

webMethods Task Engine API and Service Reference

Version 10.3

November 2018

This document applies to webMethods Task Engine Version 10.3 and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 2007-2018 Software AG, Darmstadt, Germany and/or Software AG USA Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at <http://softwareag.com/licenses>.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <http://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at <http://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

Use, reproduction, transfer, publication or disclosure is prohibited except as specifically provided for in your License Agreement with Software AG.

Table of Contents

About this Guide.....	7
Document Conventions.....	7
Online Information and Support.....	8
Data Protection.....	9
Using the Task Engine Built-in Services.....	11
Task Engine Built-In Services Location.....	13
Summary of Available Services.....	13
pub.task.taskclient:addTagsToTask.....	16
pub.task.taskclient:addTaskAttachment.....	16
pub.task.taskclient:addTaskComment.....	17
pub.task.taskclient:countTasksIndexed.....	18
pub.task.taskclient:countTasksIndexedHPSTRA.....	19
pub.task.taskclient:deleteTagsFromTask.....	21
pub.task.taskclient:deleteTask.....	21
pub.task.taskclient:deleteTaskAttachment.....	22
pub.task.taskclient:deleteTaskComment.....	22
pub.task.taskclient:formFlowTaskNotify.....	23
pub.task.taskclient:getTask.....	24
pub.task.taskclient:getTaskAttachments.....	26
pub.task.taskclient:getTaskAudit.....	26
pub.task.taskclient:getTaskComments.....	26
pub.task.taskclient:getTaskExpertList.....	27
pub.task.taskclient:listTagsByTask.....	27
pub.task.taskclient:queueTask.....	28
pub.task.taskclient:rollbackTask.....	29
pub.task.taskclient:searchTasks.....	30
pub.task.taskclient:searchTasksHPSTRA.....	32
pub.task.taskclient:searchTasksFields.....	34
pub.task.taskclient:searchTasksFieldsHPSTRA.....	36
pub.task.taskclient:searchTasksIndexed.....	38
pub.task.taskclient:searchTasksIndexedHPSTRA.....	40
pub.task.taskclient:updateTask.....	42
pub.task.taskclient:updateTaskAttachment.....	44
pub.task.taskclient:updateTaskComment.....	44
pub.task.taskclient:updateTaskTags.....	45
Summary of Data Structures.....	45
pub.task:Guidance.....	47
pub.task:Task.....	47
pub.task:TaskAttachment.....	48
pub.task:TaskAudit.....	49

pub.task:TaskCallbackService.....	50
pub.task:TaskComment.....	51
pub.task:TaskCompletionInfo.....	51
pub.task:TaskFields.....	52
pub.task:TaskInfo.....	53
pub.task:TaskQueueInfo.....	58
pub.task:TaskSearchQuery.....	60
pub.task:TaskSearchQueryTerm.....	62
pub.task.TaskSearchQueryV2.....	66
Using the Task Engine RESTful Web Services in Integration Server.....	69
Overview.....	71
Summary of REST Resources and Requests.....	71
DELETE a Task Instance.....	73
GET Task Instance Information.....	74
POST a New Task Instance.....	77
PUT Information in a Task Instance.....	78
GET Tasks with a Simple Search Request.....	80
GET HPSTRA-enabled Tasks with a Simple Search Request.....	81
POST a Complex Task Search Request.....	83
POST a Complex Search Request for HPSTRA-enabled Tasks.....	89
GET Task Audit Information.....	95
DELETE Audit Entries to Rollback Task.....	96
GET or POST a Form Flow Notification.....	96
DELETE Task Comment.....	98
GET Task Comments.....	98
PUT Updates into a Task Comment.....	99
POST Task Comment.....	100
DELETE Tags From Task.....	101
GET Tags By Task.....	101
GET Task Expert List.....	102
POST Tags To Task.....	102
PUT Task Tags.....	103
DELETE Task Attachment.....	103
GET Task Attachments.....	104
PUT Task Attachment.....	105
POST Task Attachment.....	105
Using the Task Engine RESTful Web Services in My webMethods Server.....	107
Overview.....	108
Summary of REST Resources and Requests Through My webMethods Server.....	108
DELETE a Task Instance.....	110
GET Task Instance Information.....	111
POST a New Task Instance.....	112
PUT Information in a Task Instance.....	113
GET Tasks with a Simple Search Request.....	115

POST a Complex Task Search Request.....	116
GET Task Audit Information.....	121
DELETE Audit Entries to Rollback Task.....	121
GET or POST a Form Flow Notification.....	122
DELETE Task Comment.....	123
GET Task Comments.....	124
PUT Updates into a Task Comment.....	124
POST Task Comment.....	125
DELETE Task Attachment.....	126
GET Task Attachments.....	126
PUT Task Attachment.....	127
POST Task Attachment.....	128
Using the Task Engine Web Service API.....	129
Overview.....	131
Task Service URLs.....	131
Extracting the WSDL for the Task Web Service.....	131
Using the Task Web Service Operations.....	132
Summary of Available Operations.....	133
addTagsToTask.....	134
addTaskAttachment.....	135
addTaskComment.....	136
countTasksIndexed.....	136
deleteTagsFromTask.....	138
deleteTask.....	138
deleteTaskAttachment.....	139
deleteTaskComment.....	139
formFlowTaskNotify.....	140
getTask.....	141
getTaskAttachments.....	142
getTaskAudit.....	142
getTaskComments.....	143
getTaskExpertList.....	143
listTagsByTask.....	143
queueTask.....	144
rollbackTask.....	145
searchTaskFields.....	145
searchTasks.....	147
searchTasksIndexed.....	148
updateTask.....	149
updateTaskAttachment.....	151
updateTaskComment.....	151
updateTaskTags.....	152
Summary of Data Structures.....	152
TaskAttachment.....	153

TaskAudit.....	154
TaskComment.....	156
TaskData.....	156
TaskInfo.....	158
Task.....	163
TaskSearchQuery.....	163
TaskSearchQueryTerm.....	165
TaskSearchQueryV2.....	168
Index.....	171

About this Guide

This guide describes how you can manipulate tasks running in webMethods Task Engine with the web service API and with the built-in Java services installed on Integration Server in the WmTaskClient package.

You can use the web service API to build web-service or .NET client applications that create, delete, locate, and modify tasks on Task Engine.

You can use the built-in services as templates to create services in Software AG Designer that run on Integration Server and that can create, modify, locate, queue, and delete tasks on the Task Engine to which Integration Server is connected.

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
Narrowfont	Identifies service names and locations in the format <i>folder.subfolder.service</i> , APIs, Java classes, methods, properties.
<i>Italic</i>	Identifies: Variables for which you must supply values specific to your own situation or environment. New terms the first time they occur in the text. References to other documentation sources.
Monospace font	Identifies: Text you must type in. Messages displayed by the system. Program code.
{ }	Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols.
	Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the symbol.

Convention	Description
[]	Indicates one or more options. Type only the information inside the square brackets. Do not type the [] symbols.
...	Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...).

Online Information and Support

Software AG Documentation Website

You can find documentation on the Software AG Documentation website at "<http://documentation.softwareag.com>". The site requires credentials for Software AG's Product Support site Empower. If you do not have Empower credentials, you must use the TECHcommunity website.

Software AG Empower Product Support Website

If you do not yet have an account for Empower, send an email to "empower@softwareag.com" with your name, company, and company email address and request an account.

Once you have an account, you can open Support Incidents online via the eService section of Empower at "<https://empower.softwareag.com/>".

You can find product information on the Software AG Empower Product Support website at "<https://empower.softwareag.com/>".

To submit feature/enhancement requests, get information about product availability, and download products, go to "[Products](#)".

To get information about fixes and to read early warnings, technical papers, and knowledge base articles, go to the "[Knowledge Center](#)".

If you have any questions, you can find a local or toll-free number for your country in our Global Support Contact Directory at "https://empower.softwareag.com/public_directory.asp" and give us a call.

Software AG TECHcommunity

You can find documentation and other technical information on the Software AG TECHcommunity website at "<http://techcommunity.softwareag.com>". You can:

- Access product documentation, if you have TECHcommunity credentials. If you do not, you will need to register and specify "Documentation" as an area of interest.
- Access articles, code samples, demos, and tutorials.

-
- Use the online discussion forums, moderated by Software AG professionals, to ask questions, discuss best practices, and learn how other customers are using Software AG technology.
 - Link to external websites that discuss open standards and web technology.

Data Protection

Software AG products provide functionality with respect to processing of personal data according to the EU General Data Protection Regulation (GDPR). Where applicable, appropriate steps are documented in the respective administration documentation.

1 Using the Task Engine Built-in Services

■ Task Engine Built-In Services Location	13
■ Summary of Available Services	13
■ pub.task.taskclient:addTagsToTask	16
■ pub.task.taskclient:addTaskAttachment	16
■ pub.task.taskclient:addTaskComment	17
■ pub.task.taskclient:countTasksIndexed	18
■ pub.task.taskclient:countTasksIndexedHPSTRA	19
■ pub.task.taskclient:deleteTagsFromTask	21
■ pub.task.taskclient:deleteTask	21
■ pub.task.taskclient:deleteTaskAttachment	22
■ pub.task.taskclient:deleteTaskComment	22
■ pub.task.taskclient:formFlowTaskNotify	23
■ pub.task.taskclient:getTask	24
■ pub.task.taskclient:getTaskAttachments	26
■ pub.task.taskclient:getTaskAudit	26
■ pub.task.taskclient:getTaskComments	26
■ pub.task.taskclient:getTaskExpertList	27
■ pub.task.taskclient:listTagsByTask	27
■ pub.task.taskclient:queueTask	28
■ pub.task.taskclient:rollbackTask	29
■ pub.task.taskclient:searchTasks	30
■ pub.task.taskclient:searchTasksHPSTRA	32
■ pub.task.taskclient:searchTasksFields	34
■ pub.task.taskclient:searchTasksFieldsHPSTRA	36
■ pub.task.taskclient:searchTasksIndexed	38
■ pub.task.taskclient:searchTasksIndexedHPSTRA	40

■ pub.task.taskclient:updateTask	42
■ pub.task.taskclient:updateTaskAttachment	44
■ pub.task.taskclient:updateTaskComment	44
■ pub.task.taskclient:updateTaskTags	45
■ Summary of Data Structures	45
■ pub.task:Guidance	47
■ pub.task:Task	47
■ pub.task:TaskAttachment	48
■ pub.task:TaskAudit	49
■ pub.task:TaskCallbackService	50
■ pub.task:TaskComment	51
■ pub.task:TaskCompletionInfo	51
■ pub.task:TaskFields	52
■ pub.task:TaskInfo	53
■ pub.task:TaskQueueInfo	58
■ pub.task:TaskSearchQuery	60
■ pub.task:TaskSearchQueryTerm	62
■ pub.task.TaskSearchQueryV2	66

Task Engine Built-In Services Location

The built-in services in this chapter are installed on Integration Server as part of the WmTaskClient package. You can find these Java services by opening the following folder location in the Package Navigator view in Software AG Designer:

Integration Server_directory\instances*instance_name* \ packages \ WmTaskClient \ pub

This chapter describes the services and supporting elements found in the \pub\task folder. You can use these services as templates to create custom services in Designer that can create, modify, locate, queue, and delete tasks on the Task Engine to which Integration Server is connected.

For more information about working with services in general, see *webMethods Service Development Help*.

Summary of Available Services

The following table lists the services, available in the \pub\task folder:

Element	Package and Description
pub.task.taskclient:addTagsToTask	WmTaskClient. Adds the specified tags to a task.
pub.task.taskclient:addTaskAttachment	WmTaskClient. Adds an attachment to a task.
pub.task.taskclient:addTaskComment	WmTaskClient. Adds a comment to a task.
pub.task.taskclient:countTasksIndexed	WmTaskClient. Service that returns the total count of tasks that match the specified search criteria. This service applies only to tasks with indexed business data.
pub.task.taskclient:countTasksIndexedHPSTR	WmTaskClient. Service that returns the total count of tasks that match the specified search criteria. This service applies only to HPSTRA-enabled tasks with indexed business data.

Element	Package and Description
pub.task.taskclient:deleteTagsFromTask	WmTaskClient. Deletes all tags from a task.
pub.task.taskclient:deleteTask	WmTaskClient. Service that deletes an instance of a task on the Task Engine to which the Integration Server is connected.
pub.task.taskclient:deleteTaskAttachment	WmTaskClient. Deletes the specified task attachment from a task.
pub.task.taskclient:deleteTaskComment	WmTaskClient. Deletes the specified task comment from a task.
pub.task.taskclient:formFlowTaskNotify	WmTaskClient. Service that notifies a waiting step in a task workflow that the workflow is completed.
pub.task.taskclient:getTask	WmTaskClient. Service that returns the TaskInfo object, and optionally the TaskData object, for a specified task.
pub.task.taskclient:getTaskAttachments	WmTaskClient. Returns all the attachments in a task.
pub.task.taskclient:getTaskAudit	WmTaskClient. Returns the audit log of all of the operations performed on a task.
pub.task.taskclient:getTaskComments	WmTaskClient. Returns all the comments in a task.
pub.task.taskclient:getTaskExpertList	WmTaskClient. This service returns a list of experts associated with the specified tags.
pub.task.taskclient:listTagsByTask	WmTaskClient. This service returns a list of tags associated with a task.
pub.task.taskclient:queueTask	WmTaskClient. Service that adds a task instance to the Task Engine's queue.

Element	Package and Description
pub.task.taskclient:rollbackTask	WmTaskClient. Enables you to roll back the task to any available audit point in the task's audit history.
pub.task.taskclient:searchTasks	WmTaskClient. Service that is used to retrieve tasks that match specified search criteria. This service applies only to tasks that use standard business data.
pub.task.taskclient:searchTasksHPSTRA	WmTaskClient. Retrieves tasks that match the specified search criteria. This service applies only to HPSTRA-enabled tasks with standard business data.
pub.task.taskclient:searchTasksFields	WmTaskClient. Service that returns a collection of named fields requested in each service that matches specified search criteria.
pub.task.taskclient:searchTasksFieldsHPSTRA	WmTaskClient. Service that returns a collection of named fields requested in each service that matches specified search criteria. This service applies only to HPSTRA-enabled tasks with indexed business data.
pub.task.taskclient:searchTasksIndexed	WmTaskClient. Service that returns a subset of tasks that match the specified search criteria on the Task Engine connected to the Integration Server.
pub.task.taskclient:searchTasksIndexedHPSTRA	WmTaskClient. Service that returns a subset of tasks that match the specified search criteria on the Task Engine connected to the Integration Server. This service applies only to HPSTRA-enabled tasks with indexed business data.
pub.task.taskclient:updateTask	WmTaskClient. This service modifies a task on the Task Engine that is

Element	Package and Description
	connected to the Integration Server. You use this service to change information in the task's TaskInfo and TaskData documents. This service applies only to tasks with Active, Error, and Suspended status. For information about updating tasks with other statuses, see <i>Usage Notes</i> .
pub.task.taskclient:updateTaskAttachment	WmTaskClient. Updates the specified task attachment.
pub.task.taskclient:updateTaskComment	WmTaskClient. Updates the specified task comment.
pub.task.taskclient:updateTaskTags	WmTaskClient. Updates the tags associated with a task.

pub.task.taskclient:addTagsToTask

WmTaskClient. Adds the specified tags to a task.

Input Parameters

<i>taskID</i>	String The unique identifier that the Task Engine assigns to the task.
<i>tags</i>	String Comma-separated list of text to be used to define tags. No character limit enforced. Alphanumeric characters, spaces, underscores, and dashes are allowed.

Output Parameters

None.

pub.task.taskclient:addTaskAttachment

WmTaskClient. Adds the specified attachment to a task.

Input Parameters

<i>attachment</i>	TaskAttachment The pub.task:TaskAttachment object representing the attachment to be added.
<i>encoding</i>	String Specify the encoding type for the file. Valid values include: <ul style="list-style-type: none"> ■ <code>binary</code> ■ <code>text</code>
<i>taskID</i>	String The unique identifier that the Task Engine assigns to the task.
<i>user</i>	String Optional. The user ID of the My webMethods Server user on whose behalf this operation executes.

Output Parameters

None.

pub.task.taskclient:addTaskComment

WmTaskClient. Adds a comment to a task.

Input Parameters

<i>attachments</i>	TaskAttachment [] Optional. An array of pub.task:TaskAttachment objects representing the attachments to be added to the comment being updated.
<i>html</i>	Boolean Optional. Set to: <ul style="list-style-type: none"> ■ <code>true</code> if the comment text contains HTML markup. ■ <code>false</code> (default) if the comment text is to be treated as plain text.
<i>taskID</i>	String The unique identifier that the Task Engine assigns to the task.
<i>text</i>	String Optional. The updated text for this comment.

user **String** Optional. The user ID of the My webMethods Server user on whose behalf this operation executes.

Output Parameters

commentID **String** The unique identifier that the Task Engine assigns to the newly added comment.

pub.task.taskclient:countTasksIndexed

WmTaskClient. Returns the total count of tasks that match the specified search criteria. This services applies only to tasks with indexed business data.

Use when specifying the *toIndex* and *fromIndex* parameters of the [pub.task.taskclient:searchTasksIndexed](#) service to avoid requesting an index beyond the size of the search results.

Input Parameters

businessData **Boolean** Optional. Indicates whether the search query operation includes indexed business data when processing the search query. Use this property when your search query contains search terms that reference indexed business data fields.

Set to:

- `true` to include business data in the search query processing. An error occurs if *businessData* is set to `true` and the task type does not contain any indexed fields.
- `false` when your search query does not require processing of business data or when your task type does not use any defined indexed business fields.

user **String** Optional. Specifies the user ID of the My webMethods Server user for which the operation executes. If *user* is not specified, the administrative user ID under which your client program logged on as is used. The operation only searches tasks the user can access.

TaskSearchQuery **TaskSearchQueryV2** Specifies the taskData search criteria. If TaskSearchQueryV2 data type does not contain the search criteria required to for the search, an error occurs. For more information, see [pub.task.TaskSearchQueryV2](#).

<i>searchUserTasks</i>	<p>Boolean Optional. Indicates whether the operation searches all tasks or only the user's inbox.</p> <p>Set to:</p> <ul style="list-style-type: none"> ■ <code>true</code> to only search the user's inbox; to use this value you must define a value for the <i>user</i> parameter. ■ <code>false</code> (default) to search all tasks to which <i>user</i> has access. <p>By default, only active tasks are returned. This can be overridden with the <i>showNonActiveTasks</i> parameter in pub.task:TaskSearchQuery.</p>
<i>retryOnFailureCount</i>	<p>String Optional. Specifies the number of times to attempt to invoke the service. The default is 0 times. When specified, this value will override the configuration value set on the WmTaskClient Home page in IS Administrator.</p>
<i>retryDelay</i>	<p>String Optional. Specifies the number of milliseconds to wait between attempts. The default is 1000 ms (this field is ignored if <i>retryOnFailureCount</i> = 0). When specified, this value will override the configuration value set on the WmTaskClient Home page in IS Administrator</p>

Output Parameters

<i>totalCount</i>	Returns the total number of tasks that match the query.
-------------------	---

pub.task.taskclient:countTasksIndexedHPSTRA

WmTaskClient. Applies only to Task Engine on My webMethods Server. Returns the total count of tasks that match the specified search criteria for tasks with indexed business data. This services applies only to tasks with indexed business data, stored in an Elasticsearch store by the HPSTRA module in Task Engine.

Use when specifying the *toIndex* and *fromIndex* parameters of the [pub.task.taskclient:searchTasksIndexed](#) service to avoid requesting an index beyond the size of the search results.

For more information about HPSTRA indexes, see *webMethods Task Engine User's Guide* and *webMethods BPM Task Development Help*.

Input Parameters

<i>businessData</i>	<p>Boolean Optional. Indicates whether the search query operation includes indexed business data when processing the search query.</p>
---------------------	---

Use this property when your search query contains search terms that reference indexed business data fields.

Set to:

- `true` to include business data in the search query processing. An error occurs if `businessData` is set to `true` and the task type does not contain any indexed fields.
- `false` when your search query does not require processing of business data or when your task type does not use any defined indexed business fields.

<i>user</i>	String Optional. Specifies the user ID of the My webMethods Server user for which the operation executes. If <i>user</i> is not specified, the administrative user ID under which your client program logged on as is used. The operation only searches tasks the user can access.
<i>TaskSearchQuery</i>	TaskSearchQueryV2 Specifies the taskData search criteria. If TaskSearchQueryV2 data type does not contain the search criteria required to for the search, an error occurs. For more information, see pub.task.TaskSearchQueryV2 .
<i>searchUserTasks</i>	<p>Boolean Optional. Indicates whether the operation searches all tasks or only the user's inbox.</p> <p>Set to:</p> <ul style="list-style-type: none"> ■ <code>true</code> to only search the user's inbox; to use this value you must define a value for the <i>user</i> parameter. ■ <code>false</code> (default) to search all tasks to which <i>user</i> has access. <p>By default, only active tasks are returned. This can be overridden with the <i>showNonActiveTasks</i> parameter in pub.task:TaskSearchQuery.</p>
<i>retryOnFailureCount</i>	String Optional. Specifies the number of times to attempt to invoke the service. The default is 0 times. When specified, this value will override the configuration value set on the WmTaskClient Home page in IS Administrator.
<i>retryDelay</i>	String Optional. Specifies the number of milliseconds to wait between attempts. The default is 1000 ms (this field is ignored if <code>retryOnFailureCount = 0</code>). When specified, this value will override the configuration value set on the WmTaskClient Home page in IS Administrator

Output Parameters

totalCount Returns the total number of tasks that match the query.

pub.task.taskclient:deleteTagsFromTask

WmTaskClient. Deletes all tags associated with a task.

Input Parameters

taskID **String** The unique identifier that the Task Engine assigns to the task.

Output Parameters

None.

pub.task.taskclient:deleteTask

WmTaskClient. Service that deletes an instance of a task on the Task Engine to which the Integration Server is connected.

Input Parameters

user **String** Optional. The user ID of the My webMethods Server user on whose behalf this service will execute.

If *user* is not specified, the user ID associated with the WmTaskClient package is used. For information about configuring this user ID, see *webMethods Task Engine User's Guide*.

taskID **String** The ID that identifies the task that you want to delete.

retryOnFailureCount **String** Optional. Specifies the number of times to attempt to invoke the service. The default is 0 times. When specified, this value will override the configuration value set on the WmTaskClient Home page in IS Administrator.

retryDelay **String** Optional. Specifies the number of milliseconds to wait between attempts. The default is 1000 ms (this field is ignored if

retryOnFailureCount = 0). When specified, this value will override the configuration value set on the WmTaskClient Home page in IS Administrator

Output Parameters

None.

Usage Notes

To delete a task using this service, you must supply the ID of the task that you want to delete. To obtain this ID, use the [pub.task.taskclient:searchTasks](#) service to locate the task and then extract the task ID from the *Task* document that the `searchTasks` service returns.

To delete a task successfully, the user ID in *user* must have permission to access the specified task. If the supplied ID does not have access to the task, or if the specified task does not exist, an exception will be thrown.

pub.task.taskclient:deleteTaskAttachment

WmTaskClient. Deletes the specified task attachment.

Input Parameters

<i>attachmentID</i>	String The unique identifier of the attachment to be deleted.
<i>taskID</i>	String The unique identifier that the Task Engine assigns to the task.
<i>user</i>	String Optional. The user ID of the My webMethods Server user on whose behalf this operation executes.

Output Parameters

None.

pub.task.taskclient:deleteTaskComment

WmTaskClient. Deletes a comment from a task.

Input Parameters

<i>commentID</i>	String The unique identifier of the comment to be deleted.
<i>taskID</i>	String The unique identifier that the Task Engine assigns to the task.
<i>user</i>	String Optional. The user ID of the My webMethods Server user on whose behalf this operation executes.

Output Parameters

None.

pub.task.taskclient:formFlowTaskNotify

WmTaskClient. Service that notifies a waiting step in a task workflow that the workflow is completed.

For example, when the final task step in a task workflow completes, that step waits for notification. However, in this case there are no more steps in the workflow to notify the waiting step. You can configure a following process service activity step to call this service and pass a result to the waiting step that indicates the workflow is complete. This enables the process to execute to completion.

For more information about task workflows, see “Working with Task Workflows” in the *webMethods BPM Task Development Help*.

Input Parameters

<i>correlationID</i>	String Use a task workflow correlation ID to synchronize the communication between the waiting step and the notifying step. This ensures correct data flow through the process. You can define any value but the task workflow correlation ID must be unique within the Process Engine environment.
----------------------	--

Important The task workflow correlation ID is completely different from and unrelated to the standard document correlation ID often used in process implementation.

<i>result</i>	String Optional. This can be any value to be passed back to the waiting component. Typically, you use this field to pass a status code to the waiting step to indicate the end of the workflow,
---------------	--

which enables the waiting step to take an action upon receiving the result.

localOnly

String Optional. Set to:

- `true` if the service is not being used in a cluster environment.
- `false` (default) if the service is being used in a clustered environment.

Output Parameters

None.

Usage Notes

Technically, you can also use this service within a workflow to notify a waiting task activity that the next task activity in the workflow is instantiated. However, the Java API `com.webmethods.portal.service.task.ITaskFormFlowService` is recommended for this task. To read the Java docs for this component, see the *webMethods CAF and My webMethods Server Java API Reference*, available from the [“Software AG Documentation web site”](#) under My webMethods Server in the corresponding webMethods Product Suite release number.

Sample Code Available

For more information about implementing a task workflow, you can examine and deploy a sample task application, process model, and Integration Server package that support a very simple loan application process. You can find the code samples in your My webMethods Server installation at: *Software AG_directory/MWS/components/samples/workflow*

A description of the sample code and its behavior can be found in the topic “Working with Task Workflows” in the *webMethods BPM Task Development Help*.

pub.task.taskclient:getTask

`WmTaskClient`. Service that returns the `TaskInfo` object, and optionally the `TaskData` object, for a specified task.

The `TaskInfo` object contains standard information about a task, including its status, expiration date, and the list of users to which it is assigned. The `TaskData` object contains the business data that is associated with the task. `TaskData` does not have a specified structure. Its content varies according to the task.

Input Parameters

user

String Optional. The user ID of the My webMethods Server user on whose behalf this service will execute.

If *user* is not specified, the user ID associated with the WmTaskClient package is used. For information about configuring this user ID, see *webMethods Task Engine User's Guide*.

