill" type="pm:skillInfo"/>
  </sequence>
</complexType>
<element name="skill" type="pm:skillInfo"/>
<complexType name="skillInfo">
  <sequence>
    <element name="skillId" type="xsd:long"/>
  </sequence>
</complexType>
<element name="anchor" type="pm:anchorType"/>
<complexType name="anchorType">
  <sequence>
    <element name="taskSequeceNumber" type="xsd:int"/>
    <element name="outLineLevel" type="xsd:long"/>
  </sequence>
</complexType>
<complexType name="taskNoteInfo">
  <sequence>
    <element name="noteText" type="xsd:string"/>
  </sequence>
</complexType>
<element name="pmReference" type="pm:pmReference"/>
<complexType name="pmReference">
  <sequence>
    <element name="targetTypeCode" type="xsd:int"/>
    <element name="relBehaviorCode">
      <simpleType>
        <restriction base="xsd:string">
          <enumeration value="INFORMATIONAL"/>
          <enumeration value="SUCCESSOR"/>
          <enumeration value="PREDECESSOR"/>
          <enumeration value="FF_PREDECESSOR"/>
          <enumeration value="AUTO_UPDATE"/>
          <enumeration value="UPDATED_BY_STEP"/>
        </restriction>
      </simpleType>
    </element>
    <element name="targetId" type="xsd:int" minOccurs="0"/>
    <element name="refRelationshipId" type="xsd:int"/>
    <element name="referenceName" type="xsd:string" minOccurs="0"/>
    <element name="refDescription" type="xsd:string" minOccurs="0"/>
    <!--targetId can be null for url and attachment -->
    <!--For url reference, uri is stored in referenceName -->
    <!--For file attachment, filename is stored in referenceName -->
    <!--For other reeference which doesn't have a name, id is stored in referenceName -->
  </sequence>
</complexType>
<!-- The following are DM related data types -->
<element name="field" type="pm:field"/>
<!-- complexType name="field" abstract="true" -->
<complexType name="field">
  <sequence>
    <element name="token" type="xsd:string"/>
  </sequence>
</complexType>

```

```

<element name="simpleFields" type="pm:simpleFields"/>
<complexType name="simpleFields">
  <complexContent>
    <extension base="common:Field">
      <choice maxOccurs="unbounded">
        <element name="stringValue" type="xsd:string" maxOccurs="unbounded"/>
        <element name="dateValue" type="xsd:dateTime"/>
      </choice>
    </extension>
  </complexContent>
</complexType>
<element name="fieldChangeNote" type="pm:fieldChangeNote"/>
<complexType name="fieldChangeNote">
  <complexContent>
    <extension base="common:Note">
      <sequence>
        <element name="fieldPrompt" type="xsd:string"/>
        <element name="oldValue" type="xsd:string"/>
        <element name="newValue" type="xsd:string"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
</schema>

```

B HP Resource Management Web Service Files

Version 7.1 WSDL File

```
<wsdl:definitions xmlns:rmMsg="http://mercury.com/ppm/rm/msg/1.0"
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
  xmlns:rs="http://mercury.com/ppm/rm/service/1.0"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  targetNamespace="http://mercury.com/ppm/rm/msg/1.0">
  <wsdl:types>
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
      xmlns:rm="http://mercury.com/ppm/rm/1.0"
      attributeFormDefault="qualified" elementFormDefault="qualified"
      targetNamespace="http://mercury.com/ppm/rm/service/1.0">
      <xs:import namespace="http://mercury.com/ppm/rm/1.0" schemaLocation="../xsd/Resource.xsd"/
    >
      <!-- Create Resource Pools types -->
      <xs:element name="createResourcePools">
        <xs:complexType>
          <xs:sequence>
            <xs:element maxOccurs="unbounded" minOccurs="0" name="resourcePool" nillable="true"
type="rm:ResourcePool" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="createResourcePoolsResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element maxOccurs="unbounded" minOccurs="0" name="resourcePoolRef"
nillable="true" type="rm:ResourcePoolReference" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <!-- Search Resource Pools types -->
      <xs:element name="searchResourcePools">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="filter" type="rm:ResourcePoolSearchFilter" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="searchResourcePoolsResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element maxOccurs="unbounded" minOccurs="0" name="resourcePool" nillable="true"
type="rm:ResourcePool" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <!-- Get Resource Pools types -->
      <xs:element name="getResourcePools">
        <xs:complexType>
          <xs:sequence>
            <xs:element maxOccurs="unbounded" minOccurs="0" name="resourcePoolId"
nillable="true" type="xs:long" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="getResourcePoolsResponse">
```

```

        <xs:complexType>
          <xs:sequence>
            <xs:element maxOccurs="unbounded" minOccurs="0" name="resourcePool" nillable="true"
type="rm:ResourcePool" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <!-- Update Resource Pools types -->
      <xs:element name="updateResourcePools">
        <xs:complexType>
          <xs:sequence>
            <xs:element maxOccurs="unbounded" minOccurs="0" name="resourcePool" nillable="true"
type="rm:ResourcePool" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="updateResourcePoolsResponse"/>
      <!-- Get Resource Participation types -->
      <xs:element name="getResourceParticipation">
        <xs:complexType>
          <xs:sequence>
            <xs:element maxOccurs="unbounded" name="resource" nillable="true"
type="rm:ResourceReference" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="getResourceParticipationResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element maxOccurs="unbounded" name="resourceParticipation" nillable="true"
type="rm:ResourceParticipation" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <!-- Set Resource Participation types -->
      <xs:element name="setResourceParticipation">
        <xs:complexType>
          <xs:sequence>
            <xs:element maxOccurs="unbounded" name="resourceParticipation" nillable="true"
type="rm:ResourceParticipation" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="setResourceParticipationResponse"/>
    </xs:schema>
  </wsdl:types>

  <!-- Create Resource Pools messages -->
  <wsdl:message name="createResourcePoolsMessage">
    <wsdl:part name="part1" element="rs:createResourcePools" />
  </wsdl:message>
  <wsdl:message name="createResourcePoolsResponseMessage">
    <wsdl:part name="part1" element="rs:createResourcePoolsResponse" />
  </wsdl:message>
  <!-- Search Resource Pools messages -->
  <wsdl:message name="searchResourcePoolsMessage">
    <wsdl:part name="part1" element="rs:searchResourcePools" />
  </wsdl:message>
  <wsdl:message name="searchResourcePoolsResponseMessage">
    <wsdl:part name="part1" element="rs:searchResourcePoolsResponse" />
  </wsdl:message>
  <!-- Get Resource Pools messages -->
  <wsdl:message name="getResourcePoolsMessage">
    <wsdl:part name="part1" element="rs:getResourcePools" />
  </wsdl:message>
  <wsdl:message name="getResourcePoolsResponseMessage">
    <wsdl:part name="part1" element="rs:getResourcePoolsResponse" />
  </wsdl:message>
  <!-- Update Resource Pools messages -->
  <wsdl:message name="updateResourcePoolsMessage">
    <wsdl:part name="part1" element="rs:updateResourcePools" />
  </wsdl:message>
  <wsdl:message name="updateResourcePoolsResponseMessage">
    <wsdl:part name="part1" element="rs:updateResourcePoolsResponse" />
  </wsdl:message>

```



```

<!-- Get Resource Participation messages -->
<wsdl:message name="getResourceParticipationMessage">
  <wsdl:part name="part1" element="rs:getResourceParticipation" />
</wsdl:message>
<wsdl:message name="getResourceParticipationResponseMessage">
  <wsdl:part name="part1" element="rs:getResourceParticipationResponse" />
</wsdl:message>
<!-- Set Resource Participation messages -->
<wsdl:message name="setResourceParticipationMessage">
  <wsdl:part name="part1" element="rs:setResourceParticipation" />
</wsdl:message>
<wsdl:message name="setResourceParticipationResponseMessage">
  <wsdl:part name="part1" element="rs:setResourceParticipationResponse" />
</wsdl:message>

<wsdl:portType name="ResourceServicePortType">
  <!-- Create Resource Pools operation -->
  <wsdl:operation name="createResourcePools">
    <wsdl:input message="rmMsg:createResourcePoolsMessage" />
    <wsdl:output message="rmMsg:createResourcePoolsResponseMessage" />
  </wsdl:operation>
  <!-- Search Resource Pools operation -->
  <wsdl:operation name="searchResourcePools">
    <wsdl:input message="rmMsg:searchResourcePoolsMessage" />
    <wsdl:output message="rmMsg:searchResourcePoolsResponseMessage" />
  </wsdl:operation>
  <!-- Get Resource Pools operation -->
  <wsdl:operation name="getResourcePools">
    <wsdl:input message="rmMsg:getResourcePoolsMessage" />
    <wsdl:output message="rmMsg:getResourcePoolsResponseMessage" />
  </wsdl:operation>
  <!-- Update Resource Pools operation -->
  <wsdl:operation name="updateResourcePools">
    <wsdl:input message="rmMsg:updateResourcePoolsMessage" />
    <wsdl:output message="rmMsg:updateResourcePoolsResponseMessage" />
  </wsdl:operation>
  <!-- Get Resource Participation operation -->
  <wsdl:operation name="getResourceParticipation">
    <wsdl:input message="rmMsg:getResourceParticipationMessage" />
    <wsdl:output message="rmMsg:getResourceParticipationResponseMessage" />
  </wsdl:operation>
  <!-- Set Resource Participation operation -->
  <wsdl:operation name="setResourceParticipation">
    <wsdl:input message="rmMsg:setResourceParticipationMessage" />
    <wsdl:output message="rmMsg:setResourceParticipationResponseMessage" />
  </wsdl:operation>
</wsdl:portType>

<wsdl:binding name="ResourceServiceSOAP11Binding"
  type="rmMsg:ResourceServicePortType">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
  <!-- Create Resource Pools operation -->
  <wsdl:operation name="createResourcePools">
    <soap:operation soapAction="urn:createResourcePools" style="document" />
    <wsdl:input>
      <soap:body use="literal" namespace="http://mercury.com/ppm/rm/msg/1.0" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" namespace="http://mercury.com/ppm/rm/msg/1.0" />
    </wsdl:output>
  </wsdl:operation>
  <!-- Search Resource Pools operation -->
  <wsdl:operation name="searchResourcePools">
    <soap:operation soapAction="urn:searchResourcePools" style="document" />
    <wsdl:input>
      <soap:body use="literal" namespace="http://mercury.com/ppm/rm/msg/1.0" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" namespace="http://mercury.com/ppm/rm/msg/1.0" />
    </wsdl:output>
  </wsdl:operation>
  <!-- Get Resource Pools operation -->
  <wsdl:operation name="getResourcePools">
    <soap:operation soapAction="urn:getResourcePools" style="document" />
    <wsdl:input>
      <soap:body use="literal" namespace="http://mercury.com/ppm/rm/msg/1.0" />
    </wsdl:input>
  </wsdl:operation>

```

```

        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" namespace="http://mercury.com/ppm/rm/msg/1.0" />
        </wsdl:output>
    </wsdl:operation>
    <!-- Update Resource Pools operation -->
    <wsdl:operation name="updateResourcePools">
        <soap:operation soapAction="urn:updateResourcePools" style="document" />
        <wsdl:input>
            <soap:body use="literal" namespace="http://mercury.com/ppm/rm/msg/1.0" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" namespace="http://mercury.com/ppm/rm/msg/1.0" />
        </wsdl:output>
    </wsdl:operation>
    <!-- Get Resource Participation operation -->
    <wsdl:operation name="getResourceParticipation">
        <soap:operation soapAction="urn:getResourceParticipation" style="document" />
        <wsdl:input>
            <soap:body use="literal" namespace="http://mercury.com/ppm/rm/msg/1.0" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" namespace="http://mercury.com/ppm/rm/msg/1.0" />
        </wsdl:output>
    </wsdl:operation>
    <!-- Set Resource Participation operation -->
    <wsdl:operation name="setResourceParticipation">
        <soap:operation soapAction="urn:setResourceParticipation" style="document" />
        <wsdl:input>
            <soap:body use="literal" namespace="http://mercury.com/ppm/rm/msg/1.0" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" namespace="http://mercury.com/ppm/rm/msg/1.0" />
        </wsdl:output>
    </wsdl:operation>
</wsdl:binding>

<wsdl:binding name="ResourceServiceSOAP12Binding" type="rmMsg:ResourceServicePortType">
    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
    <!-- Create Resource Pools operation -->
    <wsdl:operation name="createResourcePools">
        <soap12:operation soapAction="urn:createResourcePools" style="document" />
        <wsdl:input>
            <soap12:body use="literal" namespace="http://mercury.com/ppm/rm/msg/1.0" />
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal" namespace="http://mercury.com/ppm/rm/msg/1.0" />
        </wsdl:output>
    </wsdl:operation>
    <!-- Search Resource Pools operation -->
    <wsdl:operation name="searchResourcePools">
        <soap12:operation soapAction="urn:searchResourcePools" style="document" />
        <wsdl:input>
            <soap12:body use="literal" namespace="http://mercury.com/ppm/rm/msg/1.0" />
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal" namespace="http://mercury.com/ppm/rm/msg/1.0" />
        </wsdl:output>
    </wsdl:operation>
    <!-- Get Resource Pools operation -->
    <wsdl:operation name="getResourcePools">
        <soap12:operation soapAction="urn:getResourcePools" style="document" />
        <wsdl:input>
            <soap12:body use="literal" namespace="http://mercury.com/ppm/rm/msg/1.0" />
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal" namespace="http://mercury.com/ppm/rm/msg/1.0" />
        </wsdl:output>
    </wsdl:operation>
    <!-- Update Resource Pools operation -->
    <wsdl:operation name="updateResourcePools">
        <soap12:operation soapAction="urn:updateResourcePools" style="document" />
        <wsdl:input>
            <soap12:body use="literal" namespace="http://mercury.com/ppm/rm/msg/1.0" />
        </wsdl:input>
        <wsdl:output>

```