<i>taskID</i>	String The ID that identifies the task that you want to retrieve.
<i>includeTaskData</i>	<p>String Optional. Specifies whether you want to retrieve the TaskData document as well as the TaskInfo document.</p> <p>Set to:</p> <ul style="list-style-type: none"> ■ <code>true</code> to retrieve TaskData in addition to TaskInfo. ■ <code>false</code> (default) to get TaskInfo only.
<i>retryOnFailureCount</i>	String Optional. Specifies the number of times to attempt to invoke the service. The default is 0 times. When specified, this value will override the configuration value set on the WmTaskClient Home page in IS Administrator.
<i>retryDelay</i>	String Optional. Specifies the number of milliseconds to wait between attempts. The default is 1000 ms (this field is ignored if <code>retryOnFailureCount = 0</code>). When specified, this value will override the configuration value set on the WmTaskClient Home page in IS Administrator

Output Parameters

<i>TaskInfo</i>	Document TaskInfo document containing standard information about the task. See " pub.task:TaskInfo " on page 53 for a description of the fields in this document.
<i>TaskData</i>	<p>Document A document containing the business data associated with the task. TaskData does not have a specified structure. Its content varies according to the task.</p> <p>TaskData is returned only if <i>includeTaskData</i> is set to true.</p>

Usage Notes

To get a task using this service, you must supply the ID of the task that you want to retrieve. To obtain this ID, use the [pub.task.taskclient:searchTasks](#) service to locate the task and then extract the task ID from the *Task* document that the `searchTasks` service returns.

To retrieve a task successfully, the user ID specified in *user* must have permission to access that task. If the supplied ID does not have access to the task, or if the specified task does not exist, an exception is thrown.

pub.task.taskclient:getTaskAttachments

WmTaskClient. Returns all the attachments in a task.

Input Parameters

taskID **String** The unique identifier that the Task Engine assigns to the task.

Output Parameters

taskAttachments **TaskAttachment[]** An array of [pub.task:TaskAttachment](#) objects that represent the attachments in the task

pub.task.taskclient:getTaskAudit

WmTaskClient. Returns the audit log of all of the operations performed on a task.

Input Parameters

taskID **String** The unique identifier that the Task Engine assigns to the task

Output Parameters

taskAudits **TaskAudit[]** An array of [pub.task:TaskAudit](#) objects representing the audit logs of the task.

pub.task.taskclient:getTaskComments

WmTaskClient. Returns all the comments in a task.

Input Parameters

taskID **String** The unique identifier that the Task Engine assigns to the task.

Output Parameters

taskComments **TaskComment[]** An array of [pub.task:TaskComment](#) objects that represent the comments in the task

pub.task.taskclient:getTaskExpertList

WmTaskClient. This service returns recommended experts for the specified tags.

Input Parameters

tags **String** Comma-separated list of tags.

Output Parameters

guidance **Guidance[]** An array of Guidance objects containing the names, ratings, and user IDs of the experts for the specified tags. See [pub.task:Guidance](#) for a description of the fields in this document.

pub.task.taskclient:listTagsByTask

WmTaskClient. This service returns a list of tags associated with a *taskID*.

Input Parameters

taskID **String** The unique identifier that the Task Engine assigns to the task.

Output Parameters

Tags **String Array** List of tags associated with a *taskID*.

pub.task.taskclient:queueTask

WmTaskClient. Service that adds a task instance to the Task Engine's queue.

Input Parameters

<i>user</i>	<p>String Optional. The user ID of the My webMethods Server user on whose behalf this service will execute.</p> <p>If <i>user</i> is not specified, the user ID associated with the WmTaskClient package is used. For information about configuring this user ID, see the PDF publication <i>webMethods Task Engine User's Guide</i>.</p>
<i>taskTypeID</i>	<p>String Specifies the type of task that you want to queue. Each task type that is deployed on the Task Engine has a unique ID. This ID is assigned by the developer when he or she creates a task application using Software AG Designer.</p> <p>If you do not know the ID for the task type that you want to queue, contact the administrator of the Task Engine to which Integration Server is connected. An administrator can obtain a list of the task types deployed on the Task Engine by viewing the Task Configuration panel on the Task Engine Administration page in My webMethods.</p> <p>Task type IDs are case-sensitive. The ID in <i>taskTypeID</i> must exactly match the ID as it is specified on the Task Engine.</p>
<i>TaskQueueInfo</i>	<p>Document Optional. A TaskQueueInfo document containing basic information about the task (e.g., name, priority, start date, list of users to which the task is assigned). See pub.task:TaskQueueInfo for a description of the fields in this document.</p>
<i>TaskData</i>	<p>Document Optional. The business data, if any, associated with this task. TaskData does not have a specified structure. Its content varies according to the task.</p>
<i>CallbackServiceName</i>	<p>String Optional. The name of a service on this Integration Server that will execute when the task ends (that is, when the status of the task changes to "completed," "cancelled," "expired," or "error").</p> <p>Example: <code>customer.accounts.closeAccount</code></p>

The signature of the callback service must match the specification described in “[pub.task:TaskCallbackService](#)” on [page 50](#).

<i>CallbackData</i>	Document Optional. Any data that you want the task to pass back to the callback service.
<i>ruleSet</i>	String Optional. The name of the rule set to be applied for this task. This must match one of the rule set names defined for the task in the Task Editor of Software AG Designer. When a rule set is specified, then assignments and events for the specified rule set only will be executed for this task instance
<i>retryOnFailureCount</i>	String Optional. Specifies the number of times to attempt to invoke the service. The default is 0 times. When specified, this value will override the configuration value set on the WmTaskClient Home page in IS Administrator.
<i>retryDelay</i>	String Optional. Specifies the number of milliseconds to wait between attempts. The default is 1000 ms (this field is ignored if <code>retryOnFailureCount = 0</code>). When specified, this value will override the configuration value set on the WmTaskClient Home page in IS Administrator

Output Parameters

taskID **String** The unique identifier that the Task Engine assigns to the task.

Usage Notes

The user ID specified in *user* must have permission to queue tasks on the Task Engine. If the supplied ID does not have this permission, an exception is generated.

If the date specified in *taskScheduleDate* is later than the current date, the status of the task will be set to `scheduled`. When the task starts or queues at the scheduled time, the global rule for task schedule changes the status of the task from `scheduled` to `active`.

pub.task.taskclient:rollbackTask

WmTaskClient. Enables you to roll back the task to any available audit point in the task's audit history.

Input Parameters

<i>auditEntryID</i>	Integer The ID of the audit log entry that you want to roll the task back to.
<i>taskID</i>	String The unique identifier that the Task Engine assigns to the task.

Output Parameters

None.

pub.task.taskclient:searchTasks

WmTaskClient. Service that is used to retrieve tasks that match specified search criteria. This service applies only to tasks that use standard business data.

Input Parameters

<i>includeTaskData</i>	<p>String Optional. Specifies whether the service will return the TaskData document as well as the TaskInfo document for each task matching the search criteria. Set to:</p> <ul style="list-style-type: none"> ■ <code>true</code> to include the TaskData document in the result set. ■ <code>false</code> (default) to include only the TaskInfo document in the result set.
------------------------	--

<i>user</i>	<p>String Optional. The user ID of the My webMethods Server user on whose behalf this service will execute. If <i>user</i> is not specified, the user ID associated with the WmTaskClient package is used. For information about configuring this user ID, see <i>webMethods Task Engine User's Guide</i>.</p>
-------------	---

Note: Only tasks to which *user* has access are searched.

<i>TaskSearchQuery</i>	<p>Document Optional. TaskSearchQuery document, which specifies the search criteria. See “pub.task:TaskSearchQuery” on page 60 for a description of the fields in this document.</p> <p>If <i>TaskSearchQuery</i> is null, all tasks for <i>user</i> will be returned. For additional behavior, see the <i>MaxResults</i> parameter in “pub.task:TaskSearchQuery” on page 60.</p>
------------------------	--

searchUserTasks **String** Optional. Specifies whether the service searches all tasks or just the user's inbox. Set to:

- `true` to search only the inbox for *user*.
- `false` (default) to search all tasks to which *user* has access.

By default, only active tasks are returned. This can be overridden with the *showNonActiveTasks* parameter in [pub.task:TaskSearchQuery](#).

retryOnFailureCount **String** Optional. Specifies the number of times to attempt to invoke the service. The default is 0 times. When specified, this value will override the configuration value set on the WmTaskClient Home page in IS Administrator.

retryDelay **String** Optional. Specifies the number of milliseconds to wait between attempts. The default is 1000 ms (this field is ignored if *retryOnFailureCount* = 0). When specified, this value will override the configuration value set on the WmTaskClient Home page in IS Administrator

Output Parameters

Task **Document List** Result set containing the tasks that match the search criteria in *TaskSearchQuery*. Each document in the result set has the following structure:

- *TaskID* **String** The identifier assigned to the task.
- *TaskInfo* **Document** A TaskInfo document containing standard information about the task. See [pub.task:TaskInfo](#) for a description of the fields in this document.
- *TaskData* **Document** A TaskData document containing the business data associated with the task. TaskData does not have a specified structure. Its content varies according to the task. Returned only if *includeTaskData* is set to `true`.

Usage Notes

When a task is configured with a standard search provider, the search examines *all business data in all the tasks in the inbox*. If the number of tasks in an inbox is relatively small, this search executes quickly. However, when a large number of tasks are present, a standard (full) search can take significantly longer.

For faster searches, consider implementing an indexed search provider, which examines only specified indexed business data fields table and ignores all other fields, resulting in

a much faster search. In this case, the task developer must mark one or more business data fields as indexed fields, otherwise the search will return no results.

For detailed information about implementing indexed searching, see *webMethods BPM Task Development Help* (available online in Software AG Designer and as a PDF publication) and the PDF publication *webMethods Task Engine User's Guide*.

For more information about searching indexed business data fields with a service, see [“pub.task.taskclient:searchTasksIndexed” on page 38](#).

pub.task.taskclient:searchTasksHPSTRA

WmTaskClient. Applies only to Task Engine on My webMethods Server. Retrieves tasks that match the specified search criteria. This service applies only to HPSTRA-enabled tasks with standard business data.

For more information about HPSTRA indexes and HPSTRA-enabled tasks, see *webMethods Task Engine User's Guide* and *webMethods BPM Task Development Help*.

Input Parameters

<i>includeTaskData</i>	<p>String Optional. Specifies whether the service will return the TaskData document as well as the TaskInfo document for each task matching the search criteria. Set to:</p> <ul style="list-style-type: none"> ■ <code>true</code> to include the TaskData document in the result set. ■ <code>false</code> (default) to include only the TaskInfo document in the result set.
<i>user</i>	<p>String Optional. The user ID of the My webMethods Server user on whose behalf this service will execute. If <i>user</i> is not specified, the user ID associated with the WmTaskClient package is used. For information about configuring this user ID, see <i>webMethods Task Engine User's Guide</i>.</p> <div style="background-color: #f0f0f0; padding: 5px; margin-top: 10px;"> <p>Note: Only tasks to which <i>user</i> has access are searched.</p> </div>
<i>TaskSearchQuery</i>	<p>Document Optional. TaskSearchQuery document, which specifies the search criteria. See “pub.task:TaskSearchQuery” on page 60 for a description of the fields in this document.</p> <p>If <i>TaskSearchQuery</i> is null, all tasks for <i>user</i> will be returned. For additional behavior, see the <i>MaxResults</i> parameter in “pub.task:TaskSearchQuery” on page 60.</p>
<i>searchUserTasks</i>	<p>String Optional. Specifies whether the service searches all tasks or just the user's inbox. Set to:</p>

- `true` to search only the inbox for *user*.
- `false` (default) to search all tasks to which *user* has access.

By default, only active tasks are returned. This can be overridden with the *showNonActiveTasks* parameter in [pub.task:TaskSearchQuery](#).

retryOnFailureCount **String** Optional. Specifies the number of times to attempt to invoke the service. The default is 0 times. When specified, this value will override the configuration value set on the WmTaskClient Home page in IS Administrator.

retryDelay **String** Optional. Specifies the number of milliseconds to wait between attempts. The default is 1000 ms (this field is ignored if *retryOnFailureCount* = 0). When specified, this value will override the configuration value set on the WmTaskClient Home page in IS Administrator

Output Parameters

Task **Document List** Result set containing the tasks that match the search criteria in *TaskSearchQuery*. Each document in the result set has the following structure:

- *TaskID* **String** The identifier assigned to the task.
- *TaskInfo* **Document** A TaskInfo document containing standard information about the task. See [pub.task:TaskInfo](#) for a description of the fields in this document.
- *TaskData* **Document** A TaskData document containing the business data associated with the task. TaskData does not have a specified structure. Its content varies according to the task. Returned only if *includeTaskData* is set to `true`.

Usage Notes

When a task is configured with a standard search provider, the search examines *all business data in all the tasks in the inbox*. If the number of tasks in an inbox is relatively small, this search executes quickly. However, when a large number of tasks are present, a standard (full) search can take significantly longer.

For faster searches, consider implementing an indexed search provider, which examines only specified indexed business data fields table and ignores all other fields, resulting in a much faster search. In this case, the task developer must mark one or more business data fields as indexed fields, otherwise the search will return no results.

For detailed information about implementing indexed searching, see *webMethods BPM Task Development Help* and *webMethods Task Engine User's Guide*.

For more information about searching indexed business data fields with a service, see [“pub.task.taskclient:searchTasksIndexed”](#) on page 38.

pub.task.taskclient:searchTasksFields

WmTaskClient. Used to return a collection of named fields requested in each service that matches specified search criteria.

Note: This service can only return primitive types (Strings, Numbers, Dates, etc.) or lists (arrays) of primitives. It cannot return complex objects such as all task business data.

Input Parameters

<i>includeTaskData</i>	<p>String Optional. Specifies whether the service will return the TaskData document as well as the TaskInfo document for each task matching the search criteria.</p> <p>Set to:</p> <ul style="list-style-type: none"> ■ <code>true</code> to include the TaskData document in the result set. ■ <code>false</code> (default) to include only the TaskInfo document in the result set.
<i>user</i>	<p>String Optional. The user ID of the My webMethods Server user on whose behalf this service will execute.</p> <p>If <i>user</i> is not specified, the user ID associated with the WmTaskClient package is used. For information about configuring this user ID, see <i>webMethods Task Engine User's Guide</i>.</p> <p>Note: Only tasks to which <i>user</i> has access are searched.</p>
<i>TaskSearchQuery</i>	<p>Document Optional. TaskSearchQuery document, which specifies the search criteria. See “pub.task:TaskSearchQuery” on page 60 for a description of the fields in this document.</p> <p>If <i>TaskSearchQuery</i> is null, all tasks for <i>user</i> will be returned. For additional behavior, see the <i>MaxResults</i> parameter in “pub.task:TaskSearchQuery” on page 60.</p>
<i>searchUserTasks</i>	<p>String Optional. Specifies whether the service searches all tasks or just the user's inbox.</p>

Set to:

- `true` to search only the inbox for *user* .
- `false` (default) to search all tasks to which *user* has access.

By default, only active tasks are returned. This can be overridden with the `showNonActiveTasks` parameter in [pub.task:TaskSearchQuery](#).

fields

String List of task fields to be returned by this service. For information about specifying fields, see the *fields* and *operator* elements in “[pub.task:TaskSearchQueryTerm](#)” on page 62. For example, suppose you want to return the following fields as result of the query:

- `taskID`
- The business data field `orderID` from task data documents.

You would then pass in following field specifications:

```
#{currentTask.taskInfo.taskID}
```

```
#{currentTask.taskData.order.orderID}
```

This API will return values for these fields only.

*retryOnFailure
Count*

String Optional. Specifies the number of times to attempt to invoke the service. The default is 0 times. When specified, this value will override the configuration value set on the WmTaskClient Home page in IS Administrator.

retryDelay

String Optional. Specifies the number of milliseconds to wait between attempts. The default is 1000 ms (this field is ignored if `retryOnFailureCount` = 0). When specified, this value will override the configuration value set on the WmTaskClient Home page in IS Administrator

Output Parameters

Tasks

Document List Result set containing a list of documents whose document type is defined in “[pub.task:TaskFields](#)” on page 52.

Usage Notes

Using [pub.task.taskclient:searchTasksFields](#) instead of searching with [pub.task:TaskSearchQuery](#) can significantly improve search performance. Searching with [pub.task:TaskSearchQuery](#) returns all TaskInfo data plus task data documents which can be very large. Because this is a Web service call, the SOAP

response size becomes an issue, severely limiting performance of this service. With [pub.task.taskclient:searchTasksFields](#), a very narrow search can be tailored, often requiring only a few taskInfo fields and some business data fields. Using [pub.task.taskclient:searchTasksFields](#) is strongly recommended to prevent problems.

pub.task.taskclient:searchTasksFieldsHPSTRA

WmTaskClient. Applies only to Task Engine on My webMethods Server. Returns a collection of named fields, requested in each service that matches specified search criteria. This service applies to HPSTRA-enabled tasks, regardless of indexing.

For more information about HPSTRA indexes and HPSTRA-enabled tasks, see *webMethods Task Engine User's Guide* and *webMethods BPM Task Development Help*.

Note: This service can only return primitive types (Strings, Numbers, Dates, etc.) or lists (arrays) of primitives. It cannot return complex objects such as all task business data.

Input Parameters

includeTaskData **String** Optional. Specifies whether the service will return the TaskData document as well as the TaskInfo document for each task matching the search criteria.

Set to:

- `true` to include the TaskData document in the result set.
- `false` (default) to include only the TaskInfo document in the result set.

user **String** Optional. The user ID of the My webMethods Server user on whose behalf this service will execute.

If *user* is not specified, the user ID associated with the WmTaskClient package is used. For information about configuring this user ID, see *webMethods Task Engine User's Guide*.

Note: Only tasks to which *user* has access are searched.

TaskSearchQuery **Document** Optional. TaskSearchQuery document, which specifies the search criteria. See [“pub.task:TaskSearchQuery” on page 60](#) for a description of the fields in this document.

If *TaskSearchQuery* is null, all tasks for *user* will be returned. For additional behavior, see the *MaxResults* parameter in [“pub.task:TaskSearchQuery” on page 60](#).

<i>searchUserTasks</i>	<p>String Optional. Specifies whether the service searches all tasks or just the user's inbox.</p> <p>Set to:</p> <ul style="list-style-type: none"> ■ <code>true</code> to search only the inbox for <i>user</i> . ■ <code>false</code> (default) to search all tasks to which <i>user</i> has access. <p>By default, only active tasks are returned. This can be overridden with the <i>showNonActiveTasks</i> parameter in pub.task:TaskSearchQuery.</p>
<i>fields</i>	<p>String List of task fields to be returned by this service. For information about specifying fields, see the <i>fields</i> and <i>operator</i> elements in “pub.task:TaskSearchQueryTerm” on page 62. For example, suppose you want to return the following fields as result of the query:</p> <ul style="list-style-type: none"> ■ <code>taskID</code> ■ The business data field <i>orderID</i> from task data documents. <p>You would then pass in following field specifications:</p> <pre>#{currentTask.taskInfo.taskID} #{currentTask.taskData.order.orderID}</pre> <p>This API will return values for these fields only.</p>
<i>retryOnFailureCount</i>	<p>String Optional. Specifies the number of times to attempt to invoke the service. The default is 0 times. When specified, this value will override the configuration value set on the WmTaskClient Home page in IS Administrator.</p>
<i>retryDelay</i>	<p>String Optional. Specifies the number of milliseconds to wait between attempts. The default is 1000 ms (this field is ignored if <i>retryOnFailureCount</i> = 0). When specified, this value will override the configuration value set on the WmTaskClient Home page in IS Administrator</p>

Output Parameters

<i>Tasks</i>	<p>Document List Result set containing a list of documents whose document type is defined in “pub.task:TaskFields” on page 52.</p>
--------------	---

Usage Notes

Using [pub.task.taskclient:searchTasksFieldsHPSTRA](#) instead of searching with [pub.task:TaskSearchQuery](#) can significantly improve search performance. Searching with [pub.task:TaskSearchQuery](#) returns all TaskInfo data plus task data documents which can be very large. Because this is a Web service call, the SOAP response size becomes an issue, severely limiting performance of this service. With [pub.task.taskclient:searchTasksFieldsHPSTRA](#), a very narrow search can be tailored, often requiring only a few taskInfo fields and some business data fields. Using [pub.task.taskclient:searchTasksFieldsHPSTRA](#) is strongly recommended to prevent problems.

pub.task.taskclient:searchTasksIndexed

WmTaskClient. Returns a subset of tasks that match the specified search criteria *for indexed business data fields only*. Only indexed fields are returned as part of the task data.

The query must contain a search term specifying the *taskTypeID* when searching for business data fields. The Task Engine uses the *taskTypeID* to identify the specific index table to use. You can search for multiple task types in a single query by specifying multiple *taskTypeID* values when you define the search criteria. If you are not searching for business data fields, then a *taskTypeID* is not required.

Use this service only when working with indexed business data fields. If you are working with standard business data fields, use “[pub.task.taskclient:searchTasks](#)” on [page 30](#).

Input Parameters

businessData

Boolean Optional. Indicates whether the search query operation includes indexed business data when processing the search query.

Use this property when your search query contains search terms that reference indexed business data fields.

Set to:

- `true` to include business data in the search query processing.
- `false` when your search query does not require processing of business data or when your task type does not use any defined indexed business fields.

The default value for *businessData* is `false`.

<i>user</i>	String Optional. Specifies the user ID of the Integration Server user for which the operation executes. If <i>user</i> is not specified, the administrative user ID under which your client program logged on as is used. The operation only searches tasks the user can access.
<i>TaskSearchQuery</i>	TaskSearchQueryV2 Specifies the pub.task.TaskSearchQueryV2 document type that provides the search criteria. If <i>TaskSearchQueryV2</i> is null, an error occurs.
<i>searchUserTasks</i>	<p>String Optional. Specifies whether the service searches all tasks or just the user's inbox.</p> <p>Set to:</p> <ul style="list-style-type: none"> ■ <code>true</code> to search only the inbox for <i>user</i>. ■ <code>false</code> (default) to search all tasks to which <i>user</i> has access. <p>By default, only active tasks are returned. This can be overridden with the <i>showNonActiveTasks</i> parameter in pub.task:TaskSearchQuery.</p>
<i>retryOnFailureCount</i>	String Optional. Specifies the number of times to attempt to invoke the service. The default is 0 times. When specified, this value will override the configuration value set on the WmTaskClient Home page in IS Administrator.
<i>retryDelay</i>	String Optional. Specifies the number of milliseconds to wait between attempts. The default is 1000 ms (this field is ignored if <i>retryOnFailureCount</i> = 0). When specified, this value will override the configuration value set on the WmTaskClient Home page in IS Administrator

Output Parameters

<i>Task</i>	<p>Document List Contains <i>TaskData</i> documents that match the search criteria in <i>TaskSearchQuery</i>. Each document in the result set has the following structure:</p> <ul style="list-style-type: none"> ■ <i>TaskID</i> String The identifier assigned to the task.
-------------	--

- **TaskInfo Document** TaskInfo document containing standard information about the task. See [pub.task:TaskInfo](#) for a description of the fields in this document.
- **TaskData Document** TaskData document containing the indexed business data associated with the task. Returned only if *businessData* is set to `true`.

Usage Notes

If no tasks match the search criteria, an empty TaskData document is returned. The search query must contain a search term specifying the *taskTypeID* to search business fields and create an index of the results.

For detailed information about implementing indexed searching, see *webMethods BPM Task Development Help* (available online in Software AG Designer and as a PDF publication) and the PDF publication *webMethods Task Engine User's Guide*.

pub.task.taskclient:searchTasksIndexedHPSTRA

WmTaskClient. Applies only to Task Engine on My webMethods Server. Returns a subset of tasks that match the specified search criteria *for indexed business data fields only*. This service applies only to tasks with indexed business data, stored in an Elasticsearch store by the HPSTRA module in Task Engine. Only indexed fields are returned as part of the task data.

The query must contain a search term specifying the *taskTypeID* when searching for business data fields. The Task Engine uses the *taskTypeID* to identify the specific index table to use. You can search for multiple task types in a single query by specifying multiple *taskTypeID* values when you define the search criteria. If you are not searching for business data fields, then a *taskTypeID* is not required.

Use this service only when working with HPSTRA-enabled indexed business data fields. If you are working with standard business data fields, use [“pub.task.taskclient:searchTasks”](#) on page 30.

For more information about HPSTRA indexes and HPSTRA-enabled tasks, see *webMethods Task Engine User's Guide* and *webMethods BPM Task Development Help*.

Input Parameters

businessData

Boolean Optional. Indicates whether the search query operation includes indexed business data when processing the search query.

Use this property when your search query contains search terms that reference indexed business data fields.

	<p>Set to:</p> <ul style="list-style-type: none"> ■ <code>true</code> to include business data in the search query processing. ■ <code>false</code> when your search query does not require processing of business data or when your task type does not use any defined indexed business fields. <p>The default value for <i>businessData</i> is <code>false</code>.</p>
<i>user</i>	<p>String Optional. Specifies the user ID of the Integration Server user for which the operation executes. If <i>user</i> is not specified, the administrative user ID under which your client program logged on as is used. The operation only searches tasks the user can access.</p>
<i>TaskSearchQuery</i>	<p>TaskSearchQueryV2 Specifies the pub.task.TaskSearchQueryV2 document type that provides the search criteria. If TaskSearchQueryV2 is null, an error occurs.</p>
<i>searchUserTasks</i>	<p>String Optional. Specifies whether the service searches all tasks or just the user's inbox.</p> <p>Set to:</p> <ul style="list-style-type: none"> ■ <code>true</code> to search only the inbox for <i>user</i>. ■ <code>false</code> (default) to search all tasks to which <i>user</i> has access. <p>By default, only active tasks are returned. This can be overridden with the <i>showNonActiveTasks</i> parameter in pub.task:TaskSearchQuery.</p>
<i>retryOnFailureCount</i>	<p>String Optional. Specifies the number of times to attempt to invoke the service. The default is 0 times. When specified, this value will override the configuration value set on the WmTaskClient Home page in IS Administrator.</p>
<i>retryDelay</i>	<p>String Optional. Specifies the number of milliseconds to wait between attempts. The default is 1000 ms (this field is ignored if <i>retryOnFailureCount</i> = 0). When specified, this value will override the configuration value set on the WmTaskClient Home page in IS Administrator</p>

Output Parameters

<i>Task</i>	<p>Document List Contains <i>TaskData</i> documents that match the search criteria in <i>TaskSearchQuery</i>. Each document in the result set has the following structure:</p> <ul style="list-style-type: none"> ■ TaskID String The identifier assigned to the task. ■ TaskInfo Document TaskInfo document containing standard information about the task. See pub.task:TaskInfo for a description of the fields in this document. ■ TaskData Document TaskData document containing the indexed business data associated with the task. Returned only if <i>businessData</i> is set to <code>true</code>.
-------------	--

Usage Notes

If no tasks match the search criteria, an empty *TaskData* document is returned. The search query must contain a search term specifying the *taskTypeID* to search business fields and create an index of the results.

For detailed information about implementing indexed searching, see *webMethods BPM Task Development Help* and *webMethods Task Engine User's Guide*.

pub.task.taskclient:updateTask

WmTaskClient. This service modifies a task on the Task Engine that is connected to the Integration Server. You use this service to change information in the task's *TaskInfo* and *TaskData* documents. This service applies only to tasks with status *Active*, *Error*, and *Suspended*. For information about updating tasks with other statuses, see *Usage Notes*.

To end a task, you use this service to change the *status* field in the task's *TaskInfo* document to "closed" or "cancelled" as appropriate.

Input Parameters

<i>user</i>	<p>String Optional. The user ID of the My webMethods Server user on whose behalf this service will execute.</p> <p>If <i>user</i> is not specified, the user ID associated with the <i>WmTaskClient</i> package is used. For information about configuring this user ID, see the PDF publication <i>webMethods Task Engine User's Guide</i>.</p>
<i>taskID</i>	<p>String The ID that identifies the task you want to modify.</p>

<i>TaskInfo</i>	Document TaskInfo object that specifies the changes you want to make to the standard information maintained for the task. See “ pub.task:TaskInfo ” on page 53 for a description of the fields in this document.
<i>TaskData</i>	Document TaskData object that contains the business data that you want to associate with the task. TaskData does not have a specified structure. Its content varies according to the task.
<i>retryOnFailureCount</i>	String Optional. Specifies the number of times to attempt to invoke the service. The default is 0 times. When specified, this value will override the configuration value set on the WmTaskClient Home page in IS Administrator.
<i>retryDelay</i>	String Optional. Specifies the number of milliseconds to wait between attempts. The default is 1000 ms (this field is ignored if <i>retryOnFailureCount</i> = 0). When specified, this value will override the configuration value set on the WmTaskClient Home page in IS Administrator

Output Parameters

None.

Usage Notes

When you use this service to update information in the task's TaskInfo document, the Task Engine modifies *only* the fields that have assigned values in *TaskInfo*. All other fields in the target task retain their existing values. For example, to change just the name assigned to a task, set only the *name* field in the *TaskInfo* document that you pass to this service.

When you use this service to update information in the task's TaskData document, the TaskData document that you pass to the service *replaces* the task's existing TaskData document in its entirety.

When updating task statuses with this service, the following limitations apply:

- If the current status of the task is Error, you can change it only to Completed.
- If the current status of the task is Suspended, you cannot set it to Completed or Expired.

This service applies only to tasks with status Active, Error, and Suspended. To enable the service to update tasks with other statuses, set the additional JVM property - `Dupdate.completed.task` to `true`, as described in *webMethods Task Engine User's Guide*.

Tasks with status Error that belong to a process instance cannot be updated, regardless of Task Engine optional settings.

You can edit the *taskScheduleDate* field only when the task is in `scheduled` state.

You cannot edit the schedules of task instances created using Task Engine 9.9 or earlier because those task instances will be in `active` state.

pub.task.taskclient:updateTaskAttachment

WmTaskClient. Updates the specified task attachment.

Input Parameters

<i>attachment</i>	TaskAttachment The pub.task:TaskAttachment object representing the updated attachment to be used to update the existing attachment.
<i>attachmentID</i>	String The unique identifier of the attachment to be updated.
<i>encoding</i>	String Specify the encoding type for the file. Valid values include: <ul style="list-style-type: none">■ binary■ text
<i>taskID</i>	String The unique identifier that the Task Engine assigns to the task.
<i>user</i>	String Optional. The user ID of the My webMethods Server user on whose behalf this operation executes.

Output Parameters

None.

pub.task.taskclient:updateTaskComment

WmTaskClient. Updates a comment in a task.

Input Parameters

<i>attachments</i>	TaskAttachment[] An array of pub.task:TaskAttachment objects representing the attachments to be added to the comment being updated.
--------------------	--

<i>commentID</i>	String The unique identifier of the comment to be deleted.
<i>deleteAttachments</i>	String[] An array of IDs of the attachments to be deleted from the comment
<i>taskID</i>	String The unique identifier that the Task Engine assigns to the task.
<i>text</i>	String Optional. The updated text for this comment.
<i>user</i>	String Optional. The user ID of the My webMethods Server user on whose behalf this operation executes.

Output Parameters

None.

pub.task.taskclient:updateTaskTags

WmTaskClient. Updates the tags associated with a *taskID*.

Input Parameters

<i>taskID</i>	String The unique identifier that the Task Engine assigns to the task.
<i>tags</i>	String Comma-separated list of text to be used to define tags. No character limit enforced. Alphanumeric characters, spaces, underscores, and dashes are allowed.

Output Parameters

None.

Summary of Data Structures

The following table lists the elements, available in the \pub\task folder:

Element	Package and Description
pub.task:Guidance	WmTaskClient. A document type containing information about the name, rating, and user ID of the expert.
pub.task:Task	WmTaskClient. A document type that identifies the characteristics of a particular task.
pub.task:TaskAttachment	WmTaskClient. Document type representing an attachment.
pub.task:TaskAudit	WmTaskClient. Document type containing information returned as output from <code>GetTaskAudit</code> .
pub.task:TaskCallbackService	WmTaskClient. Specification that describes the inputs and outputs required for a callback service.
pub.task:TaskComment	WmTaskClient. Document type representing a comment in a task.
pub.task:TaskCompletionInfo	WmTaskClient. Document type containing information about the completion of a task. The Task Engine uses this document type to return data about the completion of a task to a callback service.
pub.task:TaskFields	WmTaskClient. Document type that contains information about task fields returned by pub.task.taskclient:searchTasksFields and pub.task.taskclient:searchTasksFieldsHPSTRA .
pub.task:TaskInfo	WmTaskClient. Document type that contains standard information about a task. You specify certain fields in <code>TaskInfo</code> when you queue a task using pub.task.taskclient:queueTask .

Element	Package and Description
pub.task:TaskQueueInfo	WmTaskClient. Document type used by pub.task.taskclient:queueTask to set basic values in TaskInfo.
pub.task:TaskSearchQuery	WmTaskClient. Document type used as input to the pub.task.taskclient:searchTasks and “ pub.task.taskclient:searchTasksHPSTRA ” on page 32 services. This service applies only to tasks that use standard business data.
pub.task:TaskSearchQueryTerm	WmTaskClient. Document type that you use to specify search criteria for the pub.task.taskclient:searchTasks service.
pub.task.TaskSearchQueryV2	WmTaskClient. Indicates the document type that you use to specify the search criteria for the pub.task.taskclient:searchTasksIndexed and pub.task.taskclient:searchTasksIndexedHPSTRA services.

pub.task:Guidance

WmTaskClient. A document type containing information about the name, rating, and user ID of the expert.

<i>displayName</i>	String Displayable name of the expert user.
<i>rating</i>	Double Rating of the expert user.
<i>userId</i>	String User ID (on My webMethods Server) of the expert user.

pub.task:Task

WmTaskClient. A document type that identifies the characteristics of a particular task.

Parameters

<i>TaskID</i>	String The value of the identifier that the Task Engine assigns to the task.
<i>TaskInfo</i>	Document TaskInfo document containing standard information about the task. See “ pub.task:TaskInfo ” on page 53 for a description of the fields in this document.
<i>TaskData</i>	Document A document containing the business data associated with the task. TaskData does not have a specified structure. Its content varies according to the task.

pub.task:TaskAttachment

WmTaskClient. Document type representing an attachment.

Parameters

<i>base64Data</i>	String A Base64 encoded string representation of the data for the attachment.
<i>contentType</i>	String[] Content-type of the data in the attachment.
<i>downloadLink</i>	String A link from My webMethods Server for downloading the attachment.
<i>fileName</i>	String Name of the file in the attachment.
<i>iconURL</i>	String URL of the icon to be used to represent the attachment in the task user interface.
<i>id</i>	String Read only. Unique ID of an existing attachment.
<i>lastModifiedDate</i>	Date The date and time of last modification of the attachment.
<i>name</i>	String Name of the attachment.
<i>contentLength</i>	Long Length of the data in the attachment.
<i>deletable</i>	Boolean Set to:

- `true` if the attachment can be deleted by a user with appropriate permissions. Default.
- `false` if the attachment cannot be deleted by a user.

*updatable***Boolean** Set to:

- `true` if the attachment can be updated by a user with appropriate permissions. Default.
- `false` if the attachment cannot be updated by a user.

pub.task:TaskAudit

WmTaskClient. Document type containing information returned as output from [pub.task.taskclient:getTaskAudit](#).

Parameters

<i>createdDate</i>	Date The date and time the operation was applied to the audited task.
<i>difference</i>	String[] An array of strings representing the changes made as part of an audit log entry.
<i>sourceURI</i>	String URI of the source of this change.
<i>TaskURI</i>	String URI of the audited task.
<i>id</i>	Integer Unique ID of this audit log entry.
<i>operation</i>	<p>Integer Numerical representation of the operation performed. See Usage Notes below for more information.</p> <p>Selected TASK_OPERATION_* fields from <code>com/webmethods/caf/faces/data/task/ITaskConstants</code> are valid values for this parameter, as described in Usage Notes below. For more information, see the Javadoc in <i>webMethods CAF and My webMethods Server Java API Reference</i>, available from the “Software AG Documentation web site” under My webMethods Server in the corresponding webMethods Product Suite release number.</p>
<i>taskVersionNumber</i>	Integer Version number of the audited task.

Usage Notes

The integer values returned in the *operation* parameter represent the following operations:

```

0 = TASK_OPERATION_CREATED
1 = TASK_OPERATION_RESUME
2 = TASK_OPERATION_SUSPEND
3 = TASK_OPERATION_UPDATE
5 = TASK_OPERATION_ERROR
9 = TASK_OPERATION_DISTRIBUTION_RULE_FIRED
10 = TASK_OPERATION_TRIGGER_RULE_FIRED
11 = TASK_OPERATION_SCHEDULE_RULE_FIRED
12 = TASK_OPERATION_DELEGATION_RULE_FIRED
13 = TASK_OPERATION_REPLY
14 = TASK_OPERATION_REVERTED
16 = TASK_OPERATION_ACCEPTED
17 = TASK_OPERATION_UNACCEPTED

```

pub.task:TaskCallbackService

WmTaskClient. Specification that describes the inputs and outputs required for a callback service.

A callback service is a service (on the Integration Server) that the Task Engine invokes when a task ends. The callback service is assigned to a task instance when the task instance is queued. The callback service is invoked when the status of the task changes to "completed," "cancelled," "expired," or "error."

If the Task Engine cannot reach the Integration Server to invoke the service, it will, by default, attempt to re-invoke the service every minute for one hour.

If the callback service cannot be reached or if it throws an exception, the error will be recorded in the Task Engine logs on My webMethods Server. Additionally, the task instance itself will be placed in an "error" state and detailed information about the error will be available in the task audit information.

Input Parameters

<i>TaskData</i>	Document A document containing the business data associated with the task. TaskData does not have a specified structure. Its content varies according to the task.
<i>TaskCompletionInfo</i>	Document TaskCompletionInfo document, which contains information about the completion of the task See “pub.task:TaskCompletionInfo” on page 51 for a description of the fields in this document.
<i>CallbackData</i>	Document Information the task is configured to pass to the callback service. The data in <i>CallbackData</i> is given to the task when the task is initially queued. See

“[pub.task.taskclient:queueTask](#)” on page 28 for information about queuing a task.

Output Parameters

None.

pub.task:TaskComment

WmTaskClient. Document type representing a comment in a task.

Parameters

<i>attachments</i>	TaskAttachment[] An array of attachments to be enclosed inside the comment.
<i>id</i>	String Unique ID of the comment.
<i>displayText</i>	String Displayable representation of the text associated with the comment. If the text is not HTML, this returns the HTML formatted representation of plain text.
<i>text</i>	String Text associated with the comment.
<i>timeStamp</i>	Date The date and time of last modification of the comment.
<i>userDisplayName</i>	String Displayable representation of the name of user associated with the comment.
<i>userID</i>	String User name of the user associated with the comment.

pub.task:TaskCompletionInfo

WmTaskClient. Document type containing information about the completion of a task. The Task Engine uses this document type to return data about the completion of a task to a callback service.

Parameters

<i>taskID</i>	String The identifier of the task that ended.
<i>status</i>	String The state of the task. Will be one of the following:

- **completed.** Task has been successfully completed and is no longer available in the user's inbox queue. Completed tasks continue to appear in the Task Management Results list.
- **cancelled.** Task has been canceled and is no longer available in the user's inbox queue. Canceled tasks continue to appear in the Task Management Results list.
- **expired.** Task has expired and is no longer available in the user's inbox queue. Expired tasks continue to appear in the Task Management Results list.
- **error.** Task has failed or an unrecoverable error occurred during the processing of the task. This is often due to an incorrect task rule. The *errorCode* and *errorMessage* fields generally contain additional information about the error.

lastAcceptedBy **String** The user ID (on My webMethods Server) of the last user that accepted the task. The field is set to null if no user has accepted the task, or when *acceptedByList* is reset to null.

assignedToList **String List** The IDs of the principals (users, groups, or roles on My webMethods Server) to which this task has been assigned.

errorCode **String** A code that identifies the error condition that caused the task to end. This field is usually present when the value in *status* is "error." However, the Task Engine does not require an application to report an error code, so this field might be null even if the task ends with an error.

errorMessage **String** A message describing the error condition that caused the task to end. This field is usually present when the value in *status* is "error." However, the Task Engine does not require an application to report an error message, so this field might be null even if the task ends with an error.

pub.task:TaskFields

WmTaskClient. Document type that contains a list of name/value pairs describing the task fields returned by "[pub.task.taskclient:queueTask](#)" on page 28.

Parameters

name **String** Name of the task field.

value **Object** The value of the field contents for the specified field(s).

pub.task:TaskInfo

WmTaskClient. Document type that contains standard information about a task. You specify certain fields in TaskInfo when you queue a task using [pub.task.taskclient:queueTask](#).

After a task is queued, you can modify fields in TaskInfo using [pub.task.taskclient:updateTask](#).

Some fields that appear in TaskInfo can only be changed by the Task Engine. These fields are marked "read only" in the description below. If you attempt to assign a new value to a read-only field, the new value is ignored.

A task is not required to maintain information for every field in TaskInfo. Most fields are optional as noted in the description below.

Parameters

<i>acceptedByList</i>	String List Optional. The IDs (on My webMethods Server) of the users, groups, or roles that have accepted this task. Setting this list accepts the task for the specified users, groups, or roles.
<i>assignedToList</i>	String List Optional. The IDs of the principals (users, groups, or roles on My webMethods Server) to which this task is assigned. Setting this list assigns the task to the specified users, groups, or roles. The Task Engine uses this list to route the task to the appropriate users.
<i>auditContext</i>	String Read-only. Value of AuditContext from pub.prt.ProcessData document. This value appears only when a task is queued by business process
<i>collaborationProcessID</i>	String Read-only. The unique ID of the collaboration process flow created when a task is used for collaboration workflow.
<i>collaborationStepID</i>	String Read-only. The unique ID of the task step in the process flow when task is used for collaboration workflow.

<i>createdDate</i>	java.util.Date Read-only. Date and time when the task was queued.
<i>createdBy</i>	String Read-only. The user ID (on My webMethods Server) of the user that initially queued the task.
<i>customTaskID</i>	String Optional. An optional, application-defined identifier for the task. This ID is separate from the internal taskID that the Task Engine uses to identify tasks. The identifier in <i>customTaskID</i> is visible in the user interface and is also searchable using the pub.task.taskclient:searchTasks service. Note: Although <i>customTaskID</i> is meant to uniquely identify a task, the Task Engine does not enforce uniqueness of the value in this field. The application is responsible for assigning unique identifiers to <i>customTaskID</i> if they are needed.
<i>delegatedFromList</i>	String Optional. List of user IDs who delegated this task.
<i>delegatedToList</i>	String Optional. List of user IDs to whom the task was delegated
<i>description</i>	String Optional. A descriptive comment or remark associated with the task. This description appears in various places in the My webMethods user interface, such as on the Details View tab in My Inbox and in the Task Management Results list. Maximum length is 255 characters.
<i>errorCode</i>	String Optional. A code that identifies the error condition that caused the task to end. This field is usually present when the value in <i>status</i> is "error." However, the Task Engine does not require an application to report an error code, so this field might be null even if the task ends with an error.
<i>errorMessage</i>	String Optional. A message describing the error condition that caused the task to end. This field is usually present when the value in

status is "error." However, the Task Engine does not require an application to report an error message, so this field might be null even if the task ends with an error.

<i>expireDate</i>	<p>java.util.Date Optional. The date and time when the task expires. When <i>expireDate</i> is reached, the Task Engine switches the <i>status</i> value for the task to "expired."</p> <p>If an expire date is not specified, the task never expires.</p>
<i>isMandatory</i>	<p>Integer Optional. Specifies if the collaboration task is a mandatory task. A parent task can be marked as completed only when the status of all the mandatory child tasks is Completed, Error, Canceled, or Expired.</p>
<i>lastAcceptedBy</i>	<p>String Read-only. The user ID (on My webMethods Server) of the last user to accept the task. The field is set to null if no user has accepted the task, or when <i>acceptedByList</i> is reset to null.</p>
<i>lastAcceptedDate</i>	<p>java.util.Date Read-only. Date and time when the task was last accepted.</p>
<i>lastModifiedBy</i>	<p>String Read-only. The user ID (on My webMethods Server) of the user that last updated the task. If a process within the Task Engine was the last entity to modify the task (for example, if the Task Engine marked the task "expired"), this field will contain the name of the task rule associated with that process.</p>
<i>lastModifiedDate</i>	<p>java.util.Date Read-only. Date and time when the task was last updated.</p>
<i>name</i>	<p>String Optional. The name of the task. This name appears in various places in the My webMethods user interface, such as on the Details View tab in My Inbox and in the Task List Management Results list. A task does not require a name. If a name is assigned, it does not need to be unique. Maximum length is 255 characters.</p>

<i>parentTaskID</i>	String . Optional. The value of the parent task ID in the case when tasks are used for collaboration workflow. When queuing a new task, if a valid taskID is specified for "parentTaskID", the task to be queued is created as a child task of the parent task. For more information, see <i>webMethods BPM Task Development Help</i> .
<i>priority</i>	<p>String Optional. The priority of the task. Must contain one of the following values:</p> <pre> none low medium high critical </pre> <p>Values are case-sensitive.</p>
<i>processInstanceID</i>	String Read-only. The ProcessInstanceID value from the pub.prt:ProcessData document. This value only appears in TaskInfo if the task was queued by a business process.
<i>processIteration</i>	String Read-only. The ProcessIteration value from the pub.prt:ProcessData document. This value only appears in TaskInfo if the task was queued by a business process.
<i>processModelID</i>	String Read-only. The ProcessModelID value from the pub.prt:ProcessData document. This value only appears in TaskInfo if the task was queued by a business process.
<i>processModelVersion</i>	String Read-only. The ProcessModelVersion value from the pub.prt:ProcessData document. This value only appears in TaskInfo if the task was queued by a business process.
<i>status</i>	<p>String Optional. The state of the task. Must contain one of the following values:</p> <ul style="list-style-type: none"> ■ <code>new</code>. Task is new and not yet started. Immediately after the task starts, the status of the task changes from <code>new</code> to <code>active</code>. ■ <code>scheduled</code>. Task is scheduled to start at the time specified in <i>taskScheduleDate</i>. When the

task starts at the scheduled time, the status of the task changes from `scheduled` to `active`.

- `active`. Task is active and is available in the user's inbox queue. Only active tasks can be modified by the user.
- `suspended`. Task is suspended and is not available in the user's inbox queue. Suspended tasks continue to appear in the Task Management Results list.
- `completed`. Task has been successfully completed and is no longer available in the user's inbox queue. Completed tasks continue to appear in the Task Management Results list.
- `cancelled`. Task has been canceled and is no longer available in the user's inbox queue. Canceled tasks continue to appear in the Task Management Results list.
- `expired`. Task has expired and is no longer available in the user's inbox queue. Expired tasks continue to appear in the Task Management Results list.
- `error`. Task has failed or an unrecoverable error occurred during the processing of the task. This is often due to an incorrect task rule. The `errorCode` and `errorMessage` fields generally contain additional information about the error.

stepID

String Read-only. The StepID value from the `pub.prt:ProcessData` document. This value only appears in `TaskInfo` if the task was queued by a business process.

stepIteration

String Read-only. The TryCount value from the `pub.prt:ProcessData` document. This value only appears in `TaskInfo` if the task was queued by a business process.

taskScheduleDate

Date Specifies the date and time the task should start. The status of the task is based on the specified value.

- If you specify a date that is later than the current date, the task status will be `scheduled`.

- If you do not specify a date, the task will be `active`.

When the task starts or queues at the scheduled time, the global rule for task schedule changes the status of the task from `scheduled` to `active`.

taskTypeID

String Read-only. The task type identifier (GUID) for this task instance.

taskUrl

String Read only. The relative URL for the task on My webMethods Server. This URL can be used to open the task instance in a browser.

taskVersionNumber

Integer Optional. This is a current version number of the task record; it is automatically incremented for each task update action.

This property can be used with [pub.task.taskclient:updateTask](#). If a value different than 0 is passed when executing [pub.task.taskclient:updateTask](#), then the Task Engine verifies this value against the current value of the task instance and generate the exception "Task is Out of Date" if they mismatch. In other words, if the task record has been updated (and **taskVersionNumber** incremented) since the client of the API has been holding the task record, the Task Engine generates the exception.

pub.task:TaskQueueInfo

WmTaskClient. Document type used by [pub.task.taskclient:queueTask](#) to set basic values in TaskInfo.

The fields in this document type constitute a subset of the fields in [pub.task:TaskInfo](#).

Parameters

name

String Optional. The name of the task. This name appears in various places in the My webMethods user interface, such as on the **Details View** tab in My Inbox and in the Task Management Results list.

A task does not require a name. If a name is assigned, it does not need to be unique. Maximum length is 255 characters.

customTaskID **String** Optional. An application-defined identifier for the task. This ID is separate from the internal taskID that the Task Engine automatically generates and assigns to a task instance. The identifier in *customTaskID* is visible in the user interface and is also searchable using the [pub.task.taskclient:searchTasks](#) service. For example, if a task relates to an order approval process, you might assign the order number to the *customTaskID*.

Note: Although *customTaskID* is meant to uniquely identify a task, the Task Engine does not enforce uniqueness of the value in this field. The application is responsible for assigning unique identifiers to *customTaskID* if they are needed.

assignedTaskList **String List** Optional. The IDs of the principals (users or roles on My webMethods Server) to which this task is assigned. Setting this list assigns the task to the specified users or roles. The Task Engine uses this list to route the task to the appropriate users.

priority **String** Optional. The priority of the task. Must be one of the following values (values are case-sensitive):

none low medium high critical

expireDate **java.util.Date** Optional. The date and time when the task instance expires. When *expireDate* is reached, the Task Engine automatically switches the *status* value for the task to "expired."

If an expire date is not specified, the task never expires.

parentTaskID **String**. The value of the parent task ID in the case when tasks are used for collaboration workflow. When queuing a new task, if a valid taskID is specified for "parentTaskID", the task to be queued is created as a child task of the parent task. For more information, see *webMethods BPM Task Development Help*.

isMandatory **Integer** Optional. Specifies if the collaboration task is a mandatory task. A parent task can be marked as completed only when the status of all the mandatory child tasks is Completed, Error, Canceled, or Expired.

taskScheduleDate **Date** Specifies the date and time the task should start. The status of the task changes based on the value specified for *taskScheduleDate*.

- If you specify a date that is later than the current date, the task status will be `Scheduled`.
- If you do not specify a date, the task will be `active`.

Note: Configure a scheduler rule in Designer so that the status of a task changes from `Scheduled` to `active` when the task starts or queues at the scheduled time.

pub.task:TaskSearchQuery

WmTaskClient. Document type used as input to the [pub.task.taskclient:searchTasks](#), [pub.task.taskclient:searchTasksHPSTRA](#), [pub.task.taskclient:searchTasksFields](#) and [pub.task.taskclient:searchTasksFieldsHPSTRA](#) services. This service applies only to tasks that use standard business data. The parameters defined here are also present in [pub.task.TaskSearchQueryV2](#).

Parameters

MaxResults

String Optional.

- If the *TaskSearchQuery* parameter is not specified in the calling service, the calling service returns all the tasks.
- If the *TaskSearchQuery* parameter is defined in the calling service, but the *MaxResults* parameter is not defined in *TaskSearchQuery*, only 200 results are returned by the calling service.
- If the *TaskSearchQuery* parameter is defined in the calling service, and the *MaxResults* parameter is defined in *TaskSearchQuery*, the calling service returns the number of records specified by *MaxResults* (type 0 to return all tasks).

Note: This element does not apply to tasks with an indexed search provider. In this case, the specified value is ignored.

doNotShowAcceptedByOthers

Boolean Optional. Exclude tasks that are accepted by a user other than the user ID used in the search. Default is false.

showNonActiveTasks

Boolean Optional. This field can be used when executing queries to fetch tasks from a user's inbox. This parameter is meant to override the

default behavior of the *searchUserTasks* parameter wherever it is used and allow the return of non-active tasks.

Set to:

- `true` to return non-active tasks.
- `false` (default) to return active tasks only.

For example, by specifying *searchUserTasks* =`true` and *showNonActiveTasks* =`true`, the query returns all non-active tasks from a user's inbox (that is, those tasks already completed by user).

The *showNonActiveTasks* parameter has no impact on non-user task searches. To construct a non-user task search to return only active tasks, you must set query terms of *searchUserTasks* =`false` and *status* =`active`.

searchEngineType

String Optional. Specifies the search engine to use when searching for tasks. The default values are:

- `db` - when using services that search task data, stored in the indexed table of the My webMethods Server database.
- `hpstra` - when using services that search in HPSTRA-enabled task data.