```

        <soap12:body use="literal" namespace="http://mercury.com/ppm/rm/msg/1.0" />
    </wsdl:output>
</wsdl:operation>
<!-- Get Resource Participation operation -->
<wsdl:operation name="getResourceParticipation">
    <soap12:operation soapAction="urn:getResourceParticipation" style="document" />
    <wsdl:input>
        <soap12:body use="literal" namespace="http://mercury.com/ppm/rm/msg/1.0" />
    </wsdl:input>
    <wsdl:output>
        <soap12:body use="literal" namespace="http://mercury.com/ppm/rm/msg/1.0" />
    </wsdl:output>
</wsdl:operation>
<!-- Set Resource Participation operation -->
<wsdl:operation name="setResourceParticipation">
    <soap12:operation soapAction="urn:setResourceParticipation" style="document" />
    <wsdl:input>
        <soap12:body use="literal" namespace="http://mercury.com/ppm/rm/msg/1.0" />
    </wsdl:input>
    <wsdl:output>
        <soap12:body use="literal" namespace="http://mercury.com/ppm/rm/msg/1.0" />
    </wsdl:output>
</wsdl:operation>
</wsdl:binding>

<wsdl:service name="ResourceService">
    <wsdl:port name="ResourceServiceSOAP11port"
        binding="rmMsg:ResourceServiceSOAP11Binding">
        <soap:address location="http://localhost:8080/itg/ppmservices/ResourceService" />
    </wsdl:port>
    <wsdl:port name="ResourceServiceSOAP12port"
        binding="rmMsg:ResourceServiceSOAP12Binding">
        <soap:address location="http://localhost:8080/itg/ppmservices/ResourceService" />
    </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

Version 7.1 XSD File

```

<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:rm="http://mercury.com/ppm/rm/1.0"
xmlns:common="http://mercury.com/ppm/common/1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://mercury.com/ppm/rm/1.0" elementFormDefault="qualified"
attributeFormDefault="qualified">

    <import namespace="http://mercury.com/ppm/common/1.0" schemaLocation="../xsd/Common.xsd"/>

    <!-- ResourcePoolSearchFilter -->
    <xs:complexType name="ResourcePoolSearchFilter">
        <xs:sequence>
            <xs:element name="name" nillable="true" type="xs:string"/>
            <xs:element name="description" nillable="true" type="xs:string"/>
            <xs:element name="parent" nillable="true" type="rm:ResourcePoolReference"/>
            <xs:element name="manager" nillable="true" type="rm:ResourceReference"/>
            <xs:element name="region" nillable="true" type="rm:RegionReference"/>
            <xs:element name="orgUnit" nillable="true" type="rm:OrgUnitReference"/>
        </xs:sequence>
    </xs:complexType>

    <!-- ResourcePool -->
    <xs:element name="ResourcePool" type="rm:ResourcePool"/>
    <xs:complexType name="ResourcePool">
        <xs:sequence>
            <xs:element name="id" nillable="true" type="xs:long"/>
            <xs:element name="name" type="xs:string"/>
            <xs:element name="description" nillable="true" type="xs:string"/>
            <xs:element name="parent" nillable="true" type="rm:ResourcePoolReference"/>
            <xs:element name="region" type="rm:RegionReference"/>
            <xs:element name="orgUnit" nillable="true" type="rm:OrgUnitReference"/>
        </xs:sequence>
    </xs:complexType>

```

```

        <xs:element name="manager" type="rm:ResourceReference" nillable="true" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="note" type="common:Note" nillable="true" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="child" type="rm:ResourcePoolReference" nillable="true" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="ACEs" type="rm:ResourcePoolAccessControlBean" nillable="true"
minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>

<!-- ResourcePoolAccessControlBean -->
<xs:element name="ResourcePoolAccessControlBean" type="rm:ResourcePoolAccessControlBean"/>
<xs:complexType name="ResourcePoolAccessControlBean">
    <xs:sequence>
        <xs:element name="user" type="rm:ResourceReference"/>
        <xs:element name="canEditHeader" nillable="true" type="xs:boolean"/>
        <xs:element name="canEditUnnamedHeadCount" nillable="true" type="xs:boolean"/>
        <xs:element name="canEditSecurity" nillable="true" type="xs:boolean"/>
        <xs:element name="isReadOnly" nillable="true" type="xs:boolean"/>
    </xs:sequence>
</xs:complexType>

<!-- ResourceParticipation -->
<xs:element name="ResourceParticipation" type="rm:ResourceParticipation"/>
<xs:complexType name="ResourceParticipation">
    <xs:sequence>
        <xs:element name="resource" type="rm:ResourceReference"/>
        <xs:element name="distributionGroup" type="rm:ResourceDistributionGroup" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>

<!-- ResourceDistributionGroup -->
<xs:element name="ResourceDistributionGroup" type="rm:ResourceDistributionGroup"/>
<xs:complexType name="ResourceDistributionGroup">
    <xs:sequence>
        <xs:element name="start" type="xs:dateTime"/>
        <xs:element name="distribution" type="rm:ResourcePoolDistribution" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>

<!-- ResourcePoolDistribution -->
<xs:element name="ResourcePoolDistribution" type="rm:ResourcePoolDistribution"/>
<xs:complexType name="ResourcePoolDistribution">
    <xs:sequence>
        <xs:element name="resourcePool" type="rm:ResourcePoolReference"/>
        <xs:element name="percent" type="xs:float"/>
    </xs:sequence>
</xs:complexType>

<!-- ResourcePoolReference -->
<xs:element name="ResourcePoolReference" type="rm:ResourcePoolReference" />
<xs:complexType name="ResourcePoolReference">
    <xs:sequence>
        <xs:element name="id" type="xs:long" nillable="true" />
        <xs:element name="name" type="xs:string" nillable="true" />
    </xs:sequence>
</xs:complexType>

<!-- ResourceReference -->
<xs:element name="ResourceReference" type="rm:ResourceReference" />
<xs:complexType name="ResourceReference">
    <xs:sequence>
        <xs:element name="id" type="xs:long" nillable="true" />
        <xs:element name="name" type="xs:string" nillable="true" />
    </xs:sequence>
</xs:complexType>

<!-- OrgUnitReference -->
<xs:element name="OrgUnitReference" type="rm:OrgUnitReference" />
<xs:complexType name="OrgUnitReference">
    <xs:sequence>
        <xs:element name="id" type="xs:long" nillable="true" />
        <xs:element name="name" type="xs:string" nillable="true" />
    </xs:sequence>
</xs:complexType>

```

```
    </xs:sequence>
  </xs:complexType>

  <!-- RegionReference -->
  <xs:element name="RegionReference" type="rm:RegionReference" />
  <xs:complexType name="RegionReference">
    <xs:sequence>
      <xs:element name="id" type="xs:long" nillable="true" />
      <xs:element name="name" type="xs:string" nillable="true" />
    </xs:sequence>
  </xs:complexType>
</schema>
```


C HP Financial Management Web Service Files

Version 7.1 WSDL File

```
<wsdl:definitions xmlns:fmMsg="http://mercury.com/ppm/fm/msg/1.0" xmlns:soap12="http://
schemas.xmlsoap.org/wsdl/soap12/" xmlns:fs="http://mercury.com/ppm/fm/service/1.0"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
targetNamespace="http://mercury.com/ppm/fm/msg/1.0">
  <wsdl:types>
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:fm="http://mercury.com/ppm/fm/
1.0" attributeFormDefault="qualified" elementFormDefault="qualified" targetNamespace="http://
mercury.com/ppm/fm/service/1.0">
      <xs:import namespace="http://mercury.com/ppm/fm/1.0" schemaLocation="../xsd/Finance.xsd"/>
      <!-- Create Cost Rules types -->
      <xs:element name="createCostRules">
        <xs:complexType>
          <xs:sequence>
            <xs:element maxOccurs="unbounded" minOccurs="0" name="costRule" nillable="true"
type="fm:CostRule"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="createCostRulesResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element maxOccurs="unbounded" minOccurs="0" name="costRule" nillable="true"
type="fm:CostRule"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <!-- Search Cost Rules types -->
      <xs:element name="searchCostRules">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="filter" type="fm:CostRuleSearchFilter"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="searchCostRulesResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element maxOccurs="unbounded" minOccurs="0" name="costRule" nillable="true"
type="fm:CostRule"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <!-- Get Cost Rules types -->
      <xs:element name="getCostRules">
        <xs:complexType>
          <xs:sequence>
            <xs:element maxOccurs="unbounded" minOccurs="0" name="costRuleId" nillable="true"
type="xs:long"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="getCostRulesResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element maxOccurs="unbounded" minOccurs="0" name="costRule" nillable="true"
type="fm:CostRule"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:schema>
  </wsdl:types>

```

```

        </xs:sequence>
    </xs:complexType>
</xs:element>
<!-- Update Cost Rules types -->
<xs:element name="updateCostRules">
    <xs:complexType>
        <xs:sequence>
            <xs:element maxOccurs="unbounded" minOccurs="0" name="costRule" nillable="true"
type="fm:CostRule"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="updateCostRulesResponse"/>
<!-- Delete Cost Rules types -->
<xs:element name="deleteCostRules">
    <xs:complexType>
        <xs:sequence>
            <xs:element maxOccurs="unbounded" minOccurs="0" name="costRuleId" nillable="true"
type="xs:long"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="deleteCostRulesResponse"/>
<!-- Get Cost Factors types -->
<xs:element name="getCostFactors">
    <xs:complexType>
        </xs:complexType>
</xs:element>
<xs:element name="getCostFactorsResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element maxOccurs="unbounded" minOccurs="0" name="costFactor" nillable="true"
type="fm:CostFactor"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<!-- Set Cost Factors types -->
<xs:element name="setCostFactors">
    <xs:complexType>
        <xs:sequence>
            <xs:element maxOccurs="unbounded" minOccurs="0" name="costFactor" nillable="true"
type="fm:CostFactor"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="setCostFactorsResponse"/>
</xs:schema>
</wsdl:types>
<wsdl:message name="createCostRulesMessage">
    <wsdl:part name="part1" element="fs:createCostRules"/>
</wsdl:message>
<wsdl:message name="createCostRulesResponseMessage">
    <wsdl:part name="part1" element="fs:createCostRulesResponse"/>
</wsdl:message>
<wsdl:message name="searchCostRulesMessage">
    <wsdl:part name="part1" element="fs:searchCostRules"/>
</wsdl:message>
<wsdl:message name="searchCostRulesResponseMessage">
    <wsdl:part name="part1" element="fs:searchCostRulesResponse"/>
</wsdl:message>
<wsdl:message name="getCostRulesMessage">
    <wsdl:part name="part1" element="fs:getCostRules"/>
</wsdl:message>
<wsdl:message name="getCostRulesResponseMessage">
    <wsdl:part name="part1" element="fs:getCostRulesResponse"/>
</wsdl:message>
<wsdl:message name="updateCostRulesMessage">
    <wsdl:part name="part1" element="fs:updateCostRules"/>
</wsdl:message>
<wsdl:message name="updateCostRulesResponseMessage">
    <wsdl:part name="part1" element="fs:updateCostRulesResponse"/>
</wsdl:message>
<wsdl:message name="deleteCostRulesMessage">
    <wsdl:part name="part1" element="fs:deleteCostRules"/>
</wsdl:message>
<wsdl:message name="deleteCostRulesResponseMessage">

```



```

    <wsdl:part name="part1" element="fs:deleteCostRulesResponse"/>
</wsdl:message>
<wsdl:message name="getCostFactorsMessage">
  <wsdl:part name="part1" element="fs:getCostFactors"/>
</wsdl:message>
<wsdl:message name="getCostFactorsResponseMessage">
  <wsdl:part name="part1" element="fs:getCostFactorsResponse"/>
</wsdl:message>
<wsdl:message name="setCostFactorsMessage">
  <wsdl:part name="part1" element="fs:setCostFactors"/>
</wsdl:message>
<wsdl:message name="setCostFactorsResponseMessage">
  <wsdl:part name="part1" element="fs:setCostFactorsResponse"/>
</wsdl:message>
<wsdl:portType name="FinanceServicePortType">
  <!-- Create Cost Rules operation -->
  <wsdl:operation name="createCostRules">
    <wsdl:input message="fmMsg:createCostRulesMessage"/>
    <wsdl:output message="fmMsg:createCostRulesResponseMessage"/>
  </wsdl:operation>
  <!-- Search Cost Rules operation -->
  <wsdl:operation name="searchCostRules">
    <wsdl:input message="fmMsg:searchCostRulesMessage"/>
    <wsdl:output message="fmMsg:searchCostRulesResponseMessage"/>
  </wsdl:operation>
  <!-- Get Cost Rules operation -->
  <wsdl:operation name="getCostRules">
    <wsdl:input message="fmMsg:getCostRulesMessage"/>
    <wsdl:output message="fmMsg:getCostRulesResponseMessage"/>
  </wsdl:operation>
  <!-- Update Cost Rules operation -->
  <wsdl:operation name="updateCostRules">
    <wsdl:input message="fmMsg:updateCostRulesMessage"/>
    <wsdl:output message="fmMsg:updateCostRulesResponseMessage"/>
  </wsdl:operation>
  <!-- Delete Cost Rules operation -->
  <wsdl:operation name="deleteCostRules">
    <wsdl:input message="fmMsg:deleteCostRulesMessage"/>
    <wsdl:output message="fmMsg:deleteCostRulesResponseMessage"/>
  </wsdl:operation>
  <!-- Get Cost Factors operation -->
  <wsdl:operation name="getCostFactors">
    <wsdl:input message="fmMsg:getCostFactorsMessage"/>
    <wsdl:output message="fmMsg:getCostFactorsResponseMessage"/>
  </wsdl:operation>
  <!-- Set Cost Factors operation -->
  <wsdl:operation name="setCostFactors">
    <wsdl:input message="fmMsg:setCostFactorsMessage"/>
    <wsdl:output message="fmMsg:setCostFactorsResponseMessage"/>
  </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="FinanceServiceSOAP11Binding" type="fmMsg:FinanceServicePortType">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <!-- Create Cost Rules operation -->
  <wsdl:operation name="createCostRules">
    <soap:operation soapAction="urn:createCostRules" style="document"/>
    <wsdl:input>
      <soap:body use="literal" namespace="http://mercury.com/ppm/fm/msg/1.0"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" namespace="http://mercury.com/ppm/fm/msg/1.0"/>
    </wsdl:output>
  </wsdl:operation>
  <!-- Search Cost Rules operation -->
  <wsdl:operation name="searchCostRules">
    <soap:operation soapAction="urn:searchCostRules" style="document"/>
    <wsdl:input>
      <soap:body use="literal" namespace="http://mercury.com/ppm/fm/msg/1.0"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" namespace="http://mercury.com/ppm/fm/msg/1.0"/>
    </wsdl:output>
  </wsdl:operation>
  <!-- Get Cost Rules operation -->
  <wsdl:operation name="getCostRules">
    <soap:operation soapAction="urn:getCostRules" style="document"/>

```

```

        <wsdl:input>
            <soap:body use="literal" namespace="http://mercury.com/ppm/fm/msg/1.0"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" namespace="http://mercury.com/ppm/fm/msg/1.0"/>
        </wsdl:output>
    </wsdl:operation>
    <!-- Update Cost Rules operation -->
    <wsdl:operation name="updateCostRules">
        <soap:operation soapAction="urn:updateCostRules" style="document"/>
        <wsdl:input>
            <soap:body use="literal" namespace="http://mercury.com/ppm/fm/msg/1.0"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" namespace="http://mercury.com/ppm/fm/msg/1.0"/>
        </wsdl:output>
    </wsdl:operation>
    <!-- Delete Cost Rules operation -->
    <wsdl:operation name="deleteCostRules">
        <soap:operation soapAction="urn:deleteCostRules" style="document"/>
        <wsdl:input>
            <soap:body use="literal" namespace="http://mercury.com/ppm/fm/msg/1.0"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" namespace="http://mercury.com/ppm/fm/msg/1.0"/>
        </wsdl:output>
    </wsdl:operation>
    <!-- Get Cost Factors operation -->
    <wsdl:operation name="getCostFactors">
        <soap:operation soapAction="urn:getCostFactors" style="document"/>
        <wsdl:input>
            <soap:body use="literal" namespace="http://mercury.com/ppm/fm/msg/1.0"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" namespace="http://mercury.com/ppm/fm/msg/1.0"/>
        </wsdl:output>
    </wsdl:operation>
    <!-- Set Cost Factors operation -->
    <wsdl:operation name="setCostFactors">
        <soap:operation soapAction="urn:setCostFactors" style="document"/>
        <wsdl:input>
            <soap:body use="literal" namespace="http://mercury.com/ppm/fm/msg/1.0"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" namespace="http://mercury.com/ppm/fm/msg/1.0"/>
        </wsdl:output>
    </wsdl:operation>
</wsdl:binding>
<wsdl:binding name="FinanceServiceSOAP12Binding" type="fmMsg:FinanceServicePortType">
    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
    <!-- Create Cost Rules operation -->
    <wsdl:operation name="createCostRules">
        <soap12:operation soapAction="urn:createCostRules" style="document"/>
        <wsdl:input>
            <soap12:body use="literal" namespace="http://mercury.com/ppm/fm/msg/1.0"/>
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal" namespace="http://mercury.com/ppm/fm/msg/1.0"/>
        </wsdl:output>
    </wsdl:operation>
    <!-- Search Cost Rules operation -->
    <wsdl:operation name="searchCostRules">
        <soap12:operation soapAction="urn:searchCostRules" style="document"/>
        <wsdl:input>
            <soap12:body use="literal" namespace="http://mercury.com/ppm/fm/msg/1.0"/>
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal" namespace="http://mercury.com/ppm/fm/msg/1.0"/>
        </wsdl:output>
    </wsdl:operation>
    <!-- Get Cost Rules operation -->
    <wsdl:operation name="getCostRules">
        <soap12:operation soapAction="urn:getCostRules" style="document"/>
        <wsdl:input>
            <soap12:body use="literal" namespace="http://mercury.com/ppm/fm/msg/1.0"/>
        </wsdl:input>
    </wsdl:operation>

```

```

        <wsdl:output>
          <soap12:body use="literal" namespace="http://mercury.com/ppm/fm/msg/1.0"/>
        </wsdl:output>
      </wsdl:operation>
      <!-- Update Cost Rules operation -->
      <wsdl:operation name="updateCostRules">
        <soap12:operation soapAction="urn:updateCostRules" style="document"/>
        <wsdl:input>
          <soap12:body use="literal" namespace="http://mercury.com/ppm/fm/msg/1.0"/>
        </wsdl:input>
        <wsdl:output>
          <soap12:body use="literal" namespace="http://mercury.com/ppm/fm/msg/1.0"/>
        </wsdl:output>
      </wsdl:operation>
      <!-- Delete Cost Rules operation -->
      <wsdl:operation name="deleteCostRules">
        <soap12:operation soapAction="urn:deleteCostRules" style="document"/>
        <wsdl:input>
          <soap12:body use="literal" namespace="http://mercury.com/ppm/fm/msg/1.0"/>
        </wsdl:input>
        <wsdl:output>
          <soap12:body use="literal" namespace="http://mercury.com/ppm/fm/msg/1.0"/>
        </wsdl:output>
      </wsdl:operation>
      <!-- Get Cost Factors operation -->
      <wsdl:operation name="getCostFactors">
        <soap12:operation soapAction="urn:getCostFactors" style="document"/>
        <wsdl:input>
          <soap12:body use="literal" namespace="http://mercury.com/ppm/fm/msg/1.0"/>
        </wsdl:input>
        <wsdl:output>
          <soap12:body use="literal" namespace="http://mercury.com/ppm/fm/msg/1.0"/>
        </wsdl:output>
      </wsdl:operation>
      <!-- Set Cost Factors operation -->
      <wsdl:operation name="setCostFactors">
        <soap12:operation soapAction="urn:setCostFactors" style="document"/>
        <wsdl:input>
          <soap12:body use="literal" namespace="http://mercury.com/ppm/fm/msg/1.0"/>
        </wsdl:input>
        <wsdl:output>
          <soap12:body use="literal" namespace="http://mercury.com/ppm/fm/msg/1.0"/>
        </wsdl:output>
      </wsdl:operation>
    </wsdl:binding>
    <wsdl:service name="FinanceService">
      <wsdl:port name="FinanceServiceSOAP11port" binding="fmMsg:FinanceServiceSOAP11Binding">
        <soap:address location="http://localhost:8080/itg/ppmservices/FinanceService"/>
      </wsdl:port>
      <wsdl:port name="FinanceServiceSOAP12port" binding="fmMsg:FinanceServiceSOAP12Binding">
        <soap12:address location="http://localhost:8080/itg/ppmservices/FinanceService"/>
      </wsdl:port>
    </wsdl:service>
    <!-- Create Cost Rules messages -->
    <!-- Search Cost Rules messages -->
    <!-- Get Cost Rules messages -->
    <!-- Update Cost Rules messages -->
    <!-- Delete Cost Rules messages -->
    <!-- Get Cost Factors messages -->
    <!-- Set Cost Factors messages -->
  </wsdl:definitions>

```

Version 7.1 XSD File

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:fm="http://mercury.com/ppm/fm/1.0"
xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="http://mercury.com/ppm/fm/1.0"
elementFormDefault="qualified" attributeFormDefault="qualified">

  <!-- CostRule -->
  <xs:element name="CostRule" type="fm:CostRule"/>
  <xs:complexType name="CostRule">
    <xs:sequence>
      <xs:element name="id" nillable="true" type="xs:long"/>
      <xs:element name="ruleNumber" nillable="true" type="xs:integer"/>
      <xs:element name="rate" type="fm:CostRate" minOccurs="1" maxOccurs="unbounded"/>
      <xs:element name="factor" nillable="true" type="fm:CostFactorValue" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <!-- CostRate -->
  <xs:element name="CostRate" type="fm:CostRate"/>
  <xs:complexType name="CostRate">
    <xs:sequence>
      <xs:element name="startDate" nillable="true" type="xs:dateTime"/>
      <xs:element name="endDate" nillable="true" type="xs:dateTime"/>
      <xs:element name="currencyCode" type="xs:string"/>
      <xs:element name="rate" type="xs:float"/>
    </xs:sequence>
  </xs:complexType>

  <!-- CostFactorValue -->
  <xs:element name="CostFactorValue" type="fm:CostFactorValue"/>
  <xs:complexType name="CostFactorValue">
    <xs:sequence>
      <xs:element name="factor" type="xs:string"/>
      <xs:element name="id" nillable="true" type="xs:string"/>
      <xs:element name="name" nillable="true" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>

  <!-- CostFactor -->
  <xs:element name="CostFactor" type="fm:CostFactor"/>
  <xs:complexType name="CostFactor">
    <xs:sequence>
      <xs:element name="factor" type="xs:string"/>
      <xs:element name="sortOrder" type="xs:integer"/>
    </xs:sequence>
  </xs:complexType>

  <!-- CostRuleSearchFilter -->
  <xs:element name="CostRuleSearchFilter" type="fm:CostRuleSearchFilter"/>
  <xs:complexType name="CostRuleSearchFilter">
    <xs:sequence>
      <xs:element name="effectiveStartDate" nillable="true" type="xs:dateTime"/>
      <xs:element name="effectiveEndDate" nillable="true" type="xs:dateTime"/>
      <xs:element name="factor" nillable="true" type="fm:CostFactorValue" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</schema>
```

Versions 6.0 and 7.0 WSDL File

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="http://mercury.com/itg/fm/1.0" xmlns="http://
schemas.xmlsoap.org/wsdl/" xmlns:apachesoap="http://xml.apache.org/xml-soap" xmlns:fmintf="http://
mercury.com/itg/fm/1.0" xmlns:fmtypes="http://mercury.com/itg/fm/1.0/types" xmlns:intf="http://
mercury.com/itg/dm/1.0" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:types="http://mercury.com/itg/dm/1.0/types" xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/"
xmlns:wSDLsoap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:xsd="http://www.w3.org/2001/
XMLSchema">

    <wsdl:types>

        <schema targetNamespace="http://mercury.com/itg/dm/1.0/types" xmlns="http://www.w3.org/
2001/XMLSchema">

            <import namespace="http://schemas.xmlsoap.org/soap/encoding/" />

            <complexType abstract="true" name="Value">
                <sequence />
            </complexType>

            <complexType name="RequestHeader">
                <sequence>
                    <element name="auditNote" nillable="false" type="xsd:string" />
                    <element name="origin" nillable="false" type="xsd:string" />
                    <element name="transactionName" nillable="false" type="xsd:string" />
                    <element name="credentials" nillable="false" type="types:Credentials" />
                </sequence>
            </complexType>

            <complexType abstract="true" name="Credentials">
                <sequence />
            </complexType>

            <complexType name="LoginAccount">
                <complexContent>
                    <extension base="types:Credentials">
                        <sequence>
                            <element name="username" nillable="false" type="xsd:string" />
                            <element name="password" nillable="false" type="xsd:string" />
                        </sequence>
                    </extension>
                </complexContent>
            </complexType>

            <complexType name="ProcessingError">
                <sequence>
                    <element name="errors" nillable="false" type="types:ArrayOf_types_
ErrorMessage" />
                </sequence>
            </complexType>
        </schema>
    </wsdl:types>
</wsdl:definitions>
```

```

        <element name="message" nillable="false" type="xsd:string"/>
    </sequence>
</complexType>
<complexType name="SingleValue">
    <complexContent>
        <extension base="types:Value">
            <sequence>
                <element name="tokenValue" nillable="false" type="xsd:string"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<complexType name="MultipleValues">
    <complexContent>
        <extension base="types:Value">
            <sequence>
                <element name="values" nillable="true" type="types:ArrayOf_xsd_
string"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<complexType name="DateValue">
    <complexContent>
        <extension base="types:Value">
            <sequence>
                <element name="date" nillable="true" type="xsd:dateTime"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<complexType name="Field">
    <sequence>
        <element name="token" nillable="false" type="xsd:string"/>
        <element name="value" nillable="false" type="types:Value"/>
    </sequence>
</complexType>
<complexType name="ErrorMessage">
    <sequence>