You can also specify the name of any custom engine that you develop for task searches and register as an OSGi service in the My webMethods Server runtime. For more information and examples about adding custom task search engines, see the Software AG TECHcommunity website.

Terms

Document List An array of `TaskSearchQueryTerm`, in which each member of the array specifies query criteria for the search. See [pub.task:TaskSearchQueryTerm](#) for a description of the fields in `TaskSearchQueryTerm`.

When a search is performed, the criteria specified by the `TaskSearchQueryTerm` documents in *Terms* are combined using the logical "AND" operator. Only tasks that satisfy all `TaskSearchQueryTerm` documents in the *Terms* array are returned in the result set.

Usage Notes

The *showNonActiveTasks* parameter overrides the default behavior of the *searchUserTasks* parameter in any service where that parameter occurs, such as:

- [pub.task.taskclient:countTasksIndexed](#)
- [pub.task.taskclient:searchTasks](#)
- [pub.task.taskclient:searchTasksFields](#)
- [pub.task.taskclient:searchTasksIndexed](#)
- [GET Task Instance Information](#)
- [GET Tasks with a Simple Search Request](#)
- [POST a Complex Task Search Request](#)

pub.task:TaskSearchQueryTerm

WmTaskClient. Document type that you use to specify search criteria for the [pub.task:TaskSearchQuery](#) document type.

You use this document to specify:

1. The *Fields* that you want the service to search.
2. The *Operator* that defines the type of comparison that you want the service to make.
3. The *Value* that you want the field's contents to be compared to.

A task is included in the result set if the comparison evaluates to true in *any* of the fields you have specified (that is, the service performs a logical OR when asked to evaluate multiple fields).

Parameters

Fields

String[] A list of fields that are to be searched. There are two different methods for specifying the field name, depending on whether you are working with standard task fields or indexed task fields. For more information about indexed fields, see the *webMethods BPM Task Development Help*.

Working with Standard Task Fields

You can search standard task fields in TaskInfo using these names:

- `acceptedByList`
- `assignedToList`
- `auditContext`

- collaborationProcessID
- collaborationStepID
- createdBy
- createdDate
- customTaskID
- description
- expireDate
- lastModifiedDate
- lastModifiedBy
- lastAcceptedBy
- name
- priority
- taskID
- taskTypeID
- parentTaskID
- processInstanceID
- processModelID
- processModelVersion
- status
- stepID
- stepIteration

To search fields in TaskData (the business data associated with the task), you must specify the field using a binding expression in the following format:

```
{currentTask.taskData.pathToDataElement }
```

For example, the following binding expression:

```
{currentTask.taskData.travelReservation.reservationNo}
```

would search the "reservationNo" field within the "travelReservation" document in TaskData.

Working with Indexed Fields

To search tasks with indexed fields, you must pass in the database name of the indexed field along with the desired operator and

value. The indexed field name is set in the Designer task editor. To determine the database index field name:

1. Open the task that contains the indexed data field.
2. On the **Business Data** tab, select the field you want to work with.
3. Click **Edit**.
4. In the Edit Business Data dialog box, obtain the indexed field name from the **Name** field in the **Database field settings** area.

For example, to search the database field "ZipCode" for matches to the value "90210":

```
SearchTerm.fields="ZipCode"
SearchTerm.operator="="
SearchTerm.value="90210"
```

This service returns the values for the specified indexed fields only. For more information about indexed fields, see the *webMethods BPM Task Development Help*.

Operator

String The comparison that the service will make when evaluating the contents of *Value* against the contents of the specified field(s).

Operator must be one of the following:

- = Is equal to.
- <> Is not equal to.
- < Is less than. Valid only for numbers or date fields.
- > Is greater than. Valid only for numbers or date fields.
- <= Is less than or equal to. Valid only for numbers or date fields.

- >= Is greater than or equal to. For numbers or date fields only.
- contains Compares *Value* to the individual elements in a String List. The comparison evaluates to true if any element in the list matches *Value*. Valid only with string list fields.

For example, if you have a string list field with three elements, this operator evaluates to true if any one of the three elements contains a string that exactly matches the value in *Value*.

- in Matches if the single string value of the task field can be found inside a string list of values passed in the term. Such as: *TaskFieldValue* in (*value1*, *value2*, *value3*). Valid for all types.
- is empty Matches if the value of the field is either null or an empty string ("").
- is not empty Matches if the value of the field is not null and not an empty string ("").

- `is null` Matches if the current term field is null.
- `is not null` Matches if current term field is not null.
- `like` Matches the pattern string specified in *Value*. See *Value* for pattern string information. Valid only with string fields.
- `not in` Matches if the single string value of the task field cannot be found in a string list of values passed in the term. Such as: *TaskFieldValue* not in (*value1*, *value2*, *value3*). Valid for all types.
- `not like` Matches for all instances where the pattern string does not match the text specified in *Value*. See *Value* for information about pattern strings; valid only with string fields.
- `is not empty` Matches if the value of the field is not null and not an empty string ("").
- `is null` Matches if the current term field is null.
- `is not null` Matches if current term field is not null.
- `like` Matches the pattern string specified in *Value*. See *Value* for pattern string information. Valid only with string fields.
- `not in` Matches if the single string value of the task field cannot be found in a string list of values passed in the term. Such as: *TaskFieldValue* not in (*value1*, *value2*, *value3*). Valid for all types.
- `not like` Matches for all instances where the pattern string does not match the text specified in *Value*. See *Value* for information about pattern strings; valid only with string fields.

Value

Object The value to which the contents of the field specified in *Field* will be compared.

Value can also specify a pattern string that can include the * wildcard character. For example:

- A *Value* of `abc` would return tasks whose field contents contain only the character sequence `abc`.
- A *Value* of `abc*` would return tasks whose field contents start with the character sequence `abc`.
- A *Value* of `*abc` would return tasks whose field contents end with the character sequence `abc`.
- A *Value* of `*abc*` would return tasks whose field contents include the character sequence `abc` anywhere within the field.

Note: To treat the * wildcard character as a literal, type `\` before the * wildcard character. A *Value* of `abc*` would return

tasks whose field contents contain only the character sequence `*abc`.

Value is case-insensitive by default. That is, `abc` will match `ABC` or `Abc`. However, for databases such as Oracle that are case-sensitive by default, you can specify the *Case insensitive* parameter to perform a case-insensitive search.

Case insensitive

Boolean Disregards case for the *Value* to which the contents of the field specified in *Field* will be compared. You specify the parameter to perform index searches on databases that are not case-insensitive by default. Values are:

- `true` - Case is disregarded.
- `false` - Case is not disregarded. This is the default value.

Usage Notes

When specifying date-based Standard Text Fields (for example, `expireDate`), supply the date information as a date object.

pub.task.TaskSearchQueryV2

WmTaskClient. Document type that you use to specify the search criteria for the [pub.task.taskclient:searchTasksIndexed](#) and [pub.task.taskclient:searchTasksIndexedHPSTRA](#) services.

The `pub.task.TaskSearchQueryV2` document type includes the same fields as [pub.task:TaskSearchQuery](#) plus the additional fields listed in the table below to support queries on indexed business data.

Parameters	Description
<i>fromIndex</i>	Integer Specifies the starting index of the page of search results. Zero represents the first index of the result set.
<i>toIndex</i>	Integer Indicates the end of the search results indexed on the page. The <i>toIndex</i> must be greater than the <i>fromIndex</i> and less than or equal to the total result count
<i>searchEngineType</i>	String Optional. Specifies the search engine to use when searching for tasks. The default values are: <ul style="list-style-type: none"> ■ <code>db</code> - for services that search task data, stored in the indexed table of the My webMethods Server database. ■ <code>hpstra</code> - for services that search in HPSTRA-enabled task data.

Parameters	Description
	<p>You can also specify the name of any custom engine that you develop for task searches and register as an OSGi service in the My webMethods Server runtime. For more information and examples about adding custom task search engines, see the Software AG TECHcommunity website.</p>
<i>sortBy</i>	<p>String Optional. Specifies the comma-separated list of field names used to sort the search results. The first entry is the primary sort key, second entry is the secondary sort key, matching the results to the order of the sort key sequence.</p>
	<p>When specifying indexed field names, you must provide the database index field name. To determine the value of the database index field name, see pub.task:TaskSearchQueryTerm.</p>
<i>sortOrder</i>	<p>String Optional. Specifies the comma-separated list of values ascending (ASC) or descending (DESC) that match the <i>sortBy</i> list of fields. <i>sortOrder</i> indicates the sort order used to list the fields, ascending or descending order.</p>
	<p>Set <i>sortBy</i> to:</p>
	<ul style="list-style-type: none">■ ASC to display the results in ascending order.■ DESC to display the results in descending order.
	<p>The default <i>sortOrder</i> is ascending (ASC).</p>

2 Using the Task Engine RESTful Web Services in Integration Server

■ Overview	71
■ Summary of REST Resources and Requests	71
■ DELETE a Task Instance	73
■ GET Task Instance Information	74
■ POST a New Task Instance	77
■ PUT Information in a Task Instance	78
■ GET Tasks with a Simple Search Request	80
■ GET HPSTRA-enabled Tasks with a Simple Search Request	81
■ POST a Complex Task Search Request	83
■ POST a Complex Search Request for HPSTRA-enabled Tasks	89
■ GET Task Audit Information	95
■ DELETE Audit Entries to Rollback Task	96
■ GET or POST a Form Flow Notification	96
■ DELETE Task Comment	98
■ GET Task Comments	98
■ PUT Updates into a Task Comment	99
■ POST Task Comment	100
■ DELETE Tags From Task	101
■ GET Tags By Task	101
■ GET Task Expert List	102
■ POST Tags To Task	102
■ PUT Task Tags	103
■ DELETE Task Attachment	103
■ GET Task Attachments	104
■ PUT Task Attachment	105

■ POST Task Attachment	105
------------------------------	-----

Overview

The WmTaskClient package contains RESTful web services that you can use in conjunction with the REST service capabilities of Integration Server. For more information about REST services in general, and about implementing the REST services in Integration Server, see the Software AG PDF publication *REST Developer's Guide*. Prior to implementing these RESTful web services, you should have a working knowledge of JavaScript Object Notation (JSON).

For more information about working with Integration Server services in general, see the *webMethods Service Development Help* in Designer.

The Task Engine RESTful web services share data structures (documents) with the Task Engine built-in services. Information about the data structures referred to in the RESTful web service reference topics can be found in the [“Summary of Data Structures” on page 45](#).

In addition to the parameters shown in the RESTful web service reference topics in this section, the Task Engine RESTful web services can also be implemented with parameters from any of the underlying built-in services. These parameters can be specified as key-value pairs in the RESTful web service URL, or as part of an associated JSON document.

See Also

[on page 71 “Summary of REST Resources and Requests”](#)

[on page 69 “Using the Task Engine RESTful Web Services in Integration Server”](#)

Summary of REST Resources and Requests

The webMethods Task Engine API provides REST requests that can be applied to the following Task Engine resources:

Task Instances

Request	Description
DELETE a Task Instance	Deletes an instance of a task on the Task Engine connected to the Integration Server.
GET Task Instance Information	Returns the TaskInfo object, and optionally the TaskData object, for one task, or for a list of the TaskData objects (and TaskInfo objects, if specified) for all the tasks the user has permission to access.

Request	Description
POST a New Task Instance	Creates a new task instance and adds it to the Task Engine queue.
PUT Information in a Task Instance	Modifies a task instance on the Task Engine connected to the Integration Server.
GET Tasks with a Simple Search Request	Retrieves tasks that match simple search criteria.
POST a Complex Task Search Request	Retrieves tasks that are specified with more complex search criteria.

Task Audit Information

GET Task Audit Information	Returns the audit log of all of the operations performed on a task, or a specific audit entry.
DELETE Audit Entries to Rollback Task	Enables you to roll back the task to any available audit point in the task's audit history.
GET or POST a Form Flow Notification	Notifies a waiting step in a task workflow that the workflow is completed.

Task Comments

Request	Description
DELETE Task Comment	Deletes the specified task comment from a task instance as well as all associated attachments.
GET Task Comments	Returns either a specific comment or all the comments in a task instance including associated attachments.
PUT Updates into a Task Comment	Updates a comment in a task instance, including attachments associated with the comment.
POST Task Comment	Adds a comment to a task instance. Attachments can be included.

Task Tags

Request	Description
DELETE Tags From Task	Deletes all tags associated with a <i>taskID</i> .
GET Tags By Task	Returns a list of tags associated with a <i>taskID</i> .
GET Task Expert List	Returns a list of experts associated with the specified tags.
POST Tags To Task	Adds specified tags to a <i>taskID</i> .
PUT Task Tags	Updates tags associated with a <i>taskID</i> .

Task Attachments

Request	Description
DELETE Task Attachment	Deletes the specified task attachment from a task instance.
GET Task Attachments	Returns all the attachments in a task instance.
PUT Task Attachment	Updates the specified task attachment in a task instance.
POST Task Attachment	Adds the specified attachment to a task instance.

DELETE a Task Instance

This REST request deletes an instance of a task on the Task Engine connected to the Integration Server. The underlying built-in service is `pub.task.taskclient:deleteTask`.

The HTTP request is formed as follows:

```
DELETE /rest/pub/opentask/taskID [?user=value]
```

or

```
DELETE /rest/pub/opentask?taskID=taskID [&user=value]
```

Input Parameters

<i>taskID</i>	String The ID that identifies the task that you want to delete. This can be either passed in the URL as shown above, or as an input parameter.
<i>user</i>	String The user ID of the My webMethods Server user on whose behalf this service will execute. If <i>user</i> is not specified, the user ID associated with the WmTaskClient package is used. For information about configuring this user ID, see <i>webMethods Task Engine User's Guide</i> .

Output Parameters

None.

Usage Notes

To delete a task with this request, you must supply the ID of the task that you want to delete. To obtain this ID, use [“GET Task Instance Information” on page 74](#) in list task mode or one of the `opentasksearch` requests to locate the task and then extract the task ID from the Task document that it returns.

To delete a task successfully, the user ID in *user* must have permission to access the specified task. If the supplied user ID does not have access to the task, or if the specified task does not exist, an exception will be thrown.

GET Task Instance Information

This REST request returns the `pub.task:TaskInfo` object, and optionally the `TaskData` object, for one task, or for a list of the `TaskData` objects (and `TaskInfo` objects, if specified) for all the tasks the user has permission to access. The underlying built-in services are:

- `pub.task.taskclient:getTask`
- `pub.task.taskclient:searchTasks`

This service can be used in the following modes:

- List tasks mode
- Get a specific task mode

The service returns a specific task if the *taskID* parameter is supplied. Otherwise, list of tasks is returned if *taskID* is not specified. The `TaskInfo` object contains standard information about a task, including its status, expiration date, and the list of users to which it is assigned. The `TaskData` object contains the business data that is associated with

the task. `TaskData` does not have a specified structure. Its content varies according to the task.

List Tasks Mode

To use the request to retrieve information about a list of tasks, the HTTP request is formed as follows:

```
GET /rest/pub/opentask[?param1=value] [&param2=value] [&param2=value]
```

Input Parameters

<i>includeTaskData</i>	<p>String Optional. Specifies whether you want to retrieve the <code>TaskData</code> document as well as the <code>TaskInfo</code> document. Set to:</p> <ul style="list-style-type: none"> ■ <code>true</code> to retrieve <code>TaskData</code> in addition to <code>TaskInfo</code>. ■ <code>false</code> (default) to get <code>TaskInfo</code> only.
<i>searchUserTasks</i>	<p>String Optional. Specifies whether the service searches all tasks or only the user's inbox. Set to:</p> <ul style="list-style-type: none"> ■ <code>true</code> to search only the inbox for user. ■ <code>false</code> (default) to search all tasks to which user has access.
<i>user</i>	<p>String Optional. The user ID of the My webMethods Server user on whose behalf this service will execute. If <i>user</i> is not specified, the user ID associated with the <code>WmTaskClient</code> package is used. For information about configuring this user ID, see <i>webMethods Task Engine User's Guide</i>. Note: Only tasks to which <i>user</i> has access are searched.</p>

Output Parameters

<i>Tasks</i>	<p>Document list The result set containing the tasks that match the search criteria. Each document in the result set has the following structure:</p> <ul style="list-style-type: none"> ■ <i>TaskID</i> String The identifier assigned to the task instance. ■ <i>TaskInfo</i> Document The <code>TaskInfo</code> document containing standard information about the task. See “pub.task:TaskInfo” on page 53 for a description of the fields in this document. ■ <i>TaskData</i> Document The <code>TaskData</code> document containing the business data associated with the task. Returned only if <i>includeTaskData</i> is set to <code>true</code>.
--------------	---

Get a Specific Task Mode

To use the request to retrieve information about a specific task, the HTTP request is formed as follows:

```
GET /rest/pub/opentask/taskID [ ?param1=value ] [ &param2=value ]
```

or

```
GET /rest/pub/opentask?taskID=taskID [ &param1=value ] [ &param2=value ]
```

Input Parameters

<i>includeTaskData</i>	<p>String Optional. Specifies whether you want to retrieve the TaskData document as well as the TaskInfo document. Set to:</p> <ul style="list-style-type: none"> ■ <code>true</code> to retrieve TaskData in addition to TaskInfo. ■ <code>false</code> (default) to get TaskInfo only.
<i>taskID</i>	<p>String The ID that identifies the task that you want to retrieve. This can be either passed in the URL as shown above, or as an input parameter.</p>
<i>user</i>	<p>String Optional. The user ID of the My webMethods Server user on whose behalf this service will execute. If <i>user</i> is not specified, the user ID associated with the WmTaskClient package is used. For information about configuring this user ID, see <i>webMethods Task Engine User's Guide</i>.</p>

Output Parameters

<i>TaskID</i>	<p>String The ID that identifies the task.</p>
<i>TaskInfo</i>	<p>Document The TaskInfo document containing standard information about the task. See “pub.task:TaskInfo” on page 53 for a description of the fields in this document.</p>
<i>TaskData</i>	<p>Document A document containing the business data associated with the task. TaskData does not have a specified structure. Its content varies according to the task. TaskData is returned only if <i>includeTaskData</i> is set to <code>true</code>.</p>

Usage Notes

To get task instance information with this request, you must supply the ID of the task instance that you want to retrieve. To obtain this ID, use the request in list tasks mode or

one of the `opentasksearch` requests to locate the task and then extract the task ID from the Task document that it returns. To retrieve a task successfully, the user ID specified in user must have permission to access that task. If the supplied ID does not have access to the task, or if the specified task does not exist, an exception is thrown.

POST a New Task Instance

This REST request creates a new task instance and adds it to the Task Engine queue. The underlying built-in service is `pub.task.taskclient:queueTask`.

Use this HTTP request and include optional parameters in the request body:

```
POST /rest/pub/opentask
```

Input Parameters

<i>CallbackData</i>	Document Optional. Any data that you want the task to pass back to the callback service.
<i>CallbackServiceName</i>	String Optional. The name of a service on this Integration Server that will execute when the task ends (that is, when the status of the task changes to "completed," "cancelled," "expired," or "error"). Example: <code>customer.accounts.closeAccount</code> The signature of the callback service must match the specification described in “pub.task:TaskCallbackService” on page 50 .
<i>ruleSet</i>	String Optional. The name of the rule set to be applied for this task. This must match one of the rule set names defined for the task in the task editor in Software AG Designer. When a rule set is specified, only the assignments and events for the specified rule set will be executed for this task instance.
<i>taskData</i> or <i>TaskData</i>	Document Optional. The <code>TaskData</code> document that contains the business data that you want to associate with the task. <code>TaskData</code> does not have a specified structure. The content varies according to the task.
<i>taskQueueInfo</i> or <i>TaskQueueInfo</i>	Document Optional. A <code>TaskQueueInfo</code> document containing basic information about the task (for example, name, priority, list of users to which the task is assigned). See “pub.task:TaskQueueInfo” on page 58 for a description of the fields in this document.
<i>taskTypeID</i>	String Specifies the type of task that you want to queue. Each task type that is deployed to the Task Engine has a unique ID.

This ID is assigned by the developer when he or she creates a task application using Software AG Designer. If you do not know the ID for the task type that you want to queue, contact the administrator of the Task Engine to which Integration Server is connected. An administrator can obtain a list of the task types deployed on the Task Engine by viewing the **Task Configuration** panel on the Task Engine Administration page in My webMethods.

user **String** Optional. The user ID of the My webMethods Server user on whose behalf this service will execute. If *user* is not specified, the user ID associated with the WmTaskClient package is used. For information about configuring this user ID, see *webMethods Task Engine User's Guide*.

Output Parameters

TaskID **String** The identifier that the Task Engine assigns to the task instance.

Usage Notes

The user ID specified in *user* must have permission to queue tasks on the Task Engine. If the supplied ID does not have this permission, an exception is generated.

PUT Information in a Task Instance

This REST request modifies a task instance on the Task Engine connected to the Integration Server. The underlying built-in service is `pub.task.taskclient:updateTask`.

You use this request to change information in the task's `TaskInfo` and `TaskData` documents. This service applies only to tasks with `Active`, `Error`, and `Suspended` status. For information about updating tasks with other statuses, see Usage Notes. To end a task instance, you use this service to change the status field in the task's `TaskInfo` document to "closed" or "cancelled" as appropriate.

To use the request to update a task, the HTTP request is formed as follows:

```
PUT /rest/pub/opentask/taskID [ ?param1=value ] [ &param2=value ] . . . [ &param4=value ]
```

or

```
PUT /rest/pub/opentask?taskID=taskID [ &param1=value ] . . . [ &param4=value ]
```

Input Parameters

<i>taskID</i>	String The ID that identifies the task that you want to modify. This can be either passed in the URL as shown above, or as an input parameter.
<i>taskInfo</i> or <i>TaskInfo</i>	Document The TaskInfo document that specifies the changes you want to make to the standard information maintained for the task. See “ pub.task:TaskInfo ” on page 53 for a description of the fields in this document.
<i>taskData</i> or <i>TaskData</i>	Document The TaskData document that contains the business data that you want to associate with the task. TaskData does not have a specified structure. The content varies according to the task
<i>user</i>	String Optional. The user ID of the My webMethods Server user on whose behalf this service will execute. If <i>user</i> is not specified, the user ID associated with the WmTaskClient package is used. For information about configuring this user ID, see <i>webMethods Task Engine User's Guide</i> .

Output Parameters

None.

Usage Notes

When you use this service to update information in the task's TaskInfo document, the Task Engine modifies only the fields that have assigned values in TaskInfo. All other fields in the target task retain their existing values. For example, to change just the name assigned to a task, set only the name field in the TaskInfo document that you pass to this service.

When you use this service to update information in the task's TaskData document, the TaskData document that you pass to the service replaces the task's existing TaskData document in its entirety.

When updating task statuses with this service, the following limitations apply:

- If the current status of the task is Error, you can change it only to Completed.
- If the current status of the task is Suspended, you cannot set it to Completed or Expired.

This service applies only to tasks with status Active, Error, and Suspended. To enable the service to update tasks with other statuses, set the additional JVM property - `Dupdate.completed.task` to `true`, as described in *webMethods Task Engine User's Guide*.

Tasks with status Error that belong to a process instance cannot be updated, regardless of Task Engine optional settings.

GET Tasks with a Simple Search Request

This REST request retrieves tasks that match simple search criteria. The underlying built-in service is `pub.task.taskclient:searchTasks`.

The HTTP request will be formed as follows:

```
GET /rest/pub/opentasksearch[?param1=value] [&param2=value] . . . [&param6=value]
```

Input Parameters

<i>includeTaskData</i>	<p>String Optional. Specifies whether the service will return the <code>TaskData</code> document as well as the <code>pub.task:TaskInfo</code> document for each task matching the search criteria. Set to:</p> <ul style="list-style-type: none"> ■ <code>true</code> to include the <code>TaskData</code> document in the result set. ■ <code>false</code> (default) to include only the <code>TaskInfo</code> document in the result set.
<i>maxRes</i>	<p>String Optional. The maximum number of tasks to return in the result set. If <i>maxRes</i> is not specified, all results in the result set are returned. Note: This element does not apply to tasks with an indexed search provider. In this case, the specified value is ignored.</p>
<i>nonActive</i>	<p>String Optional. This field can be used when executing queries to fetch tasks from a user's inbox. Set to:</p> <ul style="list-style-type: none"> ■ <code>true</code> to include non-active tasks from a user's inbox (that is, those tasks already completed by user). ■ <code>false</code> (default) to return only active tasks.
<i>notOthers</i>	<p>String Optional. Exclude tasks which are accepted by a user other than the user ID used in the search. Set to:</p> <ul style="list-style-type: none"> ■ <code>true</code> to exclude tasks that are accepted by a user other than <i>user</i>. ■ <code>false</code> (default) to search all tasks to which <i>user</i> has access.
<i>searchUserTasks</i>	<p>String Optional. Specifies whether the service searches all tasks or just the user's inbox. Set to:</p> <ul style="list-style-type: none"> ■ <code>true</code> to search only the inbox for user.

- `false` (default) to search all tasks to which user has access.

user

String Optional. The user ID of the My webMethods Server user on whose behalf this service will execute. If *user* is not specified, the user ID associated with the WmTaskClient package is used. For information about configuring this user ID, see *webMethods Task Engine User's Guide*.

Note: Only tasks to which *user* has access are searched.

Output Parameters

Tasks

Document list The result set containing the tasks that match the search criteria in [pub.task:TaskSearchQuery](#). Each document in the result set has the following structure:

- *TaskID* **String** The identifier assigned to the task instance.
- *TaskInfo* **Document** The TaskInfo document containing standard information about the task. See "[pub.task:TaskInfo](#)" on [page 53](#) for a description of the fields in this document.
- *TaskData* **Document** The TaskData document containing the business data associated with the task. Returned only if *includeTaskData* is set to `true`.

Usage Notes

This request can be used for simple listing of tasks based on parameters. To run a more complex query, see "[POST a Complex Task Search Request](#)" on [page 83](#), which allows submitting various search queries.

GET HPSTRA-enabled Tasks with a Simple Search Request

This REST request retrieves HPSTRA-enabled tasks that match simple search criteria. The underlying built-in service is `pub.task.taskclient:searchTasksHPSTRA`.