```

```

        <element name="code" nillable="true" type="xsd:int"/>
        <element name="message" nillable="true" type="xsd:string"/>
    </sequence>
</complexType>
<complexType name="ResponseMessage">
    <sequence>
        <element name="code" nillable="false" type="xsd:int"/>
        <element name="message" nillable="false" type="xsd:string"/>
    </sequence>
</complexType>
<complexType name="ArrayOf_types_Field">
    <complexContent>
        <restriction base="soapenc:Array">
            <attribute ref="soapenc:arrayType" wsdl:arrayType="types:Field[]"/>
        </restriction>
    </complexContent>
</complexType>
<complexType name="ArrayOf_xsd_string">
    <complexContent>
        <restriction base="soapenc:Array">
            <attribute ref="soapenc:arrayType" wsdl:arrayType="xsd:string[]"/>
        </restriction>
    </complexContent>
</complexType>
<complexType name="ArrayOf_types_ErrorMessage">
    <complexContent>
        <restriction base="soapenc:Array">
            <attribute ref="soapenc:arrayType"
wsdl:arrayType="types:ErrorMessage[]"/>
        </restriction>
    </complexContent>
</complexType>
</schema>

<schema targetNamespace="http://mercury.com/itg/fm/1.0/types" xmlns="http://www.w3.org/
2001/XMLSchema">
    <complexType name="BudgetFilters">
        <sequence>
            <element name="budgetName" nillable="true" type="xsd:string"/>

```

```

        <element name="budgetID" nillable="true" type="xsd:int"/>
        <element name="budgetStatuses" nillable="true" type="types:ArrayOf_xsd_
string"/>
        <element name="active" nillable="true" type="xsd:boolean"/>
        <element name="associatedWithType" nillable="true" type="xsd:string"/>
        <element name="associatedWithName" nillable="true" type="xsd:string"/>
        <element name="budgetUserData" nillable="true" type="types:ArrayOf_types_
Field"/>
    </sequence>
</complexType>
<complexType name="ReadMessage">
    <sequence>
        <element name="header" nillable="false" type="types:RequestHeader"/>
        <element name="budgetFilters" nillable="false"
type="fmtypes:BudgetFilters"/>
    </sequence>
</complexType>
<complexType name="ReadResponse">
    <sequence>
        <element name="header" nillable="false" type="types:RequestHeader"/>
        <element name="responseMessage" nillable="false"
type="types:ResponseMessage"/>
        <element name="Budget" nillable="true" type="fmtypes:ArrayOf_fmtypes_
Budget"/>
    </sequence>
</complexType>
<complexType name="CreateMessage">
    <sequence>
        <element name="header" nillable="false" type="types:RequestHeader"/>
        <element name="budget" nillable="false" type="fmtypes:Budget"/>
    </sequence>
</complexType>
<complexType name="UpdateMessage">
    <sequence>
        <element name="header" nillable="false" type="types:RequestHeader"/>
        <element name="budget" nillable="false" type="fmtypes:Budget"/>
    </sequence>
</complexType>
<complexType name="UpdateResponse">
    <sequence>
        <element name="header" nillable="false" type="types:RequestHeader"/>

```



```

        <element name="responseMessage" nillable="false"
type="types:ResponseMessage"/>
        <element name="budgetID" nillable="false" type="xsd:int"/>
    </sequence>
</complexType>
<complexType name="CreateResponse">
    <sequence>
        <element name="header" nillable="false" type="types:RequestHeader"/>
        <element name="responseMessage" nillable="false"
type="types:ResponseMessage"/>
        <element name="budgetID" nillable="false" type="xsd:int"/>
    </sequence>
</complexType>
<complexType name="BudgetLineDetail">
    <sequence>
        <element name="periodName" nillable="true" type="xsd:string"/>
        <element name="periodStartDate" nillable="false" type="xsd:dateTime"/>
        <element name="planValue" nillable="true" type="xsd:decimal"/>
        <element name="actualValue" nillable="true" type="xsd:decimal"/>
    </sequence>
</complexType>
<complexType name="BudgetLine">
    <sequence>
        <element name="budgetLineID" nillable="true" type="xsd:int"/>
        <element name="typeCode" nillable="false" type="xsd:string"/>
        <element name="categoryCode" nillable="false" type="xsd:string"/>
        <element name="expenseTypeCode" nillable="true" type="xsd:string"/>
        <element name="userData" nillable="true" type="types:ArrayOf_types_Field"/>
        <element name="budgetLineDetails" nillable="true" type="fmtypes:ArrayOf_
fmtypes_BudgetLineDetail"/>
    </sequence>
</complexType>
<complexType name="Budget">
    <sequence>
        <element name="budgetID" nillable="true" type="xsd:int"/>
        <element name="budgetName" nillable="false" type="xsd:string"/>
        <element name="capexOpexEnabledFlag" nillable="true" type="xsd:boolean"/>
        <element name="actualsRolledUpCode" nillable="true" type="xsd:string"/>
        <element name="active" nillable="true" type="xsd:boolean"/>
    </sequence>

```

```

        <element name="associatedWithType" nillable="true" type="xsd:string"/>
        <element name="associatedWithName" nillable="true" type="xsd:string"/>
        <element name="startPeriodName" nillable="true" type="xsd:string"/>
        <element name="endPeriodName" nillable="true" type="xsd:string"/>
        <element name="startPeriodStartDate" nillable="false" type="xsd:dateTime"/>
        <element name="endPeriodStartDate" nillable="false" type="xsd:dateTime"/>
        <element name="periodType" nillable="false" type="xsd:string"/>
        <element name="description" nillable="true" type="xsd:string"/>
        <element name="budgetStatus" nillable="false" type="xsd:string"/>
        <element name="regionName" nillable="false" type="xsd:string"/>
        <element name="baseCurrency" nillable="true" type="xsd:string"/>
        <element name="userData" nillable="true" type="types:ArrayOf_types_Field"/>
        <element name="budgetLines" nillable="true" type="fmtypes:ArrayOf_fmtypes_
BudgetLine"/>
    </sequence>
</complexType>
<complexType name="ArrayOf_fmtypes_BudgetLineDetail">
    <complexContent>
        <restriction base="soapenc:Array">
            <attribute ref="soapenc:arrayType"
wsdl:arrayType="fmtypes:BudgetLineDetail[]"/>
        </restriction>
    </complexContent>
</complexType>
<complexType name="ArrayOf_fmtypes_BudgetLine">
    <complexContent>
        <restriction base="soapenc:Array">
            <attribute ref="soapenc:arrayType"
wsdl:arrayType="fmtypes:BudgetLine[]"/>
        </restriction>
    </complexContent>
</complexType>
<complexType name="ArrayOf_fmtypes_Budget">
    <complexContent>
        <restriction base="soapenc:Array">
            <attribute ref="soapenc:arrayType" wsdl:arrayType="fmtypes:Budget[]"/>
        </restriction>
    </complexContent>
</complexType>
</schema>

```

```

</wsdl:types>

<wsdl:message name="ProcessingErrorFault">
  <wsdl:part name="fault" type="types:ProcessingError"/>
</wsdl:message>

<wsdl:message name="ReadResponse">
  <wsdl:part name="readReturn" type="fmtypes:ReadResponse"/>
</wsdl:message>

<wsdl:message name="ReadRequest">
  <wsdl:part name="readMessage" type="fmtypes:ReadMessage"/>
</wsdl:message>

<wsdl:message name="UpdateResponse">
  <wsdl:part name="updateReturn" type="fmtypes:UpdateResponse"/>
</wsdl:message>

<wsdl:message name="UpdateRequest">
  <wsdl:part name="updateMessage" type="fmtypes:UpdateMessage"/>
</wsdl:message>

<wsdl:message name="CreateResponse">
  <wsdl:part name="createReturn" type="fmtypes:CreateResponse"/>
</wsdl:message>

<wsdl:message name="CreateRequest">
  <wsdl:part name="createMessage" type="fmtypes:CreateMessage"/>
</wsdl:message>

<wsdl:portType name="FinanceServices">
  <wsdl:operation name="read" parameterOrder="readMessage">
    <wsdl:input message="fmintf:ReadRequest" name="readRequest"/>
    <wsdl:output message="fmintf:ReadResponse" name="readResponse"/>
    <wsdl:fault message="fmintf:ProcessingErrorFault" name="ProcessingErrorFault"/>
  </wsdl:operation>

  <wsdl:operation name="update" parameterOrder="updateMessage">
    <wsdl:input message="fmintf:UpdateRequest" name="updateRequest"/>
    <wsdl:output message="fmintf:UpdateResponse" name="updateResponse"/>
    <wsdl:fault message="fmintf:ProcessingErrorFault" name="ProcessingErrorFault"/>
  </wsdl:operation>

```

```

    <wsdl:operation name="create" parameterOrder="createMessage">
      <wsdl:input message="fmintf:CreateRequest" name="createRequest"/>
      <wsdl:output message="fmintf:CreateResponse" name="createResponse"/>
      <wsdl:fault message="fmintf:ProcessingErrorFault" name="ProcessingErrorFault"/>
    </wsdl:operation>
  </wsdl:portType>

  <wsdl:binding name="FinanceSoapBinding" type="fmintf:FinanceServices">
    <wsdlsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>

    <wsdl:operation name="read">
      <wsdlsoap:operation soapAction=""/>
      <wsdl:input name="readRequest">
        <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
          namespace="http://mercury.com/itg/fm/1.0" use="encoded"/>
      </wsdl:input>
      <wsdl:output name="readResponse">
        <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
          namespace="http://mercury.com/itg/fm/1.0" use="encoded"/>
      </wsdl:output>
      <wsdl:fault name="ProcessingErrorFault">
        <wsdlsoap:fault name="ProcessingErrorFault" use="encoded"/>
      </wsdl:fault>
    </wsdl:operation>

    <wsdl:operation name="update">
      <wsdlsoap:operation soapAction=""/>
      <wsdl:input name="updateRequest">
        <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
          namespace="http://mercury.com/itg/fm/1.0" use="encoded"/>
      </wsdl:input>
      <wsdl:output name="updateResponse">
        <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
          namespace="http://mercury.com/itg/fm/1.0" use="encoded"/>
      </wsdl:output>
      <wsdl:fault name="ProcessingErrorFault">
        <wsdlsoap:fault name="ProcessingErrorFault" use="encoded"/>
      </wsdl:fault>
    </wsdl:operation>

    <wsdl:operation name="create">
      <wsdlsoap:operation soapAction=""/>

```

```
        <wsdl:input name="createRequest">
            <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://mercury.com/itg/fm/1.0" use="encoded"/>
        </wsdl:input>
        <wsdl:output name="createResponse">
            <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://mercury.com/itg/fm/1.0" use="encoded"/>
        </wsdl:output>
        <wsdl:fault name="ProcessingErrorFault">
            <wsdlsoap:fault name="ProcessingErrorFault" use="encoded"/>
        </wsdl:fault>
    </wsdl:operation>
</wsdl:binding>

<wsdl:service name="FinanceServicesService">
    <wsdl:port binding="fmintf:FinanceSoapBinding" name="Finance">
        <wsdlsoap:address location="http://16.89.26.41:46300/itg/services/Finance"/>
    </wsdl:port>
</wsdl:service>
</wsdl:definitions>
```


D HP Demand Management Web Service Files

Version 7.1 WSDL File

```
<wsdl:definitions xmlns:dmMsg="http://mercury.com/ppm/dm/msg/1.0" xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/" xmlns:ds="http://mercury.com/ppm/dm/service/1.0" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://mercury.com/ppm/dm/msg/1.0">
  <wsdl:documentation>DemandService</wsdl:documentation>
  <wsdl:types>
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:dm="http://mercury.com/ppm/dm/1.0" xmlns:common="http://mercury.com/ppm/common/1.0" attributeFormDefault="qualified" elementFormDefault="qualified" targetNamespace="http://mercury.com/ppm/dm/service/1.0">
      <xs:import namespace="http://mercury.com/ppm/dm/1.0" schemaLocation="../xsd/Demand.xsd"/>
      <xs:element name="createRequest">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="requestObj" type="dm:Request"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="createRequestResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="return" nillable="true" type="dm:RemoteReference"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="getRequests">
        <xs:complexType>
          <xs:sequence>
            <xs:element maxOccurs="unbounded" name="requestIds" nillable="true" type="dm:Identifier"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="getRequestsResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element maxOccurs="unbounded" name="return" nillable="true" type="dm:Request"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="addRequestNotes">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="requestId" type="dm:Identifier"/>
            <xs:element maxOccurs="unbounded" name="notes" type="common:Note"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="addRequestNotesResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="return" nillable="true" type="xs:int"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="deleteRequests">
        <xs:complexType>
```

```

        <xs:sequence>
          <xs:element maxOccurs="unbounded" name="requestIds" type="dm:Identifier"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="deleteRequestsResponse">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="return" nillable="true" type="xs:int"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="setRequestRemoteReferenceStatus">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="receiver" type="dm:Identifier"/>
          <xs:element name="source" type="dm:Identifier"/>
          <xs:element name="status" type="xs:string"/>
          <xs:element maxOccurs="unbounded" name="fields" nillable="true"
type="dm:SimpleField"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="setRequestRemoteReferenceStatusResponse">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="return" nillable="true" type="xs:int"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="setRequestFields">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="requestId" type="dm:Identifier"/>
          <xs:element maxOccurs="unbounded" name="fields" type="dm:SimpleField"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="setRequestFieldsResponse">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="return" nillable="true" type="xs:int"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:schema>
</wsdl:types>
<wsdl:message name="createRequestMessage">
  <wsdl:part name="part1" element="ds:createRequest"/>
</wsdl:message>
<wsdl:message name="createRequestResponseMessage">
  <wsdl:part name="part1" element="ds:createRequestResponse"/>
</wsdl:message>
<wsdl:message name="getRequestsMessage">
  <wsdl:part name="part1" element="ds:getRequests"/>
</wsdl:message>
<wsdl:message name="getRequestsResponseMessage">
  <wsdl:part name="part1" element="ds:getRequestsResponse"/>
</wsdl:message>
<wsdl:message name="addRequestNotesMessage">
  <wsdl:part name="part1" element="ds:addRequestNotes"/>
</wsdl:message>
<wsdl:message name="addRequestNotesResponseMessage">
  <wsdl:part name="part1" element="ds:addRequestNotesResponse"/>
</wsdl:message>
<wsdl:message name="deleteRequestsMessage">
  <wsdl:part name="part1" element="ds:deleteRequests"/>
</wsdl:message>
<wsdl:message name="deleteRequestsResponseMessage">
  <wsdl:part name="part1" element="ds:deleteRequestsResponse"/>
</wsdl:message>
<wsdl:message name="setRequestRemoteReferenceStatusMessage">
  <wsdl:part name="part1" element="ds:setRequestRemoteReferenceStatus"/>
</wsdl:message>
<wsdl:message name="setRequestRemoteReferenceStatusResponseMessage">
  <wsdl:part name="part1" element="ds:setRequestRemoteReferenceStatusResponse"/>