The HTTP request will be formed as follows:

```
GET /rest/pub/opentasksearchhpstra[?param1=value] [&param2=value] . . .
[&param6=value]
```

Input Parameters

<i>includeTaskData</i>	<p>String Optional. Specifies whether the service will return the TaskData document as well as the pub.task:TaskInfo document for each task matching the search criteria. Set to:</p> <ul style="list-style-type: none"> ■ <code>true</code> to include the TaskData document in the result set. ■ <code>false</code> (default) to include only the TaskInfo document in the result set.
<i>maxRes</i>	<p>String Optional. The maximum number of tasks to return in the result set. If <i>maxRes</i> is not specified, all results in the result set are returned. Note: This element does not apply to tasks with an indexed search provider. In this case, the specified value is ignored.</p>
<i>nonActive</i>	<p>String Optional. This field can be used when executing queries to fetch tasks from a user's inbox. Set to:</p> <ul style="list-style-type: none"> ■ <code>true</code> to include non-active tasks from a user's inbox (that is, those tasks already completed by user). ■ <code>false</code> (default) to return only active tasks.
<i>notOthers</i>	<p>String Optional. Exclude tasks which are accepted by a user other than the user ID used in the search. Set to:</p> <ul style="list-style-type: none"> ■ <code>true</code> to exclude tasks that are accepted by a user other than <i>user</i>. ■ <code>false</code> (default) to search all tasks to which <i>user</i> has access.
<i>searchUserTasks</i>	<p>String Optional. Specifies whether the service searches all tasks or just the user's inbox. Set to:</p> <ul style="list-style-type: none"> ■ <code>true</code> to search only the inbox for user. ■ <code>false</code> (default) to search all tasks to which user has access.
<i>user</i>	<p>String Optional. The user ID of the My webMethods Server user on whose behalf this service will execute. If <i>user</i> is not specified, the user ID associated with the WmTaskClient package is used. For information about configuring this user ID, see <i>webMethods Task Engine User's Guide</i>.</p>

Note: Only tasks to which *user* has access are searched.

Output Parameters

Tasks

Document list The result set containing the tasks that match the search criteria in [pub.task:TaskSearchQuery](#). Each document in the result set has the following structure:

- **TaskID String** The identifier assigned to the task instance.
- **TaskInfo Document** The TaskInfo document containing standard information about the task. See “[pub.task:TaskInfo](#)” on [page 53](#) for a description of the fields in this document.
- **TaskData Document** The TaskData document containing the business data associated with the task. Returned only if *includeTaskData* is set to `true`.

Usage Notes

This request can be used for simple listing of HPSTRA-enabled tasks based on parameters. To run a more complex query, see “[POST a Complex Search Request for HPSTRA-enabled Tasks](#)” on [page 89](#), which allows submitting various search queries.

POST a Complex Task Search Request

This REST request retrieves tasks that are specified with more complex search criteria. The underlying built-in services are:

- `pub.task.taskclient:searchTasks`
- `pub.task.taskclient:searchTasksFields`
- `pub.task.taskclient:searchTasksIndexed`
- `pub.task.taskclient:countTasksIndexed`

This request can be used in the following four modes:

- **searchTasks mode.** Retrieve tasks that match specified search criteria, when working with tasks that use standard business data.
- **searchTasksFields mode.** Return a collection of named fields requested in each service that matches specified search criteria.
- **searchTasksIndexed mode.** Return a subset of tasks that match the specified search criteria, when working with indexed business data fields.
- **countTasksIndexed mode.** Return the total count of tasks that match the specified search criteria. This service applies only to tasks with indexed business data.

For additional information on these modes, see [Usage Notes](#).

searchTasks Mode

Use this mode to retrieve tasks that match specified search criteria. This request applies only to tasks that use standard business data.

Use this HTTP request and include optional parameters in the request body:

```
POST /rest/pub/opentasksearch
```

Input Parameters

<i>includeTaskData</i>	<p>String Optional. Specifies whether the service will return the <code>TaskData</code> document as well as the <code>pub.task:TaskInfo</code> document for each task matching the search criteria. Set to:</p> <ul style="list-style-type: none"> ■ <code>true</code> to include the <code>TaskData</code> document in the result set. ■ <code>false</code> (default) to include only the <code>TaskInfo</code> document in the result set.
<i>searchUserTasks</i>	<p>String Optional. Specifies whether the service searches all tasks or only the user's inbox. Set to:</p> <ul style="list-style-type: none"> ■ <code>true</code> to search only the inbox for <i>user</i>. ■ <code>false</code> (default) to search all tasks to which <i>user</i> has access.
<i>taskSearchQuery</i> or <i>TaskSearchQuery</i>	<p>Document Optional. The <code>TaskSearchQuery</code> document, which specifies the search criteria. See “<code>pub.task:TaskSearchQuery</code>” on page 60 for a description of the fields in this document. If <code>TaskSearchQuery</code> is null, all tasks for <i>user</i> will be returned.</p>
<i>user</i>	<p>String Optional. The user ID of the My webMethods Server user on whose behalf this service will execute. If <i>user</i> is not specified, the user ID associated with the <code>WmTaskClient</code> package is used. For information about configuring this user ID, see <i>webMethods Task Engine User's Guide</i>.</p>

Note: Only tasks to which *user* has access are searched.

Output Parameters

<i>Tasks</i>	<p>Document list Result set containing the tasks that match the search criteria in <code>pub.task:TaskSearchQuery</code>. Each document in the result set has the following structure:</p> <ul style="list-style-type: none"> ■ <i>TaskID</i> String The identifier assigned to the task instance.
--------------	---

- **TaskInfo Document** TaskInfo document containing standard information about the task. See “[pub.task:TaskInfo](#)” on [page 53](#) for a description of the fields in this document.
- **TaskData Document** The TaskData document containing the business data associated with the task. Returned only if `includeTaskData` is set to `true`.

searchTasksFields Mode

This mode is used to return a collection of named fields requested in each service that matches specified search criteria.

Note: This request can only return primitive types (strings, numbers, dates, and so on) or lists (arrays) of primitives. It cannot return complex objects such as all task business data.

Use this HTTP request and include optional parameters in the request body:

```
POST /rest/pub/opentasksearch/fields
```

Input Parameters

fields

String list Optional. List of task fields to be returned by this service. For information about specifying fields, see the fields and operator elements in “[pub.task:TaskSearchQueryTerm](#)” on [page 62](#). For example, suppose you want to return the following fields as result of the query:

- `taskID`
- The business data field `orderID` from task data documents.

You would then pass in following field specifications:

```
#{currentTask.taskInfo.taskID}
#{currentTask.taskData.order.orderID}
```

The service will return values for these fields only.

includeTaskData

String Optional. Specifies whether the service will return the TaskData document as well as the [pub.task:TaskInfo](#) document for each task matching the search criteria. Set to:

- `true` to include the TaskData document in the result set.
- `false` (default) to include only the TaskInfo document in the result set.

searchUserTasks

String Optional. Specifies whether the service searches all tasks or only the user's inbox. Set to:

- `true` to search only the inbox for *user*.
- `false` (default) to search all tasks to which *user* has access.

taskSearchQuery
or
TaskSearchQuery

Document Optional. The *TaskSearchQuery* document, which specifies the search criteria. See “[pub.task:TaskSearchQuery](#)” on page 60 for a description of the fields in this document. If *TaskSearchQuery* is null, all tasks for *user* will be returned

user

String Optional. The user ID of the My webMethods Server user on whose behalf this service will execute. If *user* is not specified, the user ID associated with the *WmTaskClient* package is used. For information about configuring this user ID, see *webMethods Task Engine User’s Guide*.

Note: Only tasks to which *user* has access are searched.

Output Parameters

TaskFields

Document list Result set containing a list of documents whose document type is defined in “[pub.task:TaskFields](#)” on page 52.

searchTasksIndexed Mode

Use this mode to return a subset of tasks that match the specified search criteria on the Task Engine connected to the Integration Server.

The query must contain a search term specifying the *taskTypeID* when searching for business data fields. The Task Engine uses the *taskTypeID* to identify the specific index table to use. You can search for multiple task types in a single query by specifying multiple *taskTypeID* values when you define the search criteria. If you are not searching for business data fields, then a *taskTypeID* is not required.

Use this service only when working with indexed business data fields. If you are working with standard business data fields, use “[searchTasks Mode](#)” on page 84.

Use this HTTP request and include optional parameters in the request body:

```
POST /rest/pub/opentasksearch/indexed
```

Input Parameters

includeTaskData

String Optional. Indicates whether the search query operation includes indexed business data when processing the search query. Use this property when your search query contains search terms that reference indexed business data fields. Set to:

- `true` to include business data in the search query processing.

	<ul style="list-style-type: none"> ■ <code>false</code> (default) when your search query does not require processing of business data or when your task type does not use any defined indexed business fields.
<i>searchUserTasks</i>	<p>String Optional. Specifies whether the service searches all tasks or only the user's inbox. Set to:</p> <ul style="list-style-type: none"> ■ <code>true</code> to search only the inbox for <i>user</i>. ■ <code>false</code> (default) to search all tasks to which <i>user</i> has access.
<i>taskSearchQueryV2</i> or <i>TaskSearchQueryV2</i>	<p>Document Optional. The <code>TaskSearchQueryV2</code> document that specifies the search criteria. See “pub.task.TaskSearchQueryV2” on page 66 for a description of the fields in this document. If <code>TaskSearchQueryV2</code> is null, an error occurs.</p>
<i>user</i>	<p>String Optional. The user ID of the My webMethods Server user on whose behalf this service will execute. If <i>user</i> is not specified, the user ID associated with the <code>WmTaskClient</code> package is used. For information about configuring this user ID, see <i>webMethods Task Engine User's Guide</i>.</p>

Note: Only tasks to which *user* has access are searched.

Output Parameters

<i>Tasks</i>	<p>Document list Result set containing the tasks that match the search criteria in <code>pub.task.TaskSearchQueryV2</code>. Each document in the result set has the following structure:</p> <ul style="list-style-type: none"> ■ <i>TaskID</i> String The identifier assigned to the task instance. ■ <i>TaskInfo</i> Document The <code>TaskInfo</code> document containing standard information about the task. See “pub.task:TaskInfo” on page 53 for a description of the fields in this document. ■ <i>TaskData</i> Document The <code>TaskData</code> document containing the business data associated with the task. Returned only if <i>includeTaskData</i> is set to <code>true</code>.
--------------	--

countTasksIndexed Mode

This mode is used to return the total count of tasks that match the specified search criteria. This request applies only to tasks with indexed business data.

Use this HTTP request and include optional parameters in the request body:

```
POST /rest/pub/opentasksearch/count
```

Input Parameters

<i>includeTaskData</i>	<p>String Optional. Indicates whether the search query operation includes indexed business data when processing the search query. Use this property when your search query contains search terms that reference indexed business data fields. Set to:</p> <ul style="list-style-type: none"> ■ <code>true</code> to include business data in the search query processing. An error occurs if <i>includeTaskData</i> is set to true and the task type does not contain any indexed fields. ■ <code>false</code> (default) when your search query does not require processing of business data or when your task type does not use any defined indexed business fields.
<i>searchUserTasks</i>	<p>String Optional. Specifies whether the service searches all tasks or only the user's inbox. Set to:</p> <ul style="list-style-type: none"> ■ <code>true</code> to search only the inbox for <i>user</i>. ■ <code>false</code> (default) to search all tasks to which <i>user</i> has access.
<i>taskSearchQueryV2</i> or <i>TaskSearchQueryV2</i>	<p>Document Optional. The <i>TaskSearchQueryV2</i> document that specifies the search criteria. See “pub.task.TaskSearchQueryV2” on page 66 for a description of the fields in this document. If <i>TaskSearchQueryV2</i> is null, an error occurs.</p>
<i>user</i>	<p>String Optional. The user ID of the My webMethods Server user on whose behalf this service will execute. If <i>user</i> is not specified, the user ID associated with the <i>WmTaskClient</i> package is used. For information about configuring this user ID, see <i>webMethods Task Engine User's Guide</i>.</p>

Note: Only tasks to which *user* has access are searched.

Output Parameters

<i>totalCount</i>	<p>String Returns the total number of tasks that match the query.</p>
-------------------	--

Usage Notes

The following considerations apply to the available modes:

- **searchTasks.** This mode applies only to tasks that use standard business data.
- **searchTasksFields.** Using this mode instead of the *searchTasks* mode can significantly improve search performance. The *searchTasks* mode returns all *TaskInfo* data plus

TaskData documents, which can result in a very large amount of data. Because this is a Web service (REST) call, the response size becomes an issue, severely limiting performance of this service. With searchTasksFields mode, a very narrow search can be tailored, often requiring only a few TaskInfo fields and some business data fields. Using searchTasksFields mode is strongly recommended to prevent problems.

- **searchTasksIndexed.** If no tasks match the search criteria, an empty TaskData document is returned. The search query must contain a search term specifying the *taskTypeID* to search business fields and create an index of the results.
- **countTasksIndexed.** To be used when specifying the *toIndex* and *fromIndex* parameters of the searchTasksIndexed mode to avoid requesting an index beyond the size of the search results. These two parameters are available in the input document “pub.task.TaskSearchQueryV2” on page 66.

POST a Complex Search Request for HPSTRA-enabled Tasks

This REST request retrieves tasks that are specified with more complex search criteria. The underlying built-in services are:

- pub.task.taskclient:searchTasksHPSTRA
- pub.task.taskclient:searchTasksFieldsHPSTRA
- pub.task.taskclient:searchTasksIndexedHPSTRA
- pub.task.taskclient:countTasksIndexedHPSTRA

This request can be used in the following four modes:

- **searchTasksHPSTRA mode.** Retrieve tasks that match specified search criteria, when working with HPSTRA-enabled tasks that use standard business data.
- **searchTasksFieldsHPSTRA mode.** Return a collection of named fields from HPSTRA-enabled tasks, requested in each service that matches specified search criteria.
- **searchTasksIndexedHPSTRA mode.** Return a subset of tasks that match the specified search criteria, when working with business data fields, indexed in an Elasticsearch store by the HPSTRA module in Task Engine.
- **countTasksIndexedHPSTRA mode.** Return the total count of tasks that match the specified search criteria. This service applies only to tasks with business data fields, indexed in an Elasticsearch store by the HPSTRA module in Task Engine.

For additional information on these modes, see [Usage Notes](#).

searchTasksHPSTRA Mode

Use this mode to retrieve tasks that match specified search criteria. This request applies only to HPSTRA-enabled tasks that use standard business data.

Use this HTTP request and include optional parameters in the request body:

```
POST /rest/pub/opentasksearchhpstra
```

Input Parameters

- includeTaskData* **String** Optional. Specifies whether the service will return the TaskData document as well as the [pub.task:TaskInfo](#) document for each task matching the search criteria. Set to:
- `true` to include the TaskData document in the result set.
 - `false` (default) to include only the TaskInfo document in the result set.
- searchUserTasks* **String** Optional. Specifies whether the service searches all tasks or only the user's inbox. Set to:
- `true` to search only the inbox for *user*.
 - `false` (default) to search all tasks to which *user* has access.
- taskSearchQuery*
or
TaskSearchQuery **Document** Optional. The TaskSearchQuery document, which specifies the search criteria. See "[pub.task:TaskSearchQuery](#)" on page 60 for a description of the fields in this document. If TaskSearchQuery is null, all tasks for *user* will be returned.
- user* **String** Optional. The user ID of the My webMethods Server user on whose behalf this service will execute. If *user* is not specified, the user ID associated with the WmTaskClient package is used. For information about configuring this user ID, see *webMethods Task Engine User's Guide*.

Note: Only tasks to which *user* has access are searched.

Output Parameters

- Tasks* **Document list** Result set containing the tasks that match the search criteria in [pub.task:TaskSearchQuery](#). Each document in the result set has the following structure:
- *TaskID* **String** The identifier assigned to the task instance.
 - *TaskInfo* **Document** TaskInfo document containing standard information about the task. See "[pub.task:TaskInfo](#)" on page 53 for a description of the fields in this document.
 - *TaskData* **Document** The TaskData document containing the business data associated with the task. Returned only if *includeTaskData* is set to `true`.

searchTasksFieldsHPSTRA Mode

This mode is used to return a collection of named fields from HPSTRA-enabled tasks, requested in each service that matches specified search criteria.

Note: This request can only return primitive types (strings, numbers, dates, and so on) or lists (arrays) of primitives. It cannot return complex objects such as all task business data.

Use this HTTP request and include optional parameters in the request body:

```
POST /rest/pub/opentasksearchhpstra/fields
```

Input Parameters

<i>fields</i>	<p>String list Optional. List of task fields to be returned by this service. For information about specifying fields, see the fields and operator elements in “pub.task:TaskSearchQueryTerm” on page 62. For example, suppose you want to return the following fields as result of the query:</p> <ul style="list-style-type: none"> ■ taskID ■ The business data field <code>orderID</code> from task data documents. <p>You would then pass in following field specifications:</p> <pre>#{currentTask.taskInfo.taskID} #{currentTask.taskData.order.orderID}</pre> <p>The service will return values for these fields only.</p>
<i>includeTaskData</i>	<p>String Optional. Specifies whether the service will return the <code>TaskData</code> document as well as the pub.task:TaskInfo document for each task matching the search criteria. Set to:</p> <ul style="list-style-type: none"> ■ <code>true</code> to include the <code>TaskData</code> document in the result set. ■ <code>false</code> (default) to include only the <code>TaskInfo</code> document in the result set.
<i>searchUserTasks</i>	<p>String Optional. Specifies whether the service searches all tasks or only the user's inbox. Set to:</p> <ul style="list-style-type: none"> ■ <code>true</code> to search only the inbox for <i>user</i>. ■ <code>false</code> (default) to search all tasks to which <i>user</i> has access.
<i>taskSearchQuery</i> or <i>TaskSearchQuery</i>	<p>Document Optional. The <code>TaskSearchQuery</code> document, which specifies the search criteria. See “pub.task:TaskSearchQuery”</p>

on [page 60](#) for a description of the fields in this document. If `TaskSearchQuery` is null, all tasks for `user` will be returned

user

String Optional. The user ID of the My webMethods Server user on whose behalf this service will execute. If `user` is not specified, the user ID associated with the `WmTaskClient` package is used. For information about configuring this user ID, see *webMethods Task Engine User's Guide*.

Note: Only tasks to which `user` has access are searched.

Output Parameters

TaskFields

Document list Result set containing a list of documents whose document type is defined in "[pub.task:TaskFields](#)" on [page 52](#).

searchTasksIndexedHPSTRA Mode

Use this mode to return a subset of tasks that match the specified search criteria on the Task Engine connected to the Integration Server.

The query must contain a search term specifying the `taskTypeID` when searching for business data fields. The Task Engine uses the `taskTypeID` to identify the specific index table to use. You can search for multiple task types in a single query by specifying multiple `taskTypeID` values when you define the search criteria. If you are not searching for business data fields, then a `taskTypeID` is not required.

Use this service only when working with tasks with business data fields, indexed in an Elasticsearch store by the HPSTRA module in Task Engine. If you are working with standard business data fields, use "[searchTasksHPSTRA Mode](#)" on [page 89](#).

Use this HTTP request and include optional parameters in the request body:

```
POST /rest/pub/opentasksearchhpstra/indexed
```

Input Parameters

includeTaskData

String Optional. Indicates whether the search query operation includes indexed business data when processing the search query. Use this property when your search query contains search terms that reference indexed business data fields. Set to:

- `true` to include business data in the search query processing.
- `false` (default) when your search query does not require processing of business data or when your task type does not use any defined indexed business fields.

<i>searchUserTasks</i>	<p>String Optional. Specifies whether the service searches all tasks or only the user's inbox. Set to:</p> <ul style="list-style-type: none"> ■ <code>true</code> to search only the inbox for <i>user</i>. ■ <code>false</code> (default) to search all tasks to which <i>user</i> has access.
<i>taskSearchQueryV2</i> or <i>TaskSearchQueryV2</i>	<p>Document Optional. The <code>TaskSearchQueryV2</code> document that specifies the search criteria. See “pub.task.TaskSearchQueryV2” on page 66 for a description of the fields in this document. If <code>TaskSearchQueryV2</code> is null, an error occurs.</p>
<i>user</i>	<p>String Optional. The user ID of the My webMethods Server user on whose behalf this service will execute. If <i>user</i> is not specified, the user ID associated with the <code>WmTaskClient</code> package is used. For information about configuring this user ID, see <i>webMethods Task Engine User's Guide</i>.</p> <p>Note: Only tasks to which <i>user</i> has access are searched.</p>

Output Parameters

<i>Tasks</i>	<p>Document list Result set containing the tasks that match the search criteria in pub.task.TaskSearchQueryV2. Each document in the result set has the following structure:</p> <ul style="list-style-type: none"> ■ <i>TaskID</i> String The identifier assigned to the task instance. ■ <i>TaskInfo</i> Document The <code>TaskInfo</code> document containing standard information about the task. See “pub.task:TaskInfo” on page 53 for a description of the fields in this document. ■ <i>TaskData</i> Document The <code>TaskData</code> document containing the business data associated with the task. Returned only if <i>includeTaskData</i> is set to <code>true</code>.
--------------	---

countTasksIndexedHPSTRA Mode

This mode is used to return the total count of tasks that match the specified search criteria. This request applies only to tasks with business data fields, indexed in an Elasticsearch store by the HPSTRA module in Task Engine.

Use this HTTP request and include optional parameters in the request body:

```
POST /rest/pub/opentasksearchhpstra/count
```

Input Parameters

- includeTaskData* **String** Optional. Indicates whether the search query operation includes indexed business data when processing the search query. Use this property when your search query contains search terms that reference indexed business data fields. Set to:
- `true` to include business data in the search query processing. An error occurs if *includeTaskData* is set to true and the task type does not contain any indexed fields.
 - `false` (default) when your search query does not require processing of business data or when your task type does not use any defined indexed business fields.
- searchUserTasks* **String** Optional. Specifies whether the service searches all tasks or only the user's inbox. Set to:
- `true` to search only the inbox for *user*.
 - `false` (default) to search all tasks to which *user* has access.
- taskSearchQueryV2*
or
TaskSearchQueryV2 **Document** Optional. The `TaskSearchQueryV2` document that specifies the search criteria. See [“pub.task.TaskSearchQueryV2” on page 66](#) for a description of the fields in this document. If `TaskSearchQueryV2` is null, an error occurs.
- user* **String** Optional. The user ID of the My webMethods Server user on whose behalf this service will execute. If *user* is not specified, the user ID associated with the `WmTaskClient` package is used. For information about configuring this user ID, see *webMethods Task Engine User's Guide*.

Note: Only tasks to which *user* has access are searched.

Output Parameters

- totalCount* **String** Returns the total number of tasks that match the query.

Usage Notes

The following considerations apply to the available modes:

- **searchTasksHPSTRA.** This mode applies only to HPSTRA-enabled tasks that use standard business data.

- **searchTasksFieldsHPSTRA.** Using this mode instead of the searchTasks mode can significantly improve search performance. The searchTasksHPSTRA mode returns all TaskInfo data plus TaskData documents, which can result in a very large amount of data. Because this is a Web service (REST) call, the response size becomes an issue, severely limiting performance of this service. With searchTasksFieldsHPSTRA mode, a very narrow search can be tailored, often requiring only a few TaskInfo fields and some business data fields. Using searchTasksFieldsHPSTRA mode is strongly recommended to prevent problems.
- **searchTasksIndexedHPSTRA.** If no tasks match the search criteria, an empty TaskData document is returned. The search query must contain a search term specifying the *taskTypeID* to search business fields and create an index of the results.
- **countTasksIndexedHPSTRA.** To be used when specifying the *toIndex* and *fromIndex* parameters of the searchTasksIndexedHPSTRA mode to avoid requesting an index beyond the size of the search results. These two parameters are available in the input document “[pub.task.TaskSearchQueryV2](#)” on page 66.

GET Task Audit Information

This REST request returns the audit log of all of the operations performed on a task, or a specific audit entry. The underlying built-in service is `pub.task.taskclient:getTaskAudit`.

The HTTP request to return all audit information is formed as follows:

```
GET /rest/pub/opentask/taskID/audit
```

The HTTP request to return a specific audit entry is formed as follows:

```
GET /rest/pub/opentask/taskID/audit/auditEntryID
```

Input Parameters

<i>taskID</i>	String The unique identifier that the Task Engine assigns to the task
<i>auditEntryID</i>	String Optional. The unique identifier of the task audit entry to be returned.

Output Parameters

<i>taskAudits</i>	TaskAudit[] An array of TaskAudit objects representing the audit logs of the task. See “ pub.task:TaskAudit ” on page 49 for information about the structure of this object.
-------------------	---

DELETE Audit Entries to Rollback Task

This REST request enables you to roll back the task to any available audit point in the task's audit history. The DELETE request removes the unwanted audit entries to enable rollback to the desired audit entry point. The underlying built-in service is `pub.task.taskclient.rollbackTask`.

The HTTP request is formed as follows:

```
DELETE /rest/pub/opentask/taskID/audit/auditEntryID
```

Input Parameters

<i>auditEntryID</i>	Integer The ID of the audit log entry that you want to roll the task back to.
<i>taskID</i>	String The unique identifier that the Task Engine assigns to the task instance.

Output Parameters

None.

GET or POST a Form Flow Notification

This REST request notifies a waiting step in a task workflow that the workflow is completed. The underlying built-in service is `pub.task.taskclient.formFlowTaskNotify`.

For example, when the final task step in a task workflow completes, that step waits for notification. However, in this case there are no more steps in the workflow to notify the waiting step. You can configure a following process service activity step to call this service and pass a result to the waiting step that indicates the workflow is complete. This enables the process to execute to completion.

For more information about task workflows, see “Working with Task Workflows” in the *webMethods BPM Task Development Help*.

Use this HTTP request if no optional parameters are required:

```
GET /rest/pub/opentask/formFlowNotify/correlationID
```

Use this HTTP request if you want to include optional parameters in the request body:

```
POST /rest/pub/opentask/formFlowNotify/correlationID
```

Input Parameters

correlationID **String** Use a task workflow correlation ID to synchronize the communication between the waiting step and the notifying step. This ensures correct data flow through the process. You can define any value but the task workflow correlation ID must be unique within the Process Engine environment.

ImportantThe task workflow correlation ID is completely different from and unrelated to the standard document correlation ID often used in process implementation.

localOnly **String** Optional. Set to:

- `true` if the service is not being used in a cluster environment.
- `false` (default) if the service is being used in a clustered environment.

result **String** Optional. This can be any value to be passed back to the waiting component. Typically, you use this field to pass a status code to the waiting step to indicate the end of the workflow, which enables the waiting step to take an action upon receiving the result.

Output Parameters

None.