```



```

</wsdl:message>
<wsdl:message name="setRequestFieldsMessage">
  <wsdl:part name="part1" element="ds:setRequestFields"/>
</wsdl:message>
<wsdl:message name="setRequestFieldsResponseMessage">
  <wsdl:part name="part1" element="ds:setRequestFieldsResponse"/>
</wsdl:message>
<wsdl:portType name="DemandServicePortType">
  <wsdl:operation name="createRequest">
    <wsdl:input message="dmMsg:createRequestMessage"/>
    <wsdl:output message="dmMsg:createRequestResponseMessage"/>
  </wsdl:operation>
  <wsdl:operation name="getRequests">
    <wsdl:input message="dmMsg:getRequestsMessage"/>
    <wsdl:output message="dmMsg:getRequestsResponseMessage"/>
  </wsdl:operation>
  <wsdl:operation name="addRequestNotes">
    <wsdl:input message="dmMsg:addRequestNotesMessage"/>
    <wsdl:output message="dmMsg:addRequestNotesResponseMessage"/>
  </wsdl:operation>
  <wsdl:operation name="deleteRequests">
    <wsdl:input message="dmMsg:deleteRequestsMessage"/>
    <wsdl:output message="dmMsg:deleteRequestsResponseMessage"/>
  </wsdl:operation>
  <wsdl:operation name="setRequestRemoteReferenceStatus">
    <wsdl:input message="dmMsg:setRequestRemoteReferenceStatusMessage"/>
    <wsdl:output message="dmMsg:setRequestRemoteReferenceStatusResponseMessage"/>
  </wsdl:operation>
  <wsdl:operation name="setRequestFields">
    <wsdl:input message="dmMsg:setRequestFieldsMessage"/>
    <wsdl:output message="dmMsg:setRequestFieldsResponseMessage"/>
  </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="DemandServiceSOAP11Binding" type="dmMsg:DemandServicePortType">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="createRequest">
    <soap:operation soapAction="urn:createRequest" style="document"/>
    <wsdl:input>
      <soap:body use="literal" namespace="http://mercury.com/ppm/dm/1.0"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" namespace="http://mercury.com/ppm/dm/1.0"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="getRequests">
    <soap:operation soapAction="urn:getRequests" style="document"/>
    <wsdl:input>
      <soap:body use="literal" namespace="http://mercury.com/ppm/dm/1.0"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" namespace="http://mercury.com/ppm/dm/1.0"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="addRequestNotes">
    <soap:operation soapAction="urn:addRequestNotes" style="document"/>
    <wsdl:input>
      <soap:body use="literal" namespace="http://mercury.com/ppm/dm/1.0"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" namespace="http://mercury.com/ppm/dm/1.0"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="deleteRequests">
    <soap:operation soapAction="urn:deleteRequests" style="document"/>
    <wsdl:input>
      <soap:body use="literal" namespace="http://mercury.com/ppm/dm/1.0"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" namespace="http://mercury.com/ppm/dm/1.0"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="setRequestRemoteReferenceStatus">
    <soap:operation soapAction="urn:setRequestRemoteReferenceStatus" style="document"/>
    <wsdl:input>
      <soap:body use="literal" namespace="http://mercury.com/ppm/dm/1.0"/>
    </wsdl:input>
  </wsdl:operation>

```

```

        <wsdl:output>
            <soap:body use="literal" namespace="http://mercury.com/ppm/dm/1.0"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="setRequestFields">
        <soap:operation soapAction="urn:setRequestFields" style="document"/>
        <wsdl:input>
            <soap:body use="literal" namespace="http://mercury.com/ppm/dm/1.0"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" namespace="http://mercury.com/ppm/dm/1.0"/>
        </wsdl:output>
    </wsdl:operation>
</wsdl:binding>
<wsdl:binding name="DemandServiceSOAP12Binding" type="dmMsg:DemandServicePortType">
    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
    <wsdl:operation name="createRequest">
        <soap12:operation soapAction="urn:createRequest" style="document"/>
        <wsdl:input>
            <soap12:body use="literal" namespace="http://mercury.com/ppm/dm/1.0"/>
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal" namespace="http://mercury.com/ppm/dm/1.0"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="getRequests">
        <soap12:operation soapAction="urn:getRequests" style="document"/>
        <wsdl:input>
            <soap12:body use="literal" namespace="http://mercury.com/ppm/dm/1.0"/>
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal" namespace="http://mercury.com/ppm/dm/1.0"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="addRequestNotes">
        <soap12:operation soapAction="urn:addRequestNotes" style="document"/>
        <wsdl:input>
            <soap12:body use="literal" namespace="http://mercury.com/ppm/dm/1.0"/>
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal" namespace="http://mercury.com/ppm/dm/1.0"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="deleteRequests">
        <soap12:operation soapAction="urn:deleteRequests" style="document"/>
        <wsdl:input>
            <soap12:body use="literal" namespace="http://mercury.com/ppm/dm/1.0"/>
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal" namespace="http://mercury.com/ppm/dm/1.0"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="setRequestRemoteReferenceStatus">
        <soap12:operation soapAction="urn:setRequestRemoteReferenceStatus" style="document"/>
        <wsdl:input>
            <soap12:body use="literal" namespace="http://mercury.com/ppm/dm/1.0"/>
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal" namespace="http://mercury.com/ppm/dm/1.0"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="setRequestFields">
        <soap12:operation soapAction="urn:setRequestFields" style="document"/>
        <wsdl:input>
            <soap12:body use="literal" namespace="http://mercury.com/ppm/dm/1.0"/>
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal" namespace="http://mercury.com/ppm/dm/1.0"/>
        </wsdl:output>
    </wsdl:operation>
</wsdl:binding>
<wsdl:service name="DemandService">
    <wsdl:port name="DemandServiceSOAP11port" binding="dmMsg:DemandServiceSOAP11Binding">
        <soap:address location="http://localhost:8080/itg/ppmservices/DemandService"/>
    </wsdl:port>
    <wsdl:port name="DemandServiceSOAP12port" binding="dmMsg:DemandServiceSOAP12Binding">

```

```

        <soap12:address location="http://localhost:8080/itg/ppmservices/DemandService"/>
    </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

Version 7.1 XSD File

```

<schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:dm="http://mercury.com/ppm/dm/1.0"
xmlns:common="http://mercury.com/ppm/common/1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://mercury.com/ppm/dm/1.0" elementFormDefault="qualified"
attributeFormDefault="qualified">
    <import namespace="http://mercury.com/ppm/common/1.0" schemaLocation="../xsd/Common.xsd"/>
    <element name="Request" type="dm:Request"/>
    <complexType name="Request">
        <sequence>
            <element name="id" type="xs:string" minOccurs="0"/>
            <element name="requestType" type="xs:string" />
            <element name="simpleFields" type="dm:SimpleField" nillable="true" maxOccurs="unbounded"/>
                <element name="tables" type="common:Table" nillable="true" maxOccurs="unbounded"/>
            <element name="notes" type="common:Note" nillable="true" maxOccurs="unbounded"/>
            <element name="URLReferences" type="dm:URLReference" nillable="true" maxOccurs="unbounded"/>
        >
            <element name="remoteReferences" type="dm:RemoteReference" nillable="true"
maxOccurs="unbounded"/>
        </sequence>
    </complexType>
    <element name="Identifier" type="dm:Identifier"/>
    <complexType name="Identifier">
        <sequence>
            <element name="id" type="xs:string" />
            <element name="serverURL" type="xs:string" minOccurs="0"/>
        </sequence>
    </complexType>
    <element name="Reference" type="dm:Reference"/>
    <complexType name="Reference">
        <sequence>
            <element name="addedBy" type="xs:string" minOccurs="0"/>
            <element name="creationDate" type="xs:dateTime" minOccurs="0"/>
            <element name="description" type="xs:string" minOccurs="0"/>
            <element name="name" type="xs:string" minOccurs="0"/>
        </sequence>
    </complexType>
    <element name="RemoteReference" type="dm:RemoteReference"/>
    <complexType name="RemoteReference">
        <complexContent>
            <extension base="dm:Reference">
                <sequence>
                    <element name="displayURL" type="xs:string" minOccurs="0"/>
                    <element name="identifier" type="dm:Identifier" nillable="true"/>
                    <element name="status" type="xs:string" minOccurs="0"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <element name="RequestRemoteReference" type="dm:RequestRemoteReference"/>
    <complexType name="RequestRemoteReference">
        <complexContent>
            <extension base="dm:RemoteReference">
                <sequence>
                    <element name="requestTypeName" type="xs:string" minOccurs="0"/>
                    <element name="relationType" type="xs:string" minOccurs="0"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <element name="URLReference" type="dm:URLReference"/>
    <complexType name="URLReference">
        <complexContent>
            <extension base="dm:Reference">
                <sequence>

```

```

        <element name="refURL" type="xs:string" />
    </sequence>
</extension>
</complexContent>
</complexType>
<element name="SimpleField" type="dm:SimpleField"/>
<complexType name="SimpleField">
    <complexContent>
        <extension base="common:Field">
            <sequence>
                <element name="stringValue" type="xs:string" maxOccurs="unbounded"/>
                <element name="dateValue" type="xs:dateTime" minOccurs="0"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<element name="FieldChangeNote" type="dm:FieldChangeNote"/>
<complexType name="FieldChangeNote">
    <complexContent>
        <extension base="common:Note">
            <sequence>
                <element name="fieldPrompt" type="xs:string"/>
                <element name="oldValue" type="xs:string"/>
                <element name="newValue" type="xs:string"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>
</schema>

```

Versions 6.0 and 7.0 WSDL File

```

<?xml version="1.0" encoding="UTF-8"?>
<wSDL:definitions targetNamespace="http://mercury.com/itg/dm/1.0" xmlns="http://
schemas.xmlsoap.org/wsdl/" xmlns:apachesoap="http://xml.apache.org/xml-soap" xmlns:intf="http://
mercury.com/itg/dm/1.0" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:types="http://mercury.com/itg/dm/1.0/types" xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/"
xmlns:wSDLsoap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:xsd="http://www.w3.org/2001/
XMLSchema">

    <wSDL:types>

        <schema targetNamespace="http://mercury.com/itg/dm/1.0/types" xmlns="http://www.w3.org/
2001/XMLSchema">

            <import namespace="http://schemas.xmlsoap.org/soap/encoding/" />

            <complexType abstract="true" name="Value">
                <sequence />
            </complexType>

            <complexType name="RequestHeader">
                <sequence>
                    <element name="auditNote" nillable="false" type="xsd:string"/>
                    <element name="origin" nillable="false" type="xsd:string"/>
                    <element name="transactionName" nillable="false" type="xsd:string"/>
                    <element name="credentials" nillable="false" type="types:Credentials"/>
                </sequence>
            </complexType>

            <complexType abstract="true" name="Credentials">

```

```

    <sequence/>
</complexType>
<complexType name="LoginAccount">
    <complexContent>
        <extension base="types:Credentials">
            <sequence>
                <element name="username" nillable="false" type="xsd:string"/>
                <element name="password" nillable="false" type="xsd:string"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<complexType name="ProcessingError">
    <sequence>
        <element name="errors" nillable="false" type="types:ArrayOf_types_ErrorMessage"/>
        <element name="message" nillable="false" type="xsd:string"/>
    </sequence>
</complexType>
<complexType name="Field">
    <sequence>
        <element name="token" nillable="false" type="xsd:string"/>
        <element name="value" nillable="false" type="types:Value"/>
    </sequence>
</complexType>
<complexType abstract="true" name="Reference">
    <sequence>
        <element name="creationDate" nillable="true" type="xsd:dateTime"/>
        <element name="name" nillable="true" type="xsd:string"/>
        <element name="description" nillable="true" type="xsd:string"/>
        <element name="addedBy" nillable="true" type="xsd:string"/>
    </sequence>
</complexType>
<complexType abstract="true" name="Note">
    <sequence>
        <element name="creationDate" nillable="true" type="xsd:dateTime"/>
        <element name="content" nillable="false" type="xsd:string"/>
        <element name="addedBy" nillable="true" type="xsd:string"/>
    </sequence>

```

```

</complexType>
<complexType name="Request">
  <sequence>
    <element name="fields" nillable="false" type="types:ArrayOf_types_Field"/>
    <element name="references" nillable="true" type="types:ArrayOf_types_Reference"/>
    <element name="notes" nillable="true" type="types:ArrayOf_types_Note"/>
  </sequence>
</complexType>
<complexType name="ErrorMessage">
  <sequence>
    <element name="code" nillable="true" type="xsd:int"/>
    <element name="message" nillable="true" type="xsd:string"/>
  </sequence>
</complexType>
<complexType name="Response">
  <sequence>
    <element name="references" nillable="true" type="types:ArrayOf_types_Reference"/>
  </sequence>
</complexType>
<complexType name="Identifier">
  <sequence>
    <element name="serverURL" nillable="false" type="xsd:anyURI"/>
    <element name="id" nillable="false" type="xsd:string"/>
  </sequence>
</complexType>
<complexType name="FieldMetaData">
  <sequence>
    <element name="prompt" nillable="false" type="xsd:string"/>
    <element name="token" nillable="false" type="xsd:string"/>
    <element name="required" nillable="false" type="xsd:boolean"/>
    <element name="description" nillable="false" type="xsd:string"/>
    <element name="fieldType" nillable="false" type="xsd:string"/>
    <element name="listName" nillable="true" type="xsd:string"/>
  </sequence>
</complexType>
<complexType name="SingleValue">
  <complexContent>
    <extension base="types:Value">

```

```

        <sequence>
            <element name="tokenValue" nillable="true" type="xsd:string"/>
        </sequence>
    </extension>
</complexContent>
</complexType>
<complexType name="MultipleValues">
    <complexContent>
        <extension base="types:Value">
            <sequence>
                <element name="values" nillable="true" type="types:ArrayOf_xsd_string"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<complexType name="DateValue">
    <complexContent>
        <extension base="types:Value">
            <sequence>
                <element name="date" nillable="true" type="xsd:dateTime"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<complexType name="Table">
    <complexContent>
        <extension base="types:Value">
            <sequence>
                <element name="tokens" nillable="false" type="types:ArrayOf_xsd_string"/>
                <element name="rows" nillable="false" type="types:ArrayOf_types_TableRow"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<complexType name="URLReference">
    <complexContent>
        <extension base="types:Reference">
            <sequence>

```

```

        <element name="URL" nillable="false" type="xsd:anyURI"/>
    </sequence>
</extension>
</complexContent>
</complexType>
<complexType name="RemoteReference">
    <complexContent>
        <extension base="types:Reference">
            <sequence>
                <element name="status" nillable="false" type="xsd:string"/>
                <element name="displayURL" nillable="true" type="xsd:anyURI"/>
                <element name="identifier" nillable="false" type="types:Identifier"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<complexType name="UserNote">
    <complexContent>
        <extension base="types:Note">
            <sequence/>
        </extension>
    </complexContent>
</complexType>
<complexType name="FieldChangeNote">
    <complexContent>
        <extension base="types:Note">
            <sequence>
                <element name="fieldPrompt" nillable="false" type="xsd:string"/>
                <element name="oldValue" nillable="false" type="xsd:string"/>
                <element name="newValue" nillable="false" type="xsd:string"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<complexType name="ArrayOf_types_Field">
    <complexContent>
        <restriction base="soapenc:Array">
            <attribute ref="soapenc:arrayType" wsdl:arrayType="types:Field[]"/>
        </restriction>
    </complexContent>
</complexType>

```



```

        </restriction>
    </complexContent>
</complexType>
<complexType name="TableRow">
    <complexContent>
        <restriction base="soapenc:Array">
            <attribute ref="soapenc:arrayType" wsdl:arrayType="types:Value[]" />
        </restriction>
    </complexContent>
</complexType>
<complexType name="ArrayOf_types_TableRow">
    <complexContent>
        <restriction base="soapenc:Array">
            <attribute ref="soapenc:arrayType" wsdl:arrayType="types:TableRow[]" />
        </restriction>
    </complexContent>
</complexType>
<complexType name="ArrayOf_xsd_string">
    <complexContent>
        <restriction base="soapenc:Array">
            <attribute ref="soapenc:arrayType" wsdl:arrayType="xsd:string[]" />
        </restriction>
    </complexContent>
</complexType>
<complexType name="ArrayOf_types_Reference">
    <complexContent>
        <restriction base="soapenc:Array">
            <attribute ref="soapenc:arrayType" wsdl:arrayType="types:Reference[]" />
        </restriction>
    </complexContent>
</complexType>
<complexType name="ArrayOf_types_Note">
    <complexContent>
        <restriction base="soapenc:Array">
            <attribute ref="soapenc:arrayType" wsdl:arrayType="types:Note[]" />
        </restriction>
    </complexContent>

```

```

</complexType>
<complexType name="ArrayOf_types_FieldMetaData">
  <complexContent>
    <restriction base="soapenc:Array">
      <attribute ref="soapenc:arrayType" wsdl:arrayType="types:FieldMetaData[]" />
    </restriction>
  </complexContent>
</complexType>
<complexType name="ArrayOf_types_ErrorMessage">
  <complexContent>
    <restriction base="soapenc:Array">
      <attribute ref="soapenc:arrayType" wsdl:arrayType="types:ErrorMessage[]" />
    </restriction>
  </complexContent>
</complexType>

<complexType name="mapItem">
  <sequence>
    <element name="key" nillable="true" type="xsd:anyType" />
    <element name="value" nillable="true" type="xsd:anyType" />
  </sequence>
</complexType>

<complexType name="Map">
  <sequence>
    <element maxOccurs="unbounded" minOccurs="0" name="item" type="types:mapItem" />
  </sequence>
</complexType>
</schema>
</wsdl:types>

<wsdl:message name="importRequestRequest">
  <wsdl:part name="header" type="types:RequestHeader" />
  <wsdl:part name="request" type="types:Request" />
</wsdl:message>

<wsdl:message name="importRequestResponse">
  <wsdl:part name="importRequestReturn" type="types:Response" />
</wsdl:message>

<wsdl:message name="updateRemoteReferenceResponse">

```

```

        <wsdl:part name="updateRemoteReferenceReturn" type="xsd:int"/>
</wsdl:message>

<wsdl:message name="ProcessingErrorFault">
    <wsdl:part name="fault" type="types:ProcessingError"/>
</wsdl:message>

<wsdl:message name="updateRemoteReferenceRequest">
    <wsdl:part name="header" type="types:RequestHeader"/>
    <wsdl:part name="receiver" type="types:Identifier"/>
    <wsdl:part name="source" type="types:Identifier"/>
    <wsdl:part name="newStatus" type="xsd:string"/>
    <wsdl:part name="fields" type="types:ArrayOf_types_Field"/>
</wsdl:message>

<wsdl:message name="getRequestTypesByFieldGroupRequest">
    <wsdl:part name="header" type="types:RequestHeader"/>
    <wsdl:part name="fieldGroup" type="xsd:string"/>
</wsdl:message>

<wsdl:message name="getRequestTypesByFieldGroupResponse">
    <wsdl:part name="getRequestTypesByFieldGroupReturn" type="types:ArrayOf_xsd_string"/>
</wsdl:message>

<wsdl:message name="getRequestTypeFieldsRequest">
    <wsdl:part name="header" type="types:RequestHeader"/>
    <wsdl:part name="requestType" type="xsd:string"/>
</wsdl:message>

<wsdl:message name="getRequestTypeFieldsResponse">
    <wsdl:part name="getRequestTypeFieldsReturn" type="types:ArrayOf_types_FieldMetaData"/>
</wsdl:message>

<wsdl:message name="deleteRequestsRequest">
    <wsdl:part name="header" type="types:RequestHeader"/>
    <wsdl:part name="requestIDs" type="types:ArrayOf_xsd_string"/>
</wsdl:message>

<wsdl:message name="deleteRequestsResponse">
    <wsdl:part name="deleteRequestsReturn" type="xsd:int"/>
</wsdl:message>

```

```

<wsdl:portType name="RequestServices">
  <wsdl:operation name="importRequest" parameterOrder="header request">
    <wsdl:input message="intf:importRequestRequest" name="importRequestRequest"/>
    <wsdl:output message="intf:importRequestResponse" name="importRequestResponse"/>
    <wsdl:fault message="intf:ProcessingErrorFault" name="ProcessingErrorFault"/>
  </wsdl:operation>

  <wsdl:operation name="updateRemoteReference" parameterOrder="header receiver source
newStatus fields">
    <wsdl:input message="intf:updateRemoteReferenceRequest"
name="updateRemoteReferenceRequest"/>
    <wsdl:output message="intf:updateRemoteReferenceResponse"
name="updateRemoteReferenceResponse"/>
    <wsdl:fault message="intf:ProcessingErrorFault" name="ProcessingErrorFault"/>
  </wsdl:operation>

  <wsdl:operation name="getRequestTypesByFieldGroup" parameterOrder="header fieldGroup">
    <wsdl:input message="intf:getRequestTypesByFieldGroupRequest"
name="getRequestTypesByFieldGroupRequest"/>
    <wsdl:output message="intf:getRequestTypesByFieldGroupResponse"
name="getRequestTypesByFieldGroupResponse"/>
    <wsdl:fault message="intf:ProcessingErrorFault" name="ProcessingErrorFault"/>
  </wsdl:operation>

  <wsdl:operation name="getRequestTypeFields" parameterOrder="header requestType">
    <wsdl:input message="intf:getRequestTypeFieldsRequest"
name="getRequestTypeFieldsRequest"/>
    <wsdl:output message="intf:getRequestTypeFieldsResponse"
name="getRequestTypeFieldsResponse"/>
    <wsdl:fault message="intf:ProcessingErrorFault" name="ProcessingErrorFault"/>
  </wsdl:operation>

  <wsdl:operation name="deleteRequests" parameterOrder="header requestIDs">
    <wsdl:input message="intf:deleteRequestsRequest" name="deleteRequestsRequest"/>
    <wsdl:output message="intf:deleteRequestsResponse" name="deleteRequestsResponse"/>
    <wsdl:fault message="intf:ProcessingErrorFault" name="ProcessingErrorFault"/>
  </wsdl:operation>

</wsdl:portType>

<wsdl:binding name="RequestSoapBinding" type="intf:RequestServices">
  <wsdlsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="importRequest">
    <wsdlsoap:operation soapAction=""/>

```

```

        <wsdl:input name="importRequestRequest">
            <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://mercury.com/itg/dm/1.0" use="encoded"/>
        </wsdl:input>
        <wsdl:output name="importRequestResponse">
            <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://mercury.com/itg/dm/1.0" use="encoded"/>
        </wsdl:output>
        <wsdl:fault name="ProcessingErrorFault">
            <wsdlsoap:fault name="ProcessingErrorFault" use="encoded"/>
        </wsdl:fault>
    </wsdl:operation>

    <wsdl:operation name="updateRemoteReference">
        <wsdlsoap:operation soapAction=""/>
        <wsdl:input name="updateRemoteReferenceRequest">
            <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://mercury.com/itg/dm/1.0" use="encoded"/>
        </wsdl:input>
        <wsdl:output name="updateRemoteReferenceResponse">
            <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://mercury.com/itg/dm/1.0" use="encoded"/>
        </wsdl:output>
        <wsdl:fault name="ProcessingErrorFault">
            <wsdlsoap:fault name="ProcessingErrorFault" use="encoded"/>
        </wsdl:fault>
    </wsdl:operation>

    <wsdl:operation name="getRequestTypesByFieldGroup">
        <wsdlsoap:operation soapAction=""/>
        <wsdl:input name="getRequestTypesByFieldGroupRequest">
            <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://mercury.com/itg/dm/1.0" use="encoded"/>
        </wsdl:input>
        <wsdl:output name="getRequestTypesByFieldGroupResponse">
            <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://mercury.com/itg/dm/1.0" use="encoded"/>
        </wsdl:output>
        <wsdl:fault name="ProcessingErrorFault">
            <wsdlsoap:fault name="ProcessingErrorFault" use="encoded"/>
        </wsdl:fault>
    </wsdl:operation>

```

```

    <wsdl:operation name="getRequestTypeFields">
      <wsdlsoap:operation soapAction=""/>
      <wsdl:input name="getRequestTypeFieldsRequest">
        <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
          namespace="http://mercury.com/itg/dm/1.0" use="encoded"/>
      </wsdl:input>
      <wsdl:output name="getRequestTypeFieldsResponse">
        <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
          namespace="http://mercury.com/itg/dm/1.0" use="encoded"/>
      </wsdl:output>
      <wsdl:fault name="ProcessingErrorFault">
        <wsdlsoap:fault name="ProcessingErrorFault" use="encoded"/>
      </wsdl:fault>
    </wsdl:operation>

    <wsdl:operation name="deleteRequests">
      <wsdlsoap:operation soapAction=""/>
      <wsdl:input name="deleteRequestsRequest">
        <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
          namespace="http://mercury.com/itg/dm/1.0" use="encoded"/>
      </wsdl:input>
      <wsdl:output name="deleteRequestsResponse">
        <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
          namespace="http://mercury.com/itg/dm/1.0" use="encoded"/>
      </wsdl:output>
      <wsdl:fault name="ProcessingErrorFault">
        <wsdlsoap:fault name="ProcessingErrorFault" use="encoded"/>
      </wsdl:fault>
    </wsdl:operation>