Usage Notes

Technically, you can also use this service within a workflow to notify a waiting task activity that the next task activity in the workflow is instantiated. However, the Java API `com.webmethods.portal.service.task.ITaskFormFlowService` is recommended for this task. To read the Java docs for this component, see the *webMethods CAF and My webMethods Server Java API Reference*, available from the “[Software AG Documentation web site](#)” under My webMethods Server in the corresponding webMethods Product Suite release number.

Sample Code Available

For more information about implementing a task workflow, you can examine and deploy a sample task application, process model, and Integration Server package that support a very simple loan application process. You can find the task workflow code samples on the Software AG Community web site

A description of the sample code and its behavior can be found in the topic “Working with Task Workflows” in the *webMethods BPM Task Development Help*.

DELETE Task Comment

This REST request deletes the specified task comment from the specified task instance. The underlying built-in service is `pub.task.taskclient:deleteTaskComment`.

The HTTP request is formed as follows:

```
DELETE /rest/pub/opentask/taskID/comments/commentID[?user=value]
```

Input Parameters

<i>commentID</i>	String The unique identifier of the comment to be deleted.
<i>taskID</i>	String The unique identifier that the Task Engine assigns to the task instance.
<i>user</i>	String Optional. The user ID of the My webMethods Server user on whose behalf this operation executes. If <i>user</i> is not specified, the user ID associated with the WmTaskClient package is used. For information about configuring this user ID, see <i>webMethods Task Engine User's Guide</i> .

Output Parameters

None.

GET Task Comments

This REST request returns either a specific comment or all the comments in a task instance, including associated attachments. The underlying built-in service is `pub.task.taskclient:getTaskComments`.

To return a specific comment, the HTTP request is formed as follows:

```
GET /rest/pub/opentask/taskID/comments/commentID
```

To return all comments, the HTTP request is formed as follows:

```
GET /rest/pub/opentask/taskID/comments
```

Input Parameters

<i>taskID</i>	String The unique identifier that the Task Engine assigns to the task instance.
<i>commentID</i>	String Optional. The unique identifier of the comment to be returned.
<i>includeAttachmentData</i>	Boolean Optional. Specifies whether or not to include Base64 encoded data in the attachments. Set to: <ul style="list-style-type: none"> ■ <code>true</code> to include Base64 data. ■ <code>false</code> to omit Base64 data (default).

Output Parameters

<i>taskComments</i>	TaskComment[] An array of TaskComment objects that represent the comments retrieved from the task instance. See “pub.task:TaskComment” on page 51 for more information about the structure of this object.
---------------------	---

PUT Updates into a Task Comment

This REST request updates a comment in a task instance, including any attachments associated with the comment. The underlying built-in service is `pub.task.taskclient:updateTaskComment`.

The HTTP request is formed as follows:

```
PUT /rest/pub/opentask/taskID/comments/commentID[?param1=value]
    [&param2=value] . . . [&param4=value]
```

Input Parameters

<i>attachments</i>	TaskAttachment[] Optional. An array of TaskAttachment objects representing any attachments to be added to the comment being updated. See “pub.task:TaskAttachment” on page 48 for the structure of this data.
<i>commentID</i>	String The unique identifier of a comment to be deleted.

<i>deleteAttachments</i>	String[] Optional. An array of IDs of any attachments to be deleted from the comment.
<i>taskID</i>	String The unique identifier that the Task Engine assigns to the task instance.
<i>text</i>	String Optional. The updated text for this comment.
<i>user</i>	String Optional. The user ID of the My webMethods Server user on whose behalf this operation executes. If <i>user</i> is not specified, the user ID associated with the WmTaskClient package is used. For information about configuring this user ID, see <i>webMethods Task Engine User's Guide</i> .

Output Parameters

None.

POST Task Comment

This REST request adds a comment to a task instance. Attachments can be included. The underlying built-in service is `pub.task.taskclient:addTaskAttachment`.

Use this HTTP request and include optional parameters in the request body:

```
POST /rest/pub/opentask/taskID/comments/
```

Input Parameters

<i>attachments</i>	TaskAttachment [] Optional. An array of TaskAttachment objects representing any attachments to be added to the comment being added.
<i>html</i>	Boolean Optional. Set to: <ul style="list-style-type: none"> ■ <code>true</code> if the comment text contains HTML markup. ■ <code>false</code> (default) if the comment text is to be treated as plain text.
<i>taskID</i>	String The unique identifier that the Task Engine assigns to the task instance.
<i>text</i>	String Optional. The updated text for this comment.

user **String** Optional. The user ID of the My webMethods Server user on whose behalf this operation executes. If *user* is not specified, the user ID associated with the WmTaskClient package is used. For information about configuring this user ID, see *webMethods Task Engine User's Guide*.

Output Parameters

commentID **String** The unique identifier that the Task Engine assigns to the newly added comment.

DELETE Tags From Task

This REST request deletes all the tags from a task. The underlying built-in service is `pub.task.taskclient:deleteTagsFromTask`.

The HTTP request is formed as follows:

```
DELETE /rest/pub/opentask/taskID/tags?taskID=value
```

Input Parameters

taskID **String** The unique identifier that the Task Engine assigns to the task.

Output Parameters

None.

GET Tags By Task

This REST request returns a list of tags associated with a *taskID*. The underlying built-in service is `pub.task.taskclient:listTagsByTask`.

The HTTP request is formed as follows:

```
GET /rest/pub/opentask/taskID/tags?taskID=value
```

Input Parameters

taskID **String** The unique identifier that the Task Engine assigns to the task.

Output Parameters

Tags **String Array** List of tags associated with a *taskID* .

GET Task Expert List

This REST request returns a list of experts associated with the specified tags. The underlying built-in service is `pub.task.taskclient:getTaskExpertList`.

The HTTP request is formed as follows:

```
GET /rest/pub/opentask/taskID/recommendation?tags=value
```

Input Parameters

tags **String** Comma-separated list of tags.

Output Parameters

guidance **Guidance[]** An array of Guidance objects containing the names, ratings, and user IDs of the experts for the specified tags. See [pub.task:Guidance](#) for a description of the fields in this document.

POST Tags To Task

This REST request adds the specified tags to a task. The underlying built-in service is `pub.task.taskclient:addTagsToTask`.

Use this HTTP request and include optional parameters in the request body:

```
POST /rest/pub/opentask/taskID/tags
```

Input Parameters

taskID **String** The unique identifier that the Task Engine assigns to the task instance.

tags **String** Comma separated list of text to be used to define tags. No character limit enforced. Alphanumeric characters, spaces, underscores, and dashes are allowed.

Output Parameters

None.

PUT Task Tags

This REST request updates the tags associated with a task. The underlying built-in service is `pub.task.taskclient:updateTaskTags`.

The HTTP request is formed as follows:

```
PUT /rest/pub/opentask/taskID/tags?taskID=value&tags=value
```

Input Parameters

<i>taskID</i>	String The unique identifier that the Task Engine assigns to the task.
<i>tags</i>	String Comma separated list of text to be used to define tags. No character limit enforced. Alphanumeric characters, spaces, underscores, and dashes are allowed.

Output Parameters

None.

DELETE Task Attachment

This REST request deletes the specified task attachment from a task instance. The underlying built-in service is `pub.task.taskclient:deleteTaskAttachment`.

The HTTP request is formed as follows:

```
DELETE /rest/pub/opentask/taskID/attachments/attachmentID [ ?user ]
```

Input Parameters

<i>attachmentID</i>	String The unique identifier of the attachment to be deleted.
<i>taskID</i>	String The unique identifier that the Task Engine assigns to the task instance.

user **String** Optional. The user ID of the My webMethods Server user on whose behalf this operation executes. If *user* is not specified, the user ID associated with the WmTaskClient package is used. For information about configuring this user ID, see *webMethods Task Engine User's Guide*.

Output Parameters

None.

GET Task Attachments

This REST request returns all the attachments in a task instance. The underlying built-in service is `pub.task.taskclient:getTaskAttachments`.

To return a specific attachment, the HTTP request is formed as follows:

```
GET /rest/pub/opentask/taskID/attachments/attachmentID
```

To return all attachments, the HTTP request is formed as follows:

```
GET /rest/pub/opentask/taskID/attachments
```

Input Parameters

attachmentID **String** The unique identifier of the attachment to be deleted.

includeAttachmentData **Boolean** Optional. Specifies whether or not to include Base64 encoded data in the attachments. Set to:

- `true` to include Base64 data.
- `false` to omit Base64 data (default).

taskID **String** The unique identifier that the Task Engine assigns to the task instance.

Output Parameters

taskAttachments **TaskAttachment[]** An array of TaskAttachment objects that represent the attachments in the task instance. See [“pub.task:TaskAttachment” on page 48](#) for the structure of this data.

PUT Task Attachment

This REST request updates the specified task attachment in a task instance. The underlying built-in service is `pub.task.taskclient:updateTaskAttachment`.

The HTTP request is formed as follows:

```
PUT /rest/pub/opentask/taskID/attachments/attachmentID [ ?attachment=value ]
    [ &encoding=value ] [ &user=value ]
```

Input Parameters

<i>attachment</i>	TaskAttachment The <code>TaskAttachment</code> object representing the new attachment to be used to update the existing attachment. See “pub.task:TaskAttachment” on page 48 for the structure of this data.
<i>attachmentID</i>	String The unique identifier of the attachment to be updated.
<i>encoding</i>	String Optional. Specify the encoding type for the file. Set to: <ul style="list-style-type: none"> ■ binary ■ text
<i>taskID</i>	String The unique identifier that the Task Engine assigns to the task instance.
<i>user</i>	String Optional. The user ID of the My webMethods Server user on whose behalf this operation executes. If <i>user</i> is not specified, the user ID associated with the <code>WmTaskClient</code> package is used. For information about configuring this user ID, see <i>webMethods Task Engine User’s Guide</i> .

Output Parameters

None.

POST Task Attachment

This REST request adds the specified attachment to a task instance. The underlying built-in service is `pub.task.taskclient:addTaskComment`.

Use this HTTP request and include optional parameters in the request body:

POST /rest/pub/opentask/*taskID*/attachments

Input Parameters

<i>attachment</i>	TaskAttachment The TaskAttachment object representing the new attachment to be used to update the existing attachment. See “pub.task:TaskAttachment” on page 48 for the structure of this data.
<i>encoding</i>	String Optional. Specify the encoding type for the file. Set to: <ul style="list-style-type: none">■ binary■ text
<i>taskID</i>	String The unique identifier that the Task Engine assigns to the task instance.
<i>user</i>	String Optional. The user ID of the My webMethods Server user on whose behalf this operation executes. If <i>user</i> is not specified, the user ID associated with the WmTaskClient package is used. For information about configuring this user ID, see <i>webMethods Task Engine User’s Guide</i> .

Output Parameters

None.

3 Using the Task Engine RESTful Web Services in My webMethods Server

■ Overview	108
■ Summary of REST Resources and Requests Through My webMethods Server	108
■ DELETE a Task Instance	110
■ GET Task Instance Information	111
■ POST a New Task Instance	112
■ PUT Information in a Task Instance	113
■ GET Tasks with a Simple Search Request	115
■ POST a Complex Task Search Request	116
■ GET Task Audit Information	121
■ DELETE Audit Entries to Rollback Task	121
■ GET or POST a Form Flow Notification	122
■ DELETE Task Comment	123
■ GET Task Comments	124
■ PUT Updates into a Task Comment	124
■ POST Task Comment	125
■ DELETE Task Attachment	126
■ GET Task Attachments	126
■ PUT Task Attachment	127
■ POST Task Attachment	128

Overview

webMethods Task Engine enables web applications to use RESTful services in My webMethods Server and directly communicate with the Task Engine.

The Task Engine RESTful services for My webMethods Server share the data structures of the web services API, as described in the [“Summary of Data Structures” on page 152](#).

You can specify the RESTful service parameters as key-value pairs in the RESTful web service URL, or as part of an associated JSON document.

Note that the parameters of the Task Engine RESTful services for My webMethods Server are case-sensitive. Also, you cannot specify the service parameters in an XML file.

Summary of REST Resources and Requests Through My webMethods Server

My webMethods Server provides REST requests that can be applied to the following Task Engine resources:

Task Instances

Request	Description
DELETE a Task Instance	Deletes an instance of a task on the Task Engine.
GET Task Instance Information	Returns the TaskInfo object, and optionally the TaskData object, for a specified task.
POST a New Task Instance	Creates a new task instance and adds it to the Task Engine queue.
PUT Information in a Task Instance	Modifies a task instance on the Task Engines.
GET Tasks with a Simple Search Request	Retrieves tasks that match simple search criteria.
POST a Complex Task Search Request	Retrieves tasks that are specified with more complex search criteria.

Task Audit Information

Request	Description
GET Task Audit Information	Returns the audit log of all of the operations performed on a task, or a specific audit entry.
DELETE Audit Entries to Rollback Task	Enables you to roll back the task to any available audit point in the task's audit history.
GET or POST a Form Flow Notification	Notifies a waiting step in a task workflow that the workflow is completed.

Task Comments

Request	Description
DELETE Task Comment	Deletes the specified task comment from a task instance as well as all associated attachments.
GET Task Comments	Returns either a specific comment or all the comments in a task instance including associated attachments.
PUT Updates into a Task Comment	Updates a comment in a task instance, including attachments associated with the comment.
POST Task Comment	Adds a comment to a task instance. Attachments can be included.

Task Attachments

Request	Description
DELETE Task Attachment	Deletes the specified task attachment from a task instance.

Request	Description
GET Task Attachments	Returns all the attachments in a task instance.
PUT Task Attachment	Updates the specified task attachment in a task instance.
POST Task Attachment	Adds the specified attachment to a task instance.

Examples

- To retrieve the details of a task instance with ID 7565, make the following REST call to the Task Engine through the My webMethods Server running on the local machine:

```
GET /rest/pub/opentask?taskID=7565
```

- To post a JSON document as task attachment:

```
POST /rest/pub/opentask/taskID/attachments
```

Sample JSON document Content:

```
{ "attachment": [ { "name": "test_comment_attach", "base64Data": "VGhpcyBpcyBhIGxvbmcgY29tbWVudCBmb3JtIG1lIGZyb20gdGhlIHHRleHQgZmlsZQ==", "contentType": "text/plain" } ] }
```

DELETE a Task Instance

This REST request deletes an instance of a task on the Task Engine.

The HTTP request is formed as follows:

```
DELETE /rest/pub/opentask/taskID [ ?user=value ]
```

or

```
DELETE /rest/pub/opentask?taskID=taskID [ &user=value ]
```

Input Parameters

<i>taskID</i>	String The ID that identifies the task that you want to delete.
<i>user</i>	String Optional. The user ID of the My webMethods Server user on whose behalf this operation executes. If <i>user</i> is not specified, the administrative user ID under which your client program logged on is used.

Output Parameters

None.

Usage Notes

To delete a task using this operation, you must supply the ID of the task that you want to delete. To obtain this ID, use the [searchTasks](#) operation to locate the task and then extract the task ID from the result set that the [searchTasks](#) operation returns.

To delete a task successfully, the user ID in *user* must have permission to access the specified task. If the supplied ID does not have access to the task, or if the specified task does not exist, an exception is thrown.

GET Task Instance Information

This REST request returns the [TaskInfo](#) object, and optionally the [TaskData](#) object, for a specified task.

The [TaskInfo](#) object contains standard information about a task, including its status, expiration date, and the list of users to which it is assigned. The [TaskData](#) object contains the business data that is associated with the task.

To use the request to retrieve information about a list of tasks, the HTTP request is formed as follows:

```
GET /rest/pub/opentask[?param1=value] [&param2=value] [&param2=value]
```

Input Parameters

<i>includeTaskData</i>	Boolean Optional. Specifies whether you want to retrieve the TaskData object as well as the TaskInfo object. Set to: <ul style="list-style-type: none">■ <code>true</code> to retrieve TaskData in addition to TaskInfo■ <code>false</code> (default) to retrieve only TaskInfo
<i>taskID</i>	String The ID that identifies the task that you want to retrieve.
<i>user</i>	String Optional. The user ID of the My webMethods Server user on whose behalf this operation executes. If <i>user</i> is not specified, the administrative user ID under which your client program logged on is used.

Output Parameters

Task **Task** A Task object containing the task's TaskInfo object and, optionally, its TaskData object. See [Task](#) for a description of the fields in this object.

The TaskData object is included in *Task* only if *includeTaskData* is set to true.

Usage Notes

To get a task using this operation, you must supply the ID of the task that you want to retrieve. To obtain this ID, use the [searchTasks](#) operation to locate the task and then extract the task ID from the result set that the [searchTasks](#) operation returns.

To retrieve a task successfully, the user ID specified in *user* must have permission to access that task. If the supplied ID does not have access to the task, or if the specified task does not exist, an exception is thrown.

You can edit the *taskScheduleDate* field only when the task is in `scheduled` state.

You cannot edit the schedules of task instances created using Task Engine 9.9 or earlier because those task instances will be in `active` state.

POST a New Task Instance

This REST request creates a new task instance and adds it to the Task Engine queue.

Use this HTTP request and include optional parameters in the request body:

```
POST /rest/pub/opentask
```

Input Parameters

taskData **Map** Optional. The business data, if any, associated with this task. See [TaskData](#) for a description of the Map object used by *taskData*.

taskInfo **TaskInfo** Optional. A complex structure that contains standard information about the task (for example, name, expiration date, list of users to which the task is assigned). See [TaskInfo](#) for a description of the fields in this object.

taskTypeID **String** Specifies the type of task that you want to queue. Each task type that is deployed on the Task Engine has a unique ID. This ID is assigned by the developer when he or she creates a task application using Software AG Designer.

If you do not know the ID for the task type that you want to queue, contact the Task Engine administrator. An administrator can obtain a list of the task types by viewing the **Task Configuration** panel on the Task Engine Administration page in My webMethods.

Task type IDs are case-sensitive. The ID in *taskTypeID* must exactly match the ID as it is specified on the Task Engine.

user **String** Optional. The user ID of the My webMethods Server user on whose behalf this operation executes.

If *user* is not specified, the user ID under which your client program logged on is used.

Output Parameters

taskID **String** The identifier that the Task Engine assigns to the task instance.

Usage Notes

The user ID specified in *user* must have permission to queue tasks on the Task Engine. If the supplied ID does not have this permission, an exception is generated.

If the date specified in *taskScheduleDate* is later than the current date, the status of the task will be set to `scheduled`. When the task starts or queues at the scheduled time, the global rule for task schedule changes the status of the task from `scheduled` to `active`.

PUT Information in a Task Instance

This REST request modifies a task instance on the Task Engine.

You use this request to change information in the task's `TaskInfo` and `TaskData` documents. This service applies only to tasks with status `Active`, `Error`, and `Suspended`. For information about updating tasks with other statuses, see *Usage Notes*.

The `TaskInfo` object contains standard information about a task, including its status, expiration date, and the list of users to which it is assigned. The `TaskData` object contains the business data that is associated with the task.

This is the operation a client program would use if it needed to change the status of a task. For example, if a program wanted to end a task, it would use this operation to change the *status* field in the task's `TaskInfo` object to "completed" or "cancelled" as appropriate.

To use the request to update a task, the HTTP request is formed as follows:

```
PUT /rest/pub/opentask/taskID [ ?param1=value ] [ &param2=value ] . . . [ &param4=value ]
```

or

PUT /rest/pub/opentask?taskID=*taskID* [¶m1=*value*] . . . [¶m4=*value*]

Input Parameters

<i>taskData</i>	Map The Map object that contains the business data that you want to assign to the task. See TaskData for a description of the Map object used by <i>taskData</i> .
<i>taskID</i>	String The ID that identifies the task that you want to modify.
<i>taskInfo</i>	TaskInfo Object that specifies the changes you want to make to the data in the task's TaskInfo object. See TaskInfo for a description of the fields in this object.
<i>user</i>	String Optional. The user ID of the My webMethods Server user on whose behalf this operation executes. If <i>user</i> is not specified, the administrative user ID under which your client program logged on is used.

Output Parameters

None.

Usage Notes

To modify a task using this operation, you must supply the ID of the task that you want to update. To obtain this ID, use the [searchTasks](#) operation to locate the task and then extract the task ID from the result set that the [searchTasks](#) operation returns.

To modify a task successfully, the user ID in *user* must have permission to access the specified task. If the supplied ID does not have access to the task, or if the specified task does not exist, an exception is thrown.

When you use this operation to update information in the task's TaskInfo object, the Task Engine modifies *only* the fields to which you have assigned values in *taskInfo*. All other fields in the target task retain their existing values. For example, to change just the name assigned to a task, you would set only the *name* field in the *TaskInfo* object that you pass to this service.

When you use this operation to update information in the task's TaskData object, the object that you pass to the service in *taskData* replaces the task's existing TaskData object in its entirety.

When updating task statuses with this service, the following limitations apply:

- If the current status of the task is Error, you can change it only to Completed.
- If the current status of the task is Suspended, you cannot set it to Completed or Expired.

This service applies only to tasks with status Active, Error, and Suspended. To enable the service to update tasks with other statuses, set the additional JVM property - `Dupdate.completed.task` to `true`, as described in *webMethods Task Engine User's Guide*.

Tasks with status Error that belong to a process instance cannot be updated, regardless of Task Engine optional settings.

You can edit the `taskScheduleDate` field only when the task is in `scheduled` state.

You cannot edit the schedules of task instances created using Task Engine 9.9 or earlier because those task instances will be in `active` state.

GET Tasks with a Simple Search Request

This REST request retrieves tasks that match simple search criteria.

The HTTP request will be formed as follows:

```
GET /rest/pub/opentasksearch[?param1=value] [&param2=value] . . . [&param6=value]
```

Input Parameters

<i>includeTaskData</i>	<p>Boolean Optional. Specifies whether the operation returns the TaskData object as well as the TaskInfo object for each task matching the search criteria. Set to:</p> <ul style="list-style-type: none"> ■ <code>true</code> to include the TaskData in the result set. ■ <code>false</code> (default) to include only the TaskInfo in the result set.
<i>taskSearchQuery</i>	<p>TaskSearchQuery The TaskSearchQuery object, which specifies the search criteria. See TaskSearchQuery for a description of the fields in this object.</p> <p>If <i>TaskSearchQuery</i> is null, all tasks for <i>user</i> are returned.</p>
<i>user</i>	<p>String Optional. The user ID of the My webMethods Server user on whose behalf this operation executes.</p> <p>If <i>user</i> is not specified, the administrative user ID under which your client program logged on is used.</p> <div style="background-color: #f0f0f0; padding: 5px; margin-top: 10px;"> <p>Note: Only tasks to which <i>user</i> has access are searched.</p> </div>
<i>userTasks</i>	<p>Boolean Optional. Specifies whether the operation searches all tasks or just the user's inbox. This parameter is overridden if the <i>user</i> parameter contains a value.</p> <p>Set to:</p>

- `true` to execute the search only on the user's inbox.
- `false` to search accessible user tasks; this is the default value.

By default, only active tasks are returned. This can be overridden with the `showNonActiveTasks` parameter in [TaskSearchQuery](#).

Output Parameters

tasks **Task[]** An array of Task objects representing the tasks that matched the search criteria in *TaskSearchQuery*. See [Task](#) for a description of the fields in a Task object.

Usage Notes

If no tasks match the search criteria, an empty array is returned.

POST a Complex Task Search Request

This REST request retrieves tasks that match the specified complex search criteria.

This request can be used in the following modes:

- **searchTasksFields mode.** Return a collection of named fields requested in each service that matches specified search criteria.
- **searchTasksIndexed mode.** Return a subset of tasks that match the specified search criteria, when working with indexed business data fields.
- **countTasksIndexed mode.** Return the total count of tasks that match the specified search criteria. This service applies only to tasks with indexed business data.

For additional information on these modes, see "[Usage Notes](#)".

searchTasksFields Mode

This mode is used to return a list of tasks that match specified field search criteria.

Note: This request can return only primitive types (for example, Strings, Numbers, Dates) or lists (arrays) of primitives. It cannot return complex objects such as all task business data.

Use this HTTP request and include optional parameters in the request body:

```
POST /rest/pub/opentasksearch/fields
```

Input Parameters

<i>fields</i>	<p>String List of tasks fields that are to be returned by this service. For information about specifying fields, see the <i>fields</i> and <i>operator</i> parameters in TaskSearchQueryTerm. For example, suppose you want to return the following fields as result of the query:</p> <ul style="list-style-type: none"> ■ taskID ■ The business data field <code>orderID</code> from task data documents. <p>You would then pass in following field specifications:</p> <pre>#{currentTask.taskInfo.taskID} #{currentTask.taskData.order.orderID}</pre> <p>This API returns the values for these fields only.</p>
<i>includeTaskData</i>	<p>String Optional. Specifies whether the operation returns the <code>TaskData</code> object as well as the <code>TaskInfo</code> object for each task matching the search criteria. Set to:</p> <ul style="list-style-type: none"> ■ <code>true</code> to include the <code>TaskData</code> in the result set. ■ <code>false</code> (default) to include only the <code>TaskInfo</code> in the result set.
<i>taskSearchQuery</i>	<p>TaskSearchQuery The <code>TaskSearchQuery</code> object, which specifies the search criteria. See TaskSearchQuery for a description of the fields in this object.</p> <p>If <i>TaskSearchQuery</i> is null, all tasks for <i>user</i> are returned.</p>
<i>user</i>	<p>String Optional. The user ID of the My webMethods Server user on whose behalf this operation executes.</p> <p>If <i>user</i> is not specified, the administrative user ID under which your client program logged on is used.</p> <div style="background-color: #f0f0f0; padding: 5px; margin-top: 10px;"> <p>Note: Only tasks to which <i>user</i> has access are searched.</p> </div>
<i>fields</i>	<p>String List of tasks fields that are to be returned by this service. For information about specifying fields, see the <i>fields</i> and <i>operator</i> elements in TaskSearchQuery. For example, suppose you want to return the following fields as result of the query:</p> <ul style="list-style-type: none"> ■ taskID ■ The business data field <code>orderID</code> from task data documents. <p>You would then pass in following field specifications:</p>

```
#{currentTask.taskInfo.taskID}
#{currentTask.taskData.order.orderID}
```

This API returns the values for these fields only.

Output Parameters

Task Field

Document List Result set containing the tasks that match the search criteria in *TaskSearchQuery*. Each document in the result set has the following structure:

- *name* **String** Name of the task.
- *value* **Object** The value of the field contents for the specified field(s).

That is, the service returns a collection of tasks that match the query. For each task, it returns a collection of specified fields.