  </wsdl:binding>

  <wsdl:service name="RequestServicesService">
    <wsdl:port binding="intf:RequestSoapBinding" name="Request">
      <wsdlsoap:address location="http://16.89.26.41:46300/itg/services/Request"/>
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>

```

E HP Time Management Web Service Files

Version 7.1 WSDL File

```
<wsdl:definitions xmlns:tmMsg="http://mercury.com/ppm/tm/msg/1.0" xmlns:soap12="http://
schemas.xmlsoap.org/wsdl/soap12/" xmlns:ts="http://mercury.com/ppm/tm/service/1.0"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
targetNamespace="http://mercury.com/ppm/tm/msg/1.0">
  <wsdl:types>
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:tm="http://mercury.com/ppm/tm/
1.0" attributeFormDefault="qualified" elementFormDefault="qualified" targetNamespace="http://
mercury.com/ppm/tm/service/1.0">
      <xs:import namespace="http://mercury.com/ppm/tm/1.0" schemaLocation="../xsd/Time.xsd"/>
      <xs:element name="getActualTime">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="param0" nillable="true" type="tm:TimeFilter"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="getActualTimeResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element maxOccurs="unbounded" minOccurs="0" name="return" nillable="true"
type="tm:WorkItemActualTime"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:schema>
  </wsdl:types>
  <wsdl:message name="getActualTimeMessage">
    <wsdl:part name="part1" element="ts:getActualTime"/>
  </wsdl:message>
  <wsdl:message name="getActualTimeResponseMessage">
    <wsdl:part name="part1" element="ts:getActualTimeResponse"/>
  </wsdl:message>
  <wsdl:portType name="TimeServicePortType">
    <wsdl:operation name="getActualTime">
      <wsdl:input message="tmMsg:getActualTimeMessage"/>
      <wsdl:output message="tmMsg:getActualTimeResponseMessage"/>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="TimeServiceSOAP11Binding" type="tmMsg:TimeServicePortType">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="getActualTime">
      <soap:operation soapAction="urn:getActualTime" style="document"/>
      <wsdl:input>
        <soap:body use="literal" namespace="http://mercury.com/ppm/tm/msg/1.0"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" namespace="http://mercury.com/ppm/tm/msg/1.0"/>
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:binding name="TimeServiceSOAP12Binding" type="tmMsg:TimeServicePortType">
    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
    <wsdl:operation name="getActualTime">
      <soap12:operation soapAction="urn:getActualTime" style="document"/>
      <wsdl:input>
        <soap12:body use="literal" namespace="http://mercury.com/ppm/tm/msg/1.0"/>
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" namespace="http://mercury.com/ppm/tm/msg/1.0"/>
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>

```

```

    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="TimeService">
    <wsdl:port name="TimeServiceSOAP11port" binding="tmMsg:TimeServiceSOAP11Binding">
      <soap:address location="http://localhost:8080/tmMsg/services/TimeService"/>
    </wsdl:port>
    <wsdl:port name="TimeServiceSOAP12port" binding="tmMsg:TimeServiceSOAP12Binding">
      <soap12:address location="http://localhost:8080/tmMsg/services/TimeService"/>
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>

```

Version 7.1 XSD File

```

<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:tm="http://mercury.com/ppm/tm/1.0"
xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="http://mercury.com/ppm/tm/1.0"
elementFormDefault="qualified" attributeFormDefault="qualified">
  <element name="TimeFilter" type="tm:TimeFilter"/>
  <complexType name="TimeFilter">
    <sequence>
      <element name="startDate" type="xs:dateTime" maxOccurs="1" minOccurs="1"/>
      <element name="endDate" type="xs:dateTime" maxOccurs="1" minOccurs="1"/>
      <element name="periodTypeName" type="xs:string" maxOccurs="1" minOccurs="1"/>
      <element name="workItemType" type="xs:string" maxOccurs="1" minOccurs="0"/>
      <element name="workItemName" type="xs:string" maxOccurs="1" minOccurs="0"/>
      <element name="workItemID" type="xs:string" maxOccurs="1" minOccurs="0"/>
      <element name="resources" type="xs:string" maxOccurs="255" minOccurs="0"/>
      <element name="timeSheetStatuses" type="xs:string" maxOccurs="unbounded" minOccurs="0"/>
      <element name="timeSheetLineStatuses" type="xs:string" maxOccurs="unbounded" minOccurs="0"/>
    </sequence>
  </complexType>
  <complexType name="WorkItemActualTime">
    <sequence>
      <element name="includeActualCost" type="xs:boolean" maxOccurs="1" minOccurs="0"/>
    </sequence>
  </complexType>
  <complexType name="WorkItemActualTime">
    <sequence>
      <element name="startDate" type="xs:dateTime"/>
      <element name="endDate" type="xs:dateTime"/>
      <element name="workItemName" type="xs:string"/>
      <element name="workItemTypeCode" type="xs:string"/>
      <element name="workItemID" type="xs:string"/>
      <element name="activityName" type="xs:string"/>
      <element name="chargeCode" type="xs:string"/>
      <element name="hours" type="xs:string"/>
      <element name="resource" type="xs:string"/>
      <element name="timeSheetLineStatus" type="xs:string"/>
      <element name="cost" type="xs:string"/>
    </sequence>
  </complexType>
</schema>

```


F Shared Web Service Files

This appendix lists the XSD file that is common to the following application modules. This XSD file is used in addition to the one listed in their respective appendix.

- HP Project Management
- HP Demand Management
- HP Resource Management

Version 7.1 Shared XSD File

```
<xs:schema attributeFormDefault="qualified" elementFormDefault="qualified"
targetNamespace="http://mercury.com/ppm/common/1.0" xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:common="http://mercury.com/ppm/common/1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="AuditHeader" type="common:AuditHeader"/>
<xs:complexType name="AuditHeader">
<xs:sequence>
<xs:element minOccurs="0" name="audit-note" type="string"/>
<xs:element minOccurs="0" name="audit-origin" type="string"/>
<xs:element minOccurs="0" name="audit-transaction-name" type="string"/>
</xs:sequence>
</xs:complexType>
<xs:element name="Note" type="common:Note"/>
<xs:complexType name="Note">
<xs:sequence>
<xs:element minOccurs="0" name="author" type="string"/>
<xs:element minOccurs="0" name="creationDate" type="dateTime"/>
<xs:element minOccurs="0" name="content" type="string"/>
</xs:sequence>
</xs:complexType>
<xs:element name="Field" type="common:Field"/>
<xs:complexType name="Field">
<xs:sequence>
<xs:element name="token" type="string"/>
</xs:sequence>
</xs:complexType>
<xs:element name="Table" type="common:Table"/>
<xs:complexType name="Table">
<xs:complexContent>
<xs:extension base="common:Field">
<xs:sequence>
<xs:element maxOccurs="unbounded" name="columns" nillable="true" type="common:Column"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:element name="Column" type="common:Column"/>
<xs:complexType name="Column">
<xs:complexContent>
<xs:extension base="common:Field">
<xs:sequence>
<xs:element maxOccurs="unbounded" name="values" nillable="true" type="string"/>
<xs:element maxOccurs="unbounded" name="dates" nillable="true" type="dateTime"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
```

```
<xs:element name="FieldMetaData" type="common:FieldMetaData"/>
<xs:complexType name="FieldMetaData">
  <xs:sequence>
    <xs:element minOccurs="0" name="description" type="string"/>
    <xs:element minOccurs="0" name="prompt" type="string"/>
    <xs:element name="required" type="boolean"/>
    <xs:element minOccurs="0" name="token" type="string"/>
  </xs:sequence>
</xs:complexType>
</xs:schema>
```

G Outdated Web Services

In This Chapter:

- *Overview of the Outdated Web Services*
 - *HP Financial Management Deprecated Operations*
 - *create*
 - *read*
 - *update*
 - *HP Demand Management Deprecated Operations*
 - *getRequestTypesByFieldGroup*
 - *getRequestTypeFields*
 - *HP Demand Management Re-implemented Operations*
 - *HP Demand Management Re-Implemented Special Commands*
-

Overview of the Outdated Web Services

Many of the version 6.0 and 7.0 Web services have been re-implemented to take advantage of Axis2 security features. Other version 6.0 and 7.0 Web services have been deprecated since they are minimally used by PPM Center customers, or have had their functionality incorporated into version 7.1 Web services.

To ensure backward compatibility for sites already using this functionality, these Web services have been retained in version 7.1.



These Web services have been retained for interim purposes only and, eventually, will be deleted.

The version 6.0 and 7.0 WSDL files are included with version 7.1 as indicated in [Chapter 4, *Developing Web Service Applications*, on page 57](#). Refer to the appropriate application module appendix for the WSDL file content.

For additional details about the Mercury IT Governance Center version 6.0 or PPM Center version 7.0 Web services, see the version 6.0 *Service-Oriented Architecture: Web Services Guide*.

HP Financial Management Deprecated Operations

The following Web service operations have been deprecated; however, HP plans to re-implement these in a future release.

create

Description

`create` allows a remote system to call PPM Center and create a new budget. If a budget with the specified name already exists, an error is generated.

A `CreateResponse` is returned and contains the header, a response message, and the budget ID.

This function allows the creation of a single budget. You can specify almost all of the data fields of a budget in the input object of this function.

Java Syntax

```
CreateResponse create (CreateMessage <createMessage>)
```

Parameters

Name	Description
createMessage	The header provides the Request Header object with a unique transaction name, as well as username and password to access PPM Center. The budget data specifies the Budget object. (This is the same as the object returned by the <code>read</code> Web service.)

Java Objects

Field	Description	Java Method
Budget ID	(Optional) Set this field to <code>NULL</code> .	<code>Budget.setBudgetID()</code>
Budget Name	When creating a new budget, specify a unique name.	<code>Budget.setBudgetName()</code>
Budget Status	Specify a valid budget status.	<code>Budget.setBudgetStatus()</code>
CAPEX/OPEX Flag	(Optional) Set to <code>TRUE</code> to enable CAPEX/OPEX.	<code>Budget.setCapexOpexEnabledFlag()</code>

Field	Description	Java Method
Active Flag	(Optional) Specify a valid active flag.	BudgetFilters.setActive()
Actuals Rollup Code	(Optional) Specify an actuals rollup code. Valid values are: <ul style="list-style-type: none"> ■ MANUAL ■ AUTO_LABOR_ AUTO_NONLABOR ■ AUTO_LABOR_ MANUAL_ NONLABOR 	Budget.actualsRolledUpCode()
Associated with Type	(Optional) Specify an associated entity type. Valid values are: <ul style="list-style-type: none"> ■ ORGANIZATION UNIT ■ PROGRAM ■ PROJECT, PROPOSAL ■ ASSET 	Budget.setAssociatedWithType()
Associated with Name	(Optional) Specify the name of the associated entity.	Budget.setAssociatedWithName()
Start Period Name	(Optionally required) Specify a start date. Do not specify if a Budget Start Date is provided.	Budget.setStartPeriodName()
Budget Start Date	(Optionally required) Specify a start date. Do not specify if a Start Period Name is provided.	Budget.setStartPeriodStartDate()
End Period Name	(Optionally required) Specify an end date. Do not specify if a Budget End Date is provided.	Budget.setEndPeriodName()

Field	Description	Java Method
Budget End Date	(Optionally required) Specify an end date. Do not specify if a End Period Name is provided.	Budget.setEndPeriodStartDate()
Period Type	Specify FISCAL MONTH.	Budget.setPeriodType()
Description	(Optional) Specify a description of the budget.	Budget.setDescription()
Region Name	Specify a valid region name.	Budget.setRegionName()
Currency	(Optional) Specify a base currency for the budget.	Budget.setBaseCurrency()
Budget User Data	(Optional) Specify budget user data.	Budget.setUserData()
Budget lines	(Optional) Specify multiple budget lines.	Budget.setBudgetLines()
Budget line ID	(Optional) Set this field to NULL.	BudgetLine.setBudgetLineID()
Line Type Code	(Optional) Specify the line type code. The default value is LABOR.	BudgetLine.setTypeCode()
Line Category	(Optional) Specify the budget line category. The line type code and the line type category must match. If they do not, the line type is not displayed in PPM Center and your budget-based reporting could be corrupted. For example, a line type of NON_LABOR with a line category of EMPLOYEE do <i>not</i> match.	BudgetLine.setCategoryCode()
Line Expense Type	(Optional) Specify the line expense type. The default is OPERATING.	BudgetLine.setExpenseTypeCode()

Field	Description	Java Method
Budget Line User Data	(Optional) Specify the budget line user data.	BudgetLine.setUserData()
Budget Line Cells	(Optional) Specify the budget line details.	BudgetLine.setBudgetLineDetails()
Cell Period Name	(Optional) Specify the name of the cell period. This can also be specified using Cell Period Start Date.	BudgetLineDetail.setPeriodName()
Cell Period Start Date	(Optional) Specify the cell period start date. This can also be specified using Cell Period Name.	BudgetLineDetail.setPeriodStartDate() ()
Cell Plan Value	(Optional) Specify the cell plan value.	BudgetLineDetail.setPlanValue()
Cell Actual Value	(Optional) Specify the cell actual value.	BudgetLineDetail.setActualValue()

read

Description

Read allows a remote system to call PPM Center and retrieve a list of budgets satisfying the filter criteria.

A `ReadResponse` is returned and contains the header, a response message, and an array of budget objects.

This function allows the client to query the system for a list of budget. It accepts an input of filter object, which describes the filter conditions for the query. When more than one filter condition is specified, they are applied to the search results using the logical AND.

Java Syntax