The purpose of this API is to ease performance issues of searchTasks. It is quite common that only few taskInfo fields along with some business data fields are returned. searchTasks returns the entire TaskInfo plus TaskData documents. The TaskData could be very large. This could cause the SOAP response size to become an issue, severely limiting the performance of this service. By using searchTaskFields instead, the user can choose to return only a few fields which can help improve performance.

searchTasksIndexed Mode

Use this mode to return a subset of tasks that match the specified search criteria against indexed business data.

The query must contain a search term specifying the *taskTypeID* when searching for business data fields. The Task Engine uses the *taskTypeID* to identify the specific index table to use. You can search for multiple task types in a single query by specifying multiple *taskTypeID* values when you define the search criteria. If you are not searching for business data fields, then a *taskTypeID* is not required.

Use this HTTP request and include optional parameters in the request body:

```
POST /rest/pub/opentasksearch/indexed
```

Input Parameters

businessData

Boolean Optional. Indicates whether the search query operation includes indexed business data when processing the search query. Use this property when your search query contains search terms that reference indexed business data fields. Set to:

- `true` to include business data in the search query processing.

	<ul style="list-style-type: none"> ■ <code>false</code> (default) when your search query does not require processing of business data or when your task type does not use any defined indexed business fields.
<i>taskSearchQuery</i>	TaskSearchQueryV2 Specifies the TaskSearchQueryV2 object that provides the search criteria. If <i>taskSearchQueryV2</i> is null, an error occurs.
<i>user</i>	String Optional. Specifies the user ID of the My webMethods Server user for which the operation executes. If <i>user</i> is not specified, the administrative user ID under which your client program logged on as is used. When you use this parameter, the search is limited to tasks that are accessible by the user.
<i>userTasks</i>	<p>Boolean Optional. Indicates whether the operation searches all tasks or only the user's inbox. This parameter is overridden if the <i>user</i> parameter contains a value. Set to:</p> <ul style="list-style-type: none"> ■ <code>true</code> to execute the search only on the user's inbox. ■ <code>false</code> (default) to search accessible user tasks. <p>By default, only active tasks are returned. This can be overridden with the <i>showNonActiveTasks</i> parameter in TaskSearchQuery.</p>

Output Parameters

<i>tasks</i>	<p>Task Specifies an array of Task objects representing the tasks that match the search criteria in TaskSearchQueryV2.</p> <p>If no tasks match the search criteria, an empty array is returned.</p>
--------------	---

countTasksIndexed Mode

This mode is used to return the total count of tasks that match the specified search criteria. This request applies only to tasks with indexed business data.

Use this HTTP request and include optional parameters in the request body:

```
POST /rest/pub/opentasksearch/count
```

Input Parameters

<i>includeTaskData</i>	<p>String Optional. Indicates whether the search query operation includes indexed business data when processing the search query. Use this property when your search query contains search terms that reference indexed business data fields. Set to:</p>
------------------------	--

	<ul style="list-style-type: none"> ■ <code>true</code> to include business data in the search query processing. An error occurs if <code>businessData</code> is set to <code>true</code> and the task type does not contain any indexed fields. ■ <code>false</code> (default) when your search query does not require processing of business data or when your task type does not use any defined indexed business fields.
<i>searchUserTasks</i>	<p>String Optional. Specifies whether the service searches all tasks or only the user's inbox. Set to:</p> <ul style="list-style-type: none"> ■ <code>true</code> to search only the inbox for <i>user</i>. ■ <code>false</code> (default) to search all tasks to which <i>user</i> has access.
<i>taskSearchQueryV2</i> or <i>TaskSearchQueryV2</i>	<p>Document Optional. The <code>TaskSearchQueryV2</code> document that specifies the search criteria. See TaskSearchQueryV2 for a description of the fields in this document. If <code>TaskSearchQueryV2</code> is null, an error occurs.</p>
<i>user</i>	<p>String Optional. The user ID of the My webMethods Server user on whose behalf this service will execute.</p> <p>Note: Only tasks to which <i>user</i> has access are searched.</p>

Output Parameters

totalCount **String** Returns the total number of tasks that match the query.

Usage Notes

The following considerations apply to the available modes:

- **searchTasksFields.** Using this mode instead of the `searchTasks` mode can significantly improve search performance. The `searchTasks` mode returns all `TaskInfo` data plus `TaskData` documents, which can result in a very large amount of data. Because this is a Web service (REST) call, the response size becomes an issue, severely limiting performance of this service. With `searchTasksFields` mode, a very narrow search can be tailored, often requiring only a few `TaskInfo` fields and some business data fields. Using `searchTasksFields` mode is strongly recommended to prevent problems.
- **searchTasksIndexed.** If no tasks match the search criteria, an empty `TaskData` document is returned. The search query must contain a search term specifying the *taskTypeID* to search business fields and create an index of the results.
- **countTasksIndexed.** To be used when specifying the *toIndex* and *fromIndex* parameters of the `searchTasksIndexed` mode to avoid requesting an index beyond the size

of the search results. These two parameters are available in the input document [TaskSearchQueryV2](#).

GET Task Audit Information

This REST request returns the audit log of all of the operations performed on a task.

The HTTP request to return all audit information is formed as follows:

```
GET /rest/pub/opentask/taskID/audit
```

The HTTP request to return a specific audit entry is formed as follows:

```
GET /rest/pub/opentask/taskID/audit/auditEntryID
```

Input Parameters

taskID **String** The unique identifier that the Task Engine assigns to the task

Output Parameters

taskAudits **TaskAudit[]** An array of TaskAudit objects representing the audit logs of the task. See [TaskAudit](#) for information about the structure of this object.

DELETE Audit Entries to Rollback Task

This REST request enables you to roll back the task to any available audit point in the task's audit history.

The HTTP request is formed as follows:

```
DELETE /rest/pub/opentask/taskID/audit/auditEntryID
```

Input Parameters

auditEntryID **Integer** The ID of the audit log entry that you want to roll the task back to.

taskID **String** The unique identifier that the Task Engine assigns to the task instance.

Output Parameters

None.

GET or POST a Form Flow Notification

This REST request notifies a waiting step in a task workflow that the workflow is completed.

For example, when the final task step in a task workflow completes, that step waits for notification. However, in this case there are no more steps in the workflow to notify the waiting step. You can configure a following process service activity step to call this service and pass a result to the waiting step that indicates the workflow is complete. This enables the process to execute to completion.

For more information about task workflows, see “Working with Task Workflows” in the *webMethods BPM Task Development Help*.

Use this HTTP request if no optional parameters are required:

```
GET /rest/pub/opentask/formFlowNotify/correlationID
```

Use this HTTP request if you want to include optional parameters in the request body:

```
POST /rest/pub/opentask/formFlowNotify/correlationID
```

Input Parameters

correlationID **String** Use a task workflow correlation ID to synchronize the communication between the waiting step and the notifying step. This ensures correct data flow through the process. You can define any value but the task workflow correlation ID must be unique within the Process Engine environment.

ImportantThe task workflow correlation ID is completely different from and unrelated to the standard document correlation ID often used in process implementation.

localOnly **String** Optional. Set to:

- `true` if the service is not being used in a cluster environment.
- `false` (default) if the service is being used in a clustered environment.

result **String** Optional. This can be any value to be passed back to the waiting component. Typically, you use this field to pass a status code to the waiting step to indicate the end of the workflow,

which enables the waiting step to take an action upon receiving the result.

Output Parameters

None.

Usage Notes

Technically, you can also use this service within a workflow to notify a waiting task activity that the next task activity in the workflow is instantiated. However, the Java API `com.webmethods.portal.service.task.ITaskFormFlowService` is recommended for this task. To read the Java docs for this component, see the *webMethods CAF and My webMethods Server Java API Reference*, available from the “[Software AG Documentation web site](#)” under My webMethods Server in the corresponding webMethods Product Suite release number.

Sample Code Available

For more information about implementing a task workflow, you can examine and deploy a sample task application, process model, and Integration Server package that support a very simple loan application process. You can find the task workflow code samples on the Software AG Community web site.

A description of the sample code and its behavior can be found in the topic “Working with Task Workflows” in the *webMethods BPM Task Development Help*.

DELETE Task Comment

This REST request deletes the specified task comment from the specified task instance as well as all associated attachments.

The HTTP request is formed as follows:

```
DELETE /rest/pub/opentask/taskID/comments/commentID[?user=value]
```

Input Parameters

<i>commentID</i>	String The unique identifier of the comment to be deleted.
<i>taskID</i>	String The unique identifier that the Task Engine assigns to the task instance.
<i>user</i>	String Optional. The user ID of the My webMethods Server user on whose behalf this operation executes.

Output Parameters

None.

GET Task Comments

This REST request returns either a specific comment or all the comments in a task instance.

To return a specific comment, the HTTP request is formed as follows:

```
GET /rest/pub/opentask/taskID/comments/commentID
```

To return all comments, the HTTP request is formed as follows:

```
GET /rest/pub/opentask/taskID/comments
```

Input Parameters

taskID **String** The unique identifier that the Task Engine assigns to the task instance.

Output Parameters

taskComments **TaskComment[]** An array of TaskComment objects that represent the comments retrieved from the task instance. See [“TaskComment” on page 156](#) for the structure of this data.

PUT Updates into a Task Comment

This REST request updates a comment in a task instance.

The HTTP request is formed as follows:

```
PUT /rest/pub/opentask/taskID/comments/commentID [ ?param1=value ]  
    [ &param2=value ] . . . [ &param4=value ]
```

Input Parameters

attachments **TaskAttachment[]**. An array of TaskAttachment objects representing any attachments to be added to the comment being updated.

<i>commentID</i>	String The unique identifier of a comment to be deleted.
<i>deleteAttachments</i>	String[] Optional. An array of IDs of any attachments to be deleted from the comment.
<i>taskID</i>	String The unique identifier that the Task Engine assigns to the task instance.
<i>text</i>	String Optional. The updated text for this comment.
<i>user</i>	String Optional. The user ID of the My webMethods Server user on whose behalf this operation executes.

Output Parameters

None.

POST Task Comment

This REST request adds a comment to a task instance. Attachments can be included. Use this HTTP request and include optional parameters in the request body:

```
POST /rest/pub/opentask/taskID/comments/
```

Input Parameters

<i>attachments</i>	TaskAttachment [] Optional. An array of TaskAttachment objects representing any attachments to be added to the comment being added. See “TaskAttachment” on page 153 for the structure of this data.
<i>html</i>	Boolean Optional. Set to: <ul style="list-style-type: none">■ <code>true</code> if the comment text contains HTML markup.■ <code>false</code> (default) if the comment text is to be treated as plain text.
<i>taskID</i>	String The unique identifier that the Task Engine assigns to the task instance.
<i>text</i>	String Optional. The updated text for this comment.

user **String** Optional. The user ID of the My webMethods Server user on whose behalf this operation executes.

Output Parameters

commentID **String** The unique identifier that the Task Engine assigns to the newly added comment.

DELETE Task Attachment

This REST request deletes the specified task attachment from a task instance.

The HTTP request is formed as follows:

```
DELETE /rest/pub/opentask/taskID/attachments/attachmentID [ ?user ]
```

Input Parameters

attachmentID **String** The unique identifier of the attachment to be deleted.

taskID **String** The unique identifier that the Task Engine assigns to the task instance.

user **String** Optional. The user ID of the My webMethods Server user on whose behalf this operation executes.

Output Parameters

None.

GET Task Attachments

This REST request returns either a specific attachment or all the attachments in a task instance.

To return a specific attachment, the HTTP request is formed as follows:

```
GET /rest/pub/opentask/taskID/attachments/attachmentID
```

To return all attachments, the HTTP request is formed as follows:

```
GET /rest/pub/opentask/taskID/attachments
```

Input Parameters

taskID **String** The unique identifier that the Task Engine assigns to the task.

Output Parameters

taskAttachments **TaskAttachment[]** An array of TaskAttachment objects that represent the attachments in the task instance. See [“TaskAttachment” on page 153](#) for the structure of this data.

PUT Task Attachment

This REST request updates the specified task attachment in a task instance.

The HTTP request is formed as follows:

```
PUT /rest/pub/opentask/taskID/attachments/attachmentID [ ?attachment=value ]  
    [ &encoding=value ] [ &user=value ]
```

Input Parameters

attachment **TaskAttachment** The TaskAttachment object representing the new attachment to be used to update the existing attachment. See [“TaskAttachment” on page 153](#) for the structure of this data.

attachmentID **String** The unique identifier of the attachment to be updated.

encoding **String** Optional. Specify the encoding type for the file. Set to:

- binary
- text

taskID **String** The unique identifier that the Task Engine assigns to the task instance.

user **String** Optional. The user ID of the My webMethods Server user on whose behalf this operation executes.

Output Parameters

None.

POST Task Attachment

This REST request adds the specified attachment to a task instance.

Use this HTTP request and include optional parameters in the request body:

```
POST /rest/pub/opentask/taskID/attachments
```

Input Parameters

<i>attachment</i>	TaskAttachment The TaskAttachment object representing the new attachment to be used to update the existing attachment. See “TaskAttachment” on page 153 for the structure of this data.
<i>encoding</i>	String Optional. Specify the encoding type for the file. Set to: <ul style="list-style-type: none">■ binary■ text
<i>taskID</i>	String The unique identifier that the Task Engine assigns to the task instance.
<i>user</i>	String Optional. The user ID of the My webMethods Server user on whose behalf this operation executes.

Output Parameters

None.

4 Using the Task Engine Web Service API

■ Overview	131
■ Task Service URLs	131
■ Extracting the WSDL for the Task Web Service	131
■ Using the Task Web Service Operations	132
■ Summary of Available Operations	133
■ addTagsToTask	134
■ addTaskAttachment	135
■ addTaskComment	136
■ countTasksIndexed	136
■ deleteTagsFromTask	138
■ deleteTask	138
■ deleteTaskAttachment	139
■ deleteTaskComment	139
■ formFlowTaskNotify	140
■ getTask	141
■ getTaskAttachments	142
■ getTaskAudit	142
■ getTaskComments	143
■ getTaskExpertList	143
■ listTagsByTask	143
■ queueTask	144
■ rollbackTask	145
■ searchTaskFields	145
■ searchTasks	147
■ searchTasksIndexed	148
■ updateTask	149

■ updateTaskAttachment	151
■ updateTaskComment	151
■ updateTaskTags	152
■ Summary of Data Structures	152
■ TaskAttachment	153
■ TaskAudit	154
■ TaskComment	156
■ TaskData	156
■ TaskInfo	158
■ Task	163
■ TaskSearchQuery	163
■ TaskSearchQueryTerm	165
■ TaskSearchQueryV2	168

Overview

The Web service API enables you to add, delete, locate, and update tasks on a webMethods Task Engine from a client program.

The Task Engine hosts two versions of the task Web service, one version for .NET clients and the other for regular Web-service clients. Both versions of the Web service provide the same set of operations, however, the schemas for the underlying data structures differ slightly between the two. The .NET version of the service uses structures that are compatible with .NET clients.

Note: If you want to manipulate tasks from the Integration Server environment, use the built-in Java services described in [“Using the Task Engine Built-in Services” on page 11](#). Do not use the Web service operations described in this chapter with Web services created within Integration Server.

Task Service URLs

The task Web service is hosted at the following endpoints on the machine where the Task Engine is installed.

- For regular Web service clients, the task Web service is hosted at:
`http://hostName:port/services/bizPolicy/task`
- For .NET clients, the task Web service is hosted at: `http://hostName:port/services/bizPolicy/taskDotNet`

Where:

hostName specifies the machine where My webMethods Server and the Task Engine are running.

port specifies the port on which My webMethods Server listens for http requests. The default port is 8585, however, this port assignment is configurable. If you do not know which port your server uses, contact your My webMethods Server administrator.

Example:`http://server:8585/services/bizPolicy/task`

Extracting the WSDL for the Task Web Service

To obtain the task WSDL

1. log on to My webMethods Server as sysadmin.
2. Navigate to one of the following URLs:

- For .NET clients: `http://server:port/services/bizPolicy/taskDotNet?wsdl`
- For other clients: `http://server:port/services/bizPolicy/task?wsdl`

Using the Task Web Service Operations

To use the task Web service operations described below, your client program must log on to My webMethods Server with a user ID that has appropriate functional and task access privileges on the Task Engine. Operations that your client program performs are executed under this user ID. If the user ID does not have the required function and task access privileges, the operation fails and an exception is generated.

You can specify an alternative user ID for the operation by setting the optional *user* input parameter, which is available on all operations in the task Web service. This parameter enables a client program to perform an operation under a user ID that is different from the one it used to log on to My webMethods Server.

Important: To use the *user* parameter, your client program *must* log on to the My webMethods Server with a user ID that belongs to the "Admin Role" role. If your program does not belong to the "Admin Role" role and it requests an operation using the *user* parameter, the operation fails and an exception is generated.

When a client requests an operation with the *user* parameter, the Task Engine executes the operation as though it were requested by the specified user. For example, if your client program logs on as "mrussel," but executes the searchTask operation under the user ID "rkosi," the Task Engine searches only tasks the user "rkosi" can access.

My webMethods Server updates a user's role membership only when the user interactively logs in into My webMethods Server. When impersonating a user with the Task Engine APIs, there is no interactive login of this user to My webMethods Server. Therefore, the Task Engine itself handles role membership changes and updates for these impersonated users.

The Task Engine updates a user's role membership when:

- A specified time has passed since last time the user ID was impersonated. The default value is 30 minutes.
- A specified time has passed since the last time the user's role membership was updated. The default value is 24 hours (session total time-to-live).

These default time periods can be modified with the following environment settings:

```
-Dtask.remote.session.timeout=<the time period in seconds between updates
of user role information. The session is not invalidated or expired.>
-Dtask.remote.session.ttl=<user session time-to-live in seconds>
```

Important: It is important to understand that the `-Dtask.remote.session.timeout` setting does *not* affect the duration of the actual session. The only purpose

of the setting is to specify the time interval between updates to the impersonated user's role membership.

- For more information about setting these options, and assigning task functional privileges and task access privileges, see the PDF publication *webMethods Task Engine User's Guide*.
- For more information about working with roles and users, see the PDF publication *webMethods Integration Server Administrator's Guide*.

Summary of Available Operations

The following table lists the operations that the task service provides.

Operation	Description
addTagsToTask	Adds tags to a task.
addTaskAttachment	Adds the specified attachment to a task.
addTaskComment	Adds a comment to a task.
countTasksIndexed	Returns the total count of tasks that match the specified search criteria in a searchTasksIndexed search.
deleteTagsFromTask	Deletes all tags from a task.
deleteTask	Deletes an instance of a task from the Task Engine.
deleteTaskAttachment	Deletes the specified task attachment.
deleteTaskComment	Deletes a comment from a task.
formFlowTaskNotify	Service that notifies a waiting step in a task workflow that the workflow is completed.
getTask	Returns the TaskInfo object, and optionally the TaskData object, for a specified task.
getTaskAttachments	Returns all the attachments in a task.
getTaskAudit	Returns the audit log of all of the operations performed on a task.

Operation	Description
getTaskComments	Returns all the comments in a task.
getTaskExpertList	Returns a list of experts associated with the tags.
listTagsByTask	Lists all the tags of a task.
queueTask	Adds a task instance to the Task Engine.
rollbackTask	Enables you to roll back the task to any available audit point in the task's audit history.
searchTasks	Returns an array of tasks that match specified search criteria.
searchTaskFields	Returns a list of tasks that match specified field search criteria.
searchTasksIndexed	Returns a subset of tasks that match the specified search criteria against indexed business data.
updateTask	Modifies an existing task instance on the Task Engine. You use this operation to change information in the task's TaskInfo and TaskData objects. This service applies only to tasks with Active, Error, and Suspended status. For information about updating tasks with other statuses, see <i>Usage Notes</i> .
updateTaskAttachment	Updates the specified task attachment.
updateTaskComment	Updates the specified task comment.
updateTaskTags	Updates the tags of a task.

addTagsToTask

Adds the specified tags to a task.

Input Parameters

<i>taskID</i>	String The unique identifier that the Task Engine assigns to the task.
<i>tags</i>	String Comma-separated list of text to be used to define tags. No character limit enforced. Alphanumeric characters, spaces, underscores, and dashes are allowed.

Output Parameters

None.

addTaskAttachment

Adds the specified attachment to a task.

Input Parameters

<i>attachment</i>	TaskAttachment The TaskAttachment object representing the attachment to be added.
<i>encoding</i>	String Specify the encoding type for the file. Valid values include: <ul style="list-style-type: none">■ binary■ text
<i>taskID</i>	String The unique identifier that the Task Engine assigns to the task.
<i>user</i>	String Optional. The user ID of the My webMethods Server user on whose behalf this operation executes.

Output Parameters

None.

addTaskComment

Adds a comment to a task.

Input Parameters

<i>attachments</i>	TaskAttachment [] Optional. An array of TaskAttachment objects representing the attachments to be added to the comment being added.
<i>html</i>	Boolean Optional. Set to: <ul style="list-style-type: none">■ <code>true</code> if the comment text contains HTML markup.■ <code>false</code> (default) if the comment text is to be treated as plain text.
<i>taskID</i>	String The unique identifier that the Task Engine assigns to the task.
<i>text</i>	String Optional. The updated text for this comment.
<i>user</i>	String Optional. The user ID of the My webMethods Server user on whose behalf this operation executes.

Output Parameters

<i>commentID</i>	String The unique identifier that the Task Engine assigns to the newly added comment.
------------------	--

countTasksIndexed

Returns the total count of tasks that match the specified search criteria in a [searchTasksIndexed](#) search.

Specify the *toIndex* and *fromIndex* parameters used in the [searchTasksIndexed](#) service to avoid requesting an index beyond the size of the search results.

Input

<i>businessData</i>	Boolean Optional. Indicates whether the search query operation includes indexed business data when processing the search query. Use this property when
---------------------	---

your search query contains search terms that reference indexed business data fields.

Set to:

- `true` to include business data in the search query processing. An error occurs if `businessData` is set to `true` and the task type does not contain any indexed fields.
- `false` when your search query does not require processing of business data or when your task type does not use any defined indexed business fields.

taskSearchQuery

TaskSearchQueryV2 Specifies the `TaskSearchQueryV2` object; this is the search criteria. If `TaskSearchQueryV2` is null, an error occurs.

user

String Optional. Specifies the user ID of the My webMethods Server user for which the operation executes. If *user* is not specified, the administrative user ID under which your client program logged on as is used. When you use this parameter, the search is limited to tasks that are accessible by the user.

userTasks

Boolean Optional. Indicates whether the operation searches all tasks or only the user's inbox. This parameter is over-ridden if the *user* parameter contains a value.

Set to:

- `true` to execute the search only on the user's inbox.
- `false` to search accessible user tasks; this is the default value.

By default, only active tasks are returned. This can be overridden with the *showNonActiveTasks* parameter in [“TaskSearchQuery” on page 163](#).

Output Parameters

totalCount

Returns the total number of tasks that match the query.

deleteTagsFromTask

Deletes all tags associated with a task.

Input Parameters

<i>taskID</i>	String The unique identifier that the Task Engine assigns to the task.
---------------	---

Output Parameters

None.

deleteTask

Deletes an instance of a task from the Task Engine.

Input

<i>taskID</i>	String The ID that identifies the task that you want to delete.
<i>user</i>	String Optional. The user ID of the My webMethods Server user on whose behalf this operation executes. If <i>user</i> is not specified, the administrative user ID under which your client program logged on is used.

Output

None.

Usage Notes

To delete a task using this operation, you must supply the ID of the task that you want to delete. To obtain this ID, use the [searchTasks](#) operation to locate the task and then extract the task ID from the result set that the [searchTasks](#) operation returns.

To delete a task successfully, the user ID in *user* must have permission to access the specified task. If the supplied ID does not have access to the task, or if the specified task does not exist, an exception is thrown.

deleteTaskAttachment

Deletes the specified task attachment.

Input Parameters

<i>attachmentID</i>	String The unique identifier of the attachment to be deleted.
<i>taskID</i>	String The unique identifier that the Task Engine assigns to the task.
<i>user</i>	String Optional. The user ID of the My webMethods Server user on whose behalf this operation executes.

Output Parameters

None.

deleteTaskComment

Deletes a comment from a task instance as well as all associated attachments.

Input Parameters

<i>commentID</i>	String The unique identifier of the comment to be deleted.
<i>taskID</i>	String The unique identifier that the Task Engine assigns to the task.
<i>user</i>	String Optional. The user ID of the My webMethods Server user on whose behalf this operation executes.

Output Parameters

None.

formFlowTaskNotify

Service that notifies a waiting step in a task workflow that the workflow is completed.

For example, when the final task step in a task workflow completes, that step waits for notification. However, in this case there are no more steps in the workflow to notify the waiting step. You can configure a following process service activity step to call this service and pass a result to the waiting step that indicates the workflow is complete. This enables the process to execute to completion.

For more information about task workflows, see “*Working with Task Workflows*” in the *webMethods BPM Task Development Help*.

Input Parameters

correlationID **String** Use a task workflow correlation ID to synchronize the communication between the waiting step and the notifying step. This ensures correct data flow through the process. You can define any value but the task workflow correlation ID must be unique within the Process Engine environment.

Important! The task workflow correlation ID is completely different from and unrelated to the standard document correlation ID often used in process implementation.

result **String** Optional. This can be any value to be passed back to the waiting component. Typically, you use this field to pass a status code to the waiting step to indicate the end of the workflow, which enables the waiting step to take an action upon receiving the result.

localOnly **String** Optional. Set to:

- `true` if the service is not being used in a cluster environment.
- `false` (default) if the service is being used in a clustered environment.

Output Parameters

None.

Usage Notes

Technically, you can also use this service within a workflow to notify a waiting task activity that the next task activity in the workflow is instantiated. However, the Java API `com.webmethods.portal.service.task.ITaskFormFlowService` is recommended for this task. To

read the Java docs for this component, see the *webMethods CAF and My webMethods Server Java API Reference*, available from the “[Software AG Documentation web site](#)” under My webMethods Server in the corresponding webMethods Product Suite release number.

Sample Code Available

For more information about implementing a task workflow, you can examine and deploy a sample task application, process model, and Integration Server package that support a very simple loan application process. You can find the code samples in your My webMethods Server installation at: *Software AG_directory/MWS/components/samples/workflow*

A description of the sample code and its behavior can be found in the topic “Working with Task Workflows” in the *webMethods BPM Task Development Help*.

getTask

Returns the [TaskInfo](#) object, and optionally the [TaskData](#) object, for a specified task.