```
ReadResponse read (ReadMessage <readMessage>)
```

Parameters

Name	Description
readMessage	<p>The header provides the Request Header object with a unique transaction name, as well as username and password to access PPM Center.</p> <p>The budget filter specifies the BudgetFilters object. There are no required filters. Budget statuses are an array. Budget user data are token-value pairs. For details on token-value pairs, see <i>Commands, Tokens, and Validations Guide and Reference</i>.</p> <p>Valid filters include:</p> <ul style="list-style-type: none">■ Budget name■ Budget ID■ Budget status■ Active■ Associated with type■ Associated with name■ Budget user data

Filter Criteria

Field	Description	Java Method
Budget ID	Return budget with the specified budget ID. This limits the result to no more than one budget.	BudgetFilters.setBudgetID()
Budget Name	Return budget with the specified budget name. This limits the result to no more than one budget since budget name is unique in the system.	BudgetFilters.setBudgetName()
Budget Status	Return budget with specified budget statuses. Multiple statuses can be specified.	BudgetFilters.setBudgetStatuses()
Active Flag	Return budget with specified active flag.	BudgetFilters.setActive()
Associated with Type	Return budget(s) associated with the specified type of entity.	BudgetFilters.setAssociatedWithType()
Associated with Name	Return budget associated with the specified entity.	BudgetFilters.setAssociatedWithName()
Budget User Data	Return budget with specified user data values. Multiple user data fields can be specified, in which case, the logical AND applies.	BudgetFilters.setBudgetUserData()

update

Description

`Update` allows a remote system to call PPM Center and modify an existing budget. If a budget with the specified name does not exist, an error is generated.

An `UpdateMessage` is returned and contains the header, a response message, and the budget ID.

This function allows update of the existing budget. It allows update to the budget fields, the fields of a budget line, or a specific budget line cell.

- For update a budget, a valid budget ID has to be provided.
- For update a budget line, the budget line ID has to be provided. If it is not present, the Web service adds the line as a new budget line.
- For update a budget line cell, the period name for the cell has to be provided. The values are overwritten only if a not-null value is provided. For example, if you specify zero as plan value and leave the actual value as 'null', the plan value is overwritten as zero, and the actual value is not changed.

Java Syntax

```
UpdateMessage update (UpdateMessage <updateMessage>)
```

Parameters

Name	Description
updateMessage	<p>The header provides the Request Header object with a unique transaction name, as well as username and password to access PPM Center.</p> <p>The budget data specifies the Budget object. (This is the same as the object returned by the <code>read</code> Web service.)</p> <p>The following budget fields cannot be updated:</p> <ul style="list-style-type: none"> ■ Budget name ■ Budget ID ■ Associated with type ■ Associated with name ■ Start period name ■ Start period start date ■ End period name ■ End period start date ■ Period type ■ Region ■ Base currency

Java Objects

Field	Description	Java Method
Budget ID	Specify the budget ID.	<code>Budget.setBudgetID()</code>
Budget Name	(Optional) Specify a unique name.	<code>Budget.setBudgetName()</code>
Budget Status	(Optional) Specify a valid budget status.	<code>Budget.setBudgetStatus()</code>
CAPEX/OPEX Flag	(Optional) Set to <code>TRUE</code> to enable CAPEX/OPEX.	<code>Budget.setCapexOpexEnabledFlag()</code>
Active Flag	(Optional) Specify a valid active flag.	<code>BudgetFilters.setActive()</code>

Field	Description	Java Method
Actuals Rollup Code	(Optional) Specify an actuals rollup code. Valid values are: <ul style="list-style-type: none"> ■ MANUAL ■ AUTO_LABOR_ AUTO_NONLABOR ■ AUTO_LABOR_ MANUAL_ NONLABOR 	Budget.actualsRolledUpCode()
Associated with Type	(Optional) Specify an associated entity type. Valid values are: <ul style="list-style-type: none"> ■ ORGANIZATION UNIT ■ PROGRAM ■ PROJECT, PROPOSAL ■ ASSET 	Budget.setAssociatedWithType()
Associated with Name	(Optional) Specify the name of the associated entity.	Budget.setAssociatedWithName()
Start Period Name	(Optionally required) Specify a start date. Do not specify if a Budget Start Date is provided.	Budget.setStartPeriodName()
Budget Start Date	(Optionally required) Specify a start date. Do not specify if a Start Period Name is provided.	Budget.setStartPeriodStartDate()
End Period Name	(Optionally required) Specify an end date. Do not specify if a Budget End Date is provided.	Budget.setEndPeriodName()
Budget End Date	(Optionally required) Specify an end date. Do not specify if a End Period Name is provided.	Budget.setEndPeriodStartDate()
Period Type	Specify FISCAL MONTH.	Budget.setPeriodType()

Field	Description	Java Method
Description	Specify a description of the budget.	Budget.setDescription()
Region Name	Specify a valid region name.	Budget.setRegionName()
Currency	(Optional) Specify a base currency for the budget.	Budget.setBaseCurrency()
Budget User Data	(Optional) Specify budget user data.	Budget.setUserData()
Budget lines	(Optional) Specify multiple budget lines.	Budget.setBudgetLines()
Budget line ID	If the ID is set to <code>NULL</code> , a new budget line is added. Otherwise, the specified budget line is updated.	BudgetLine.setBudgetLineID()
Line Type Code	(Optional) Specify the line type code. The default value is <code>LABOR</code> .	BudgetLine.setTypeCode()
Line Category	(Optional) Specify the budget line category. The line type code and the line type category must match. If they do not, the line type is not displayed in PPM Center and your budget-based reporting could be corrupted. For example, a line type of <code>NON_LABOR</code> with a line category of <code>EMPLOYEE</code> do <i>not</i> match.	BudgetLine.setCategoryCode()
Line Expense Type	If not specified, it defaults to <code>OPERATING</code>	BudgetLine.setExpenseTypeCode()
Budget Line User Data	(Optional) Specify the line expense type. The default is <code>OPERATING</code> .	BudgetLine.setUserData()

Field	Description	Java Method
Budget Line Cells	(Optional) Specify the budget line user data.	BudgetLine.setBudgetLineDetails()
Cell Period Name	(Optional) Specify the name of the cell period. This can also be specified using Cell Period Start Date.	BudgetLineDetail.setPeriodName()
Cell Period Start Date	(Optional) Specify the cell period start date. This can also be specified using Cell Period Name.	BudgetLineDetail.setPeriodStartDate() ()
Cell Plan Value	(Optional) Specify the cell plan value. If set to <code>NULL</code> , the existing value is kept.	BudgetLineDetail.setPlanValue()
Cell Actual Value	(Optional) Specify the cell actual value. If set to <code>NULL</code> , the existing value is kept.	BudgetLineDetail.setActualValue()

HP Demand Management Deprecated Operations

The following Web service operations have been deprecated. HP recommends that you discontinue use of the Web services at your earliest convenience. Also, you should not design long-term solutions that are reliant on these Web services.

getRequestTypesByFieldGroup

Description

getRequestTypesByFieldGroup allows a remote system to call PPM Center and retrieve a list of request types that contain the specified field group.

Java Syntax

```
String[] getRequestTypesByFieldGroup (RequestHeader <header>,
String <fieldGroup>)
```

Parameters

Name	Description
header	Provides the Request Header object with a unique transaction name, as well as the username and password to access PPM Center.
fieldGroup	Specifies the field group to be used as a filter for the request types. If no field group is specified, all request types on the PPM Center instance are returned.

getRequestTypeFields

Description

getRequestTypeFields allows a remote system to call PPM Center and retrieve the description of all the fields in the specified request type.

An array of FieldMetaData objects are returned, each describing a single field in the specified request type.

Java Syntax

```
FieldMetaData[] getRequestTypeFields (RequestHeader <header>,
String <requestType>)
```

Parameters

Name	Description
header	Provides the Request Header object with a unique transaction name, as well as the username and password to access PPM Center.
requestType	Specifies a request type name (must be valid). An error is generated if no request type is provided.

HP Demand Management Re-implemented Operations

The following Web service operations have been re-implemented. HP strongly recommends that you convert to the new Axis2 implementation as soon as practical. If you are creating new Web service applications, use the version 7.1 Web services.

- deleteRequests
- importRequest
- updateRemoteReference

HP Demand Management Re-Implemented Special Commands

The following special command is re-implemented. HP strongly recommends that you convert to the new Axis2 implementation as soon as practical. If you are creating new Web service applications, use the version 7.1 Web services.

- `ksc_export_request`

Index

Symbols

.NET

sample Web services 87

A

addRequestNotes 19, 44

addTaskToExistingWorkPlan 17, 39

application modules

having Web services 38, 59

authentication

HTTP 33

HTTP, basic 24, 82, 84

HTTPS 35, 36

Web services, for 24

WS-Security 24, 84

authorization

Web services 24

C

client, Web services

compiling code 86

configuring 25, 29, 31, 32, 84

executing code 87

sample code 80

common XSD 145

configuration file, Web services

axis2.xml 27, 29

client_axis2.xml 28, 29

server.conf 25

webservices.conf 30

configuring

environment variables 79

Java CLASSPATH 79

Web services 25

Web services security 25, 84

create 18, 41, 149

createBlankWorkPlan 17, 39

createCostRules 18, 42

createProject 17, 39

createRequest 19, 44, 81

createResourcePools 18, 40

createWorkPlanFromTemplate 17, 39

D

debugging

Web services 26

deleteCostRules 18, 42

deleted special commands

Financial Management 43

deleteRequests 19, 43, 44

Demand Management

.NET client sample 88

Java client sample 78

special commands 45

Web services 43

Web services, deprecated 160, 161

WSDL 127

XSD 131

deprecated Web services

Demand Management 160, 161

Financial Management 149

E

encryption, password 31, 32, 84

executeWorkflowTransition 17, 39

F

Financial Management

special commands, deleted 43

Web services 41

Web services, deprecated 149

WSDL 111, 117, 132

XSD 116

G

getActualTime 20, 50
getCostFactors 18, 42
getCostRules 18, 42
getRequests 19, 44
getRequestTypeByFieldGroup 43
getRequestTypeFields 19, 43, 161
getRequestTypesByFieldGroup 19, 160
getResourceParticipation 18, 40
getResourcePools 18, 40

H

HTTP

See also authentication.

HTTPS

See also authentication.

I

importRequest 19, 43, 46
importWorkPlanTasks 17, 39

J

Java

sample Web services 78

javadocs

Web services 77

K

keystore

Web services 36, 84

keytool application 36

klsc_local_exec 69

ksc_export_actual_time 20, 53, 54

ksc_export_actual_time_as_xml 20, 53

ksc_export_budget 18, 43

ksc_export_budget_as_xml 18, 43

ksc_export_fields 19, 48

ksc_export_request 19, 46, 67, 162

ksc_export_status_change 19, 49, 68

ksc_export_XXX 30, 35

See also special commands.

L

logging

Web services 26

O

operations, Web services

Demand Management 43

Financial Management 41

Project Management 39

Resource Management 40

Time Management 50

P

password

clear-text 30

digest 30

encryption 31, 32, 84

passwordCallbackHandler 86

PPM Server

configuring Web services 25

Project Management

Web services 39

WSDL 91

XSD 96

R

read 18, 20, 41, 50, 153

readTasks 17, 39

request

Web services 81

Resource Management

Web services 40

WSDL 103

XSD 107

return message

Web services 83

S

SDK

Web services 75, 76

searchCostRules 18, 42

searchResourcePools 18, 41

searchTasks 17, 40

security

configuring Web services 84

Web services 24

server parameters

Web services 25, 35

setCostFactors 18, 42

setRequestFields 19, 44

setRequestRemoteReferenceStatus 19, 44, 45

setResourceParticipation 18, 41

SOAP message 60

special commands

Demand Management 45

ksc_local_exec 69

Time Management 53

Web services 15, 30, 35

specifications

Web services 24

system requirements

Web services 24

T

Time Management

special commands 53

Web services 50

Web services, replaced 50

WSDL 143

XSD 144

TimeExportService 53

Timestamp

Web services 29

toolkit, Web services 75

U

update 18, 41, 155

updateActualTime 53

updateCostRules 18, 43

updateProject 17, 40

updateRemoteReference 19, 43, 48, 49

updateResourcePools 18, 41

updateTaskActuals 17, 40

UsernameToken profile 24, 30

V

version 6.0

WSDL 63

W

Web services

See also operations.

application modules, for 38

configuring PPM Server 25

configuring security 84

integration 58

invoking 82

Java stubs 78

sample, .NET 87

sample, Java 78

specification 61, 63

WSDL 61, 63

XSD 62

Web Services Definition Language

See WSDL.

WSDL 61, 63

See also XSD.

Demand Management 127

Financial Management 111, 117, 132

Project Management 91

Resource Management 103

Time Management 143

version 6.0 63

WS-Security 24

X

XSD 62

See also WSDL.

Demand Management [131](#)

Financial Management [116](#)

Project Management [96](#)

Resource Management [107](#)

shared, in selected application modules [145](#)

Time Management [144](#)