The TaskInfo object contains standard information about a task, including its status, expiration date, and the list of users to which it is assigned. The TaskData object contains the business data that is associated with the task.

Input

<i>includeTaskData</i>	Boolean Optional. Specifies whether you want to retrieve the TaskData object as well as the TaskInfo object. Set to true to retrieve TaskData in addition to TaskInfo. Default is false.
<i>taskID</i>	String The ID that identifies the task that you want to retrieve.
<i>user</i>	String Optional. The user ID of the My webMethods Server user on whose behalf this operation executes. If <i>user</i> is not specified, the administrative user ID under which your client program logged on is used.

Output

<i>Task</i>	Task A Task object containing the task's TaskInfo object and, optionally, its TaskData object. See “ Task ” on page 163 for a description of the fields in this object. The TaskData object is included in <i>Task</i> only if <i>includeTaskData</i> is set to true.
-------------	---

Usage Notes

To get a task using this operation, you must supply the ID of the task that you want to retrieve. To obtain this ID, use the [searchTasks](#) operation to locate the task and then extract the task ID from the result set that the [searchTasks](#) operation returns.

To retrieve a task successfully, the user ID specified in *user* must have permission to access that task. If the supplied ID does not have access to the task, or if the specified task does not exist, an exception is thrown.

getTaskAttachments

Returns all the attachments in a task.

Input Parameters

taskID **String** The unique identifier that the Task Engine assigns to the task.

Output Parameters

taskAttachments **TaskAttachment[]** An array of [TaskAttachment](#) objects that represent the attachments in the task.

getTaskAudit

Returns the audit log of all of the operations performed on a task.

Input Parameters

taskID **String** The unique identifier that the Task Engine assigns to the task

Output Parameters

taskAudits **TaskAudit[]** An array of [TaskAudit](#) objects representing the audit logs of the task.

getTaskComments

Returns all the comments in a task.

Input Parameters

taskID **String** The unique identifier that the Task Engine assigns to the task.

Output Parameters

taskComments **TaskComment[]** An array of TaskComment objects that represent the comments in the task

getTaskExpertList

Returns a list of experts associated with the specified tags.

Input Parameters

tags **String** Comma-separated list of tags.

Output Parameters

guidance **Guidance[]** An array of Guidance objects containing the names, ratings, and user IDs of the experts for the specified tags. See [pub.task:Guidance](#) for a description of the fields in this document.

listTagsByTask

This service returns a list of tags associated with a taskID.

Input Parameters

taskID **String** The unique identifier that the Task Engine assigns to the task.

Output Parameters

Tags **String Array** List of tags associated with the taskID.

queueTask

Adds a task instance to the Task Engine.

Input Message

taskData **Map** Optional. The business data, if any, associated with this task. See [“TaskData” on page 156](#) for a description of the Map object used by *taskData*.

taskInfo **TaskInfo** Optional. A complex structure that contains standard information about the task (for example, name, expiration date, list of users to which the task is assigned). See [“TaskInfo” on page 158](#) for a description of the fields in this object.

taskTypeID **String** Specifies the type of task that you want to queue. Each task type that is deployed on the Task Engine has a unique ID. This ID is assigned by the developer when he or she creates a task application using Software AG Designer.

If you do not know the ID for the task type that you want to queue, contact the Task Engine administrator. An administrator can obtain a list of the task types by viewing the **Task Configuration** panel on the Task Engine Administration page in My webMethods.

Task type IDs are case-sensitive. The ID in *taskTypeID* must exactly match the ID as it is specified on the Task Engine.

user **String** Optional. The user ID of the My webMethods Server user on whose behalf this operation executes.

If *user* is not specified, the user ID under which your client program logged on is used.

Output Message

taskID **String** The identifier that the Task Engine assigns to the task.

Usage Notes

The user ID specified in *user* must have permission to queue tasks on the Task Engine. If the supplied ID does not have this permission, an exception is generated.

rollbackTask

Enables you to roll back the task to any available audit point in the task's audit history.

Input Parameters

<i>auditEntryID</i>	Integer The ID of the audit log entry that you want to roll the task back to.
<i>taskID</i>	String The unique identifier that the Task Engine assigns to the task.

Output Parameters

None.

searchTaskFields

Returns a list of tasks that match specified field search criteria.

Note: This API can return only primitive types (for example, Strings, Numbers, Dates) or lists (arrays) of primitives. It cannot return complex objects such as all task business data.

Input Message

<i>includeTaskData</i>	String Optional. Specifies whether the operation returns the TaskData object as well as the TaskInfo object for each task matching the search criteria. Set to: <ul style="list-style-type: none">■ <code>true</code> to include the TaskData in the result set.■ <code>false</code> (default) to include only the TaskInfo in the result set.
<i>taskSearchQuery</i>	TaskSearchQuery The TaskSearchQuery object, which specifies the search criteria. See " TaskSearchQuery " on page 163 for a description of the fields in this object.

If *TaskSearchQuery* is null, all tasks for *user* are returned.

user

String Optional. The user ID of the My webMethods Server user on whose behalf this operation executes.

If *user* is not specified, the administrative user ID under which your client program logged on is used.

Note: Only tasks to which *user* has access are searched.

fields

String List of tasks fields that are to be returned by this service. For information about specifying fields, see the *fields* and *operator* elements in [TaskSearchQueryTerm](#). For example, suppose you want to return the following fields as result of the query:

- taskID
- The business data field *orderID* from task data documents.

You would then pass in following field specifications:

```
#{currentTask.taskInfo.taskID}
```

```
#{currentTask.taskData.order.orderID}
```

This API returns the values for these fields only.

Output Parameters

Task Field

Document List Result set containing the tasks that match the search criteria in *TaskSearchQuery*. Each document in the result set has the following structure:

- *name* **String** Name of the task.
- *value* **Object** The value of the field contents for the specified field(s).

That is, the service returns a collection of tasks that match the query. For each task, it returns a collection of specified fields.

The purpose of this API is to ease performance issues with `searchTasks()`. It is quite common that only few `taskInfo` fields along with some business data fields are returned. When using `searchTasks()` it returns the entire `TaskInfo` plus `TaskData` documents; the `TaskData` could be very large. This could cause the SOAP response size to become an issue, severely limiting performance of this service. By using `searchTaskFields()` instead, the user can choose to return only a few fields which can help improve performance.

Usage Notes

If no tasks match the search criteria, an empty list is returned.

searchTasks

Returns an array of tasks that match specified search criteria.

Input Message

<i>includeTaskData</i>	<p>Boolean Optional. Specifies whether the operation returns the TaskData object as well as the TaskInfo object for each task matching the search criteria.</p> <p>Set to:</p> <ul style="list-style-type: none"> ■ <code>true</code> to include the TaskData in the result set. ■ <code>false</code> (default) to include only the TaskInfo in the result set.
<i>taskSearchQuery</i>	<p>TaskSearchQuery The TaskSearchQuery object, which specifies the search criteria. See “TaskSearchQuery” on page 163 for a description of the fields in this object.</p> <p>If <i>TaskSearchQuery</i> is null, all tasks for <i>user</i> are returned.</p>
<i>user</i>	<p>String Optional. The user ID of the My webMethods Server user on whose behalf this operation executes.</p> <p>If <i>user</i> is not specified, the administrative user ID under which your client program logged on is used.</p> <div style="background-color: #f0f0f0; padding: 5px; margin-top: 10px;"> <p>Note: Only tasks to which <i>user</i> has access are searched.</p> </div>
<i>userTasks</i>	<p>Boolean Optional. Specifies whether the operation searches all tasks or just the user's inbox. This parameter is over-ridden if the <i>user</i> parameter contains a value.</p> <p>Set to:</p> <ul style="list-style-type: none"> ■ <code>true</code> to execute the search only on the user's inbox. ■ <code>false</code> to search accessible user tasks; this is the default value. <p>By default, only active tasks are returned. This can be overridden with the <i>showNonActiveTasks</i> parameter in “TaskSearchQuery” on page 163.</p>

Output Parameters

tasks **Task[]** An array of Task objects representing the tasks that matched the search criteria in *TaskSearchQuery*. See [“Task” on page 163](#) for a description of the fields in a Task object.

Usage Notes

If no tasks match the search criteria, an empty array is returned.

searchTasksIndexed

Returns a subset of tasks that match the specified search criteria against indexed business data.

The query must contain a search term specifying the *taskTypeID* when searching for business data fields. The Task Engine uses the *taskTypeID* to identify the specific index table to use. You can search for multiple task types in a single query by specifying multiple *taskTypeID* values when you define the search criteria. If you are not searching for business data fields, then *taskTypeID* is not required.

If you are working with standard business data fields use [“searchTasks” on page 147](#).

Input Message

businessData **Boolean** Optional. Indicates whether the search query operation includes indexed business data when processing the search query. Use this property when your search query contains search terms that reference indexed business data fields.

Set to:

- `true` to include business data in the search query processing.
- `false` when your search query does not require processing of business data or when your task type does not use any defined indexed business fields.

The default value for *businessData* is `false`.

taskSearchQuery **TaskSearchQueryV2** Specifies the [TaskSearchQueryV2](#) object that provides the search criteria. If *taskSearchQueryV2* is null, an error occurs.

user **String** Optional. Specifies the user ID of the My webMethods Server user for which the operation executes. If *user* is not specified, the administrative user ID under which your client

program logged on as is used. When you use this parameter, the search is limited to tasks that are accessible by the user.

userTasks

Boolean Optional. Indicates whether the operation searches all tasks or only the user's inbox. This parameter is over-ridden if the *user* parameter contains a value.

Set to:

- `true` to execute the search only on the user's inbox.
- `false` to search accessible user tasks; this is the default value.

By default, only active tasks are returned. This can be overridden with the *showNonActiveTasks* parameter in "[TaskSearchQuery](#)" on [page 163](#).

Usage Notes

The search query must contain a search term specifying the *taskTypeID*; this is required to search business fields and page the results.

The [TaskSearchQuery](#) data structure referenced in [TaskSearchQueryV2](#) must contain a value for *taskTypeID* in the field's element to identify the search target. You can search for multiple task types in a single query by specifying multiple *taskTypeID* values. When specifying multiple *taskTypeID*s, enter them as a string array.

Output Parameters

tasks

Task Specifies an array of Task objects representing the tasks that match the search criteria in [TaskSearchQueryV2](#).

If no tasks match the search criteria, an empty array is returned.

updateTask

Modifies an existing task instance on the Task Engine. You use this operation to change information in the task's [TaskInfo](#) and [TaskData](#) objects. This service applies only to tasks with status Active, Error, and Suspended. For information about updating tasks with other statuses, see *Usage Notes*.

The [TaskInfo](#) object contains standard information about a task, including its status, expiration date, and the list of users to which it is assigned. The [TaskData](#) object contains the business data that is associated with the task.

This is the operation a client program would use if it needed to change the status of a task. For example, if a program wanted to end a task, it would use this operation to change the *status* field in the task's [TaskInfo](#) object to "completed" or "cancelled" as appropriate.

Input Message

<i>taskData</i>	Map The Map object that contains the business data that you want to assign to the task. See “TaskData” on page 156 for a description of the Map object used by <i>taskData</i> .
<i>taskID</i>	String The ID that identifies the task that you want to modify.
<i>taskInfo</i>	TaskInfo Object that specifies the changes you want to make to the data in the task's TaskInfo object. See “TaskInfo” on page 158 for a description of the fields in this object.
<i>user</i>	String Optional. The user ID of the My webMethods Server user on whose behalf this operation executes. If <i>user</i> is not specified, the administrative user ID under which your client program logged on is used.

Output Message

None.

Usage Notes

To modify a task using this operation, you must supply the ID of the task that you want to update. To obtain this ID, use the [searchTasks](#) operation to locate the task and then extract the task ID from the result set that the searchTasks operation returns.

To modify a task successfully, the user ID in *user* must have permission to access the specified task. If the supplied ID does not have access to the task, or if the specified task does not exist, an exception is thrown.

When you use this operation to update information in the task's TaskInfo object, the Task Engine modifies *only* the fields to which you have assigned values in *taskInfo*. All other fields in the target task retain their existing values. For example, to change just the name assigned to a task, you would set only the *name* field in the *TaskInfo* object that you pass to this service.

When you use this operation to update information in the task's TaskData object, the object that you pass to the service in *taskData* replaces the task's existing TaskData object in its entirety.

When updating task statuses with this service, the following limitations apply:

- If the current status of the task is Error, you can change it only to Completed.
- If the current status of the task is Suspended, you cannot set it to Completed or Expired.

This service applies only to tasks with status Active, Error, and Suspended. To enable the service to update tasks with other statuses, set the additional JVM property – `Dupdate.completed.task` to `true`, as described in *webMethods Task Engine User's Guide*.

Tasks with status Error that belong to a process instance cannot be updated, regardless of Task Engine optional settings.

updateTaskAttachment

Updates the specified task attachment.

Input Parameters

<i>attachment</i>	TaskAttachment The TaskAttachment object representing the updated attachment to be used to update the existing attachment.
<i>attachmentID</i>	String The unique identifier of the attachment to be updated.
<i>encoding</i>	String Specify the encoding type for the file. Valid values include: <ul style="list-style-type: none">■ <code>binary</code>■ <code>text</code>
<i>taskID</i>	String The unique identifier that the Task Engine assigns to the task.
<i>user</i>	String Optional. The user ID of the My webMethods Server user on whose behalf this operation executes.

Output Parameters

None.

updateTaskComment

Updates a comment in a task.

Input Parameters

<i>attachments</i>	TaskAttachment[] An array of TaskAttachment objects representing any attachments to be added to the comment being updated.
--------------------	---

<i>commentID</i>	String The unique identifier of the comment to be deleted.
<i>deleteAttachments</i>	String[] An array of IDs of the attachments to be deleted from the comment
<i>taskID</i>	String The unique identifier that the Task Engine assigns to the task.
<i>text</i>	String Optional. The updated text for this comment.
<i>user</i>	String Optional. The user ID of the My webMethods Server user on whose behalf this operation executes.

Output Parameters

None.

updateTaskTags

Updates the tags of a task.

Input Parameters

<i>taskID</i>	String The unique identifier that the Task Engine assigns to the task.
<i>tags</i>	String Comma-separated list of tags associated with a <i>taskID</i> . No character limit enforced. Alphanumeric characters, spaces, underscores, and dashes are allowed.

Output Parameters

None.

Summary of Data Structures

The following table lists the complex data structures that the task service uses.

Data Structure	Description
TaskAttachment	Document type representing an attachment.
TaskAudit	Document type containing information returned as output from getTaskAudit .
TaskComment	Document type representing a comment in a task.
TaskData	Holds business data relating to the task. The underlying data structures used to hold the business data in a TaskData object is different for regular Web-service clients and .NET clients.
TaskInfo	Contains standard information about a task. You specify certain fields in TaskInfo when you start a task using the queueTask operation. After a task is queued, you can modify fields in TaskInfo using the updateTask operation.
Task	Data object returned as output from the getTask and searchTasks operations. This data structure contains the TaskInfo and TaskData objects associated with a task.
TaskSearchQuery	Used as input to the searchTasks operation.
TaskSearchQueryTerm	Contains search criteria for the searchTasks operation.
TaskSearchQueryV2	Use the TaskSearchQueryV2 data type as input to searchTasks .

TaskAttachment

Document type representing an attachment.

Parameters

<i>base64Data</i>	String A Base64 encoded string representation of the data for the attachment. [Need examples or more info?]
<i>contentType</i>	String[] Content-type of the data in the attachment.
<i>downloadLink</i>	String A link from My webMethods Server for downloading the attachment.
<i>fileName</i>	String Name of the file in the attachment.
<i>iconURL</i>	String URL of the icon to be used to represent the attachment in the task user interface.
<i>id</i>	String Read-only. Unique ID of the attachment.
<i>lastModifiedDate</i>	Date The date and time of last modification of the attachment.
<i>name</i>	String Name of the attachment.
<i>contentLength</i>	Long Length of the data in the attachment.
<i>deletable</i>	Boolean Set to: <ul style="list-style-type: none">■ <code>true</code> if the attachment can be deleted by a user with appropriate permissions. Default.■ <code>false</code> if the attachment cannot be deleted by a user.
<i>updatable</i>	Boolean Set to: <ul style="list-style-type: none">■ <code>true</code> if the attachment can be updated by a user with appropriate permissions. Default.■ <code>false</code> if the attachment cannot be updated by a user.

TaskAudit

Document type containing information returned as output from [getTaskAudit](#).

Parameters

<i>createdDate</i>	Date The date and time the operation was applied to the audited task.
<i>difference</i>	String[] An array of strings representing the changes made as part of an audit log entry.
<i>sourceURI</i>	String URI of the source of this change.
<i>TaskURI</i>	String URI of the audited task.
<i>id</i>	Integer Unique ID of this audit log entry.
<i>operation</i>	<p>Integer Numerical representation of the operation performed. See Usage Notes below for more information.</p> <p>Selected TASK_OPERATION_* fields from com/webmethods/caf/faces/data/task/ITaskConstants are valid values for this parameter, as described in Usage Notes below. For more information, see the Javadoc in <i>webMethods CAF and My webMethods Server Java API Reference</i>, available from the “Software AG Documentation web site” under My webMethods Server in the corresponding webMethods Product Suite release number.</p>
<i>taskVersionNumber</i>	Integer Version number of the audited task.

Usage Notes

The integer values returned in the *operation* parameter represent the following operations:

```

0 = TASK_OPERATION_CREATED
1 = TASK_OPERATION_RESUME
2 = TASK_OPERATION_SUSPEND
3 = TASK_OPERATION_UPDATE
5 = TASK_OPERATION_ERROR
9 = TASK_OPERATION_DISTRIBUTION_RULE_FIRED
10 = TASK_OPERATION_TRIGGER_RULE_FIRED
11 = TASK_OPERATION_SCHEDULE_RULE_FIRED
12 = TASK_OPERATION_DELEGATION_RULE_FIRED
13 = TASK_OPERATION_REPLY
14 = TASK_OPERATION_REVERTED
16 = TASK_OPERATION_ACCEPTED
17 = TASK_OPERATION_UNACCEPTED

```

TaskComment

Document type representing a comment in a task.

Parameters

<i>attachments</i>	TaskAttachment [] An array of attachments to be enclosed inside the comment.
<i>ID</i>	String Unique ID of the comment.
<i>displayText</i>	String Displayable representation of the text associated with the comment.
<i>text</i>	String Text associated with the comment.
<i>timeStamp</i>	Date The date and time of last modification of the comment.
<i>userDisplayName</i>	String Displayable representation of the name of user associated with the comment.
<i>userID</i>	String User name of the user associated with the comment.

TaskData

Holds business data relating to the task. The underlying data structures used to hold the business data in a TaskData object is different for regular Web-service clients and .NET clients.

TaskData Structure for Regular Web-service clients

For regular, Java-based Web-service clients, the TaskData object is a Map object in which business data is structured as a set of key-value pairs. Keys must be string values. Values can be any data type.

In the following example, the TaskData object holds customer information in a set of five key-value pairs.

Key	Value
<i>firstName</i>	String Anna

Key	Value
<i>lastName</i>	String Pareo
<i>emailAddress</i>	String apareo@msx.com
<i>acctNum</i>	String J5468-6268-508
<i>TransactionCodes</i>	String [] 20060819A1430 20060330P0604 20060314A1128

TaskData Structure for .NET Clients

For .NET clients, TaskData is a Map object that is made up of an array of MapEntry data structures. A MapEntry data structure contains two fields, as shown below:

Field	Description
<i>key</i>	String A name given to the field.
<i>value</i>	AnyType The data that belongs to the field.

In the following example, the TaskData array contains five MapEntry data structures. Each entry consists of a field called "key," which represents the name of the field, and a field called "value," which contains the value of the field.

Element #	key field	value field
0	<i>firstName</i>	String Anna
1	<i>lastName</i>	String Pareo
2	<i>emailAddress</i>	String apareo@msx.com
3	<i>acctNum</i>	String J5468-6268-508
4	<i>TransactionCodes</i>	String[] 20060819A1430 20060330P0604 20060314A1128

TaskInfo

Contains standard information about a task. You specify certain fields in TaskInfo when you start a task using the [queueTask](#) operation. After a task is queued, you can modify fields in TaskInfo using the [updateTask](#) operation.

Some fields that appear in TaskInfo can only be changed by the Task Engine. These fields are marked "read only" in the description below. If you attempt to assign a new value to a read-only field, the new value is ignored.

A task is not required to maintain information for every field in TaskInfo. Most fields are optional as noted in the description below.

Element	Description
<i>acceptedByList</i>	String[] Optional. The IDs (on My webMethods Server) of the users, groups, and roles that have accepted this task. Setting this field accepts the task for the specified users, groups, and roles.
<i>assignedToList</i>	String[] Optional. The IDs of the principals (users, groups, and roles on My webMethods Server) to which this task is assigned. Setting this field assigns the task to the specified users, groups, and roles. The Task Engine uses this field to route the task to the appropriate users.
<i>attributes</i>	Map Optional. Contains data that is used by the process run-time. For internal use only.
<i>auditContext</i>	String Read-only. The AuditContext value from the pub.prt:ProcessData document. This value appears in TaskInfo only if the task was queued by a business process.
<i>collaborationProcessID</i>	String Read-only. The unique ID of the collaboration process flow created when a task is used for collaboration workflow.
<i>collaborationStepID</i>	String Read-only. The unique ID of the task step in the process flow when task is used for collaboration workflow.
<i>createdBy</i>	String Read-only. The user ID (on My webMethods Server) of the user that initially queued the task.

Element	Description
<i>createdDate</i>	DateTime Read-only. The date and time when the task was queued.
<i>currentUserAccepted</i>	Boolean Read-only. Returns true if the task is accepted by the current user.
<i>customTaskID</i>	<p>String Optional. An optional, application-defined identifier for the task. This ID is separate from the internal taskID that the Task Engine uses to identify tasks. The identifier in <i>customTaskID</i> is visible in the user interface and is also searchable using the searchTasks operation.</p> <p>Note: Although <i>customTaskID</i> is meant to uniquely identify a task, the Task Engine does not enforce uniqueness of the value in this field. The application is responsible for assigning unique identifiers to <i>customTaskID</i> if they are needed.</p>
<i>delegatedFrom</i>	String Read-only. If a task is delegated to the current user, this field contains the user ID of the user who delegated the task to the current user.
<i>delegatedFromList</i>	String[] Optional. List of user IDs who delegated this task.
<i>delegatedToList</i>	String[] Optional. List of user IDs to whom the task was delegated
<i>delegationMap</i>	<p>Map Optional. A Map containing task delegation information (<i>not available for .NET service</i>). The map key is the user ID of the user who delegates the task; the value for the key is the user ID of the users to whom the task is delegated.</p> <ul style="list-style-type: none"> ■ <i>key</i> String User ID of delegating user. ■ <i>value</i> String User ID of target user.
<i>description</i>	<p>String Optional. A descriptive comment or remark associated with the task. This description appears in various places in the My webMethods user interface, such as on the Details View tab in My Inbox and in the Task Management Results list.</p> <p>Maximum length is 255 characters.</p>

Element	Description
<i>errorCode</i>	String Optional. A code that identifies the error condition that caused the task to end. This field is usually present when the value in <i>status</i> is "error." However, the Task Engine does not require an application to report an error code, so this field might be null even if the task ends with an error.
<i>errorMessage</i>	String Optional. A message describing the error condition that caused the task to end. This field is usually present when the value in <i>status</i> is "error." However, the Task Engine does not require an application to report an error message, so this field might be null even if the task ends with an error.
<i>expireDate</i>	DateTime Optional. The date and time when the task expires. When <i>expireDate</i> is reached, the Task Engine switches the <i>status</i> value for the task to "expired." If an expire date is not specified, the task never expires.
<i>lastAcceptedBy</i>	String Read-only. The user ID (on My webMethods Server) of the last user to accept the task. The field is set to null if no user has accepted the task, or when <i>acceptedByList</i> is reset to null.
<i>lastAcceptedDate</i>	DateTime Read-only. The date and time the task was last accepted
<i>lastModifiedBy</i>	String Read-only. The user ID (on My webMethods Server) of the user that last updated the task. If a process within the Task Engine was the last entity to modify the task (for example, if the Task Engine marked the task "expired"), this field contains the name of the task rule associated with that process.
<i>lastModifiedDate</i>	DateTime Read-only. Date and time when the task was last updated.
<i>name</i>	String Optional. The name of the task. This name appears in various places in the My webMethods user interface, such as on the Details View tab in My Inbox and in the Task List Management Results list.

Element	Description
	A task does not require a name. If a name is assigned, it does not need to be unique. Maximum length is 255 characters.
<i>parentTaskID</i>	String Optional. The value of the parent task ID in the case when tasks are used for collaboration workflow. When queuing a new task, if a valid taskID is specified for "parentTaskID", the task to be queued is created as a child task of the parent task.
<i>priority</i>	String Optional. The priority of the task. Must contain one of the following values: <div style="background-color: #f0f0f0; padding: 2px;"><code>none low medium high critical</code></div> Values are case-sensitive.
<i>processInstanceID</i>	String Read-only. The ProcessInstanceID value from the <code>pub.prt:ProcessData</code> document. This value only appears in TaskInfo if the task was queued by a business process.
<i>processIteration</i>	Integer Read-only. The ProcessIteration value from the <code>pub.prt:ProcessData</code> document. This value only appears in TaskInfo if the task was queued by a business process.
<i>processModelID</i>	String Read-only. The ProcessModelID value from the <code>pub.prt:ProcessData</code> document. This value only appears in TaskInfo if the task was queued by a business process.
<i>processModelVersion</i>	String Read-only. The ProcessModelVersion value from the <code>pub.prt:ProcessData</code> document. This value only appears in TaskInfo if the task was queued by a business process.
<i>status</i>	String Optional. The state of the task. Must contain one of the following values: <ul style="list-style-type: none"> ■ <code>new</code>. Task is new and not yet started. Immediately after the task starts, the status of the task changes from <code>new</code> to <code>active</code>. ■ <code>scheduled</code>. Task is scheduled to start at the time specified in <code>taskScheduleDate</code>. When the task starts at the scheduled time, the status of the task changes from <code>scheduled</code> to <code>active</code>. ■ <code>active</code>. Task is active and is available in the user's inbox queue. Only active tasks can be modified by the user.

Element	Description
	<ul style="list-style-type: none"> <li data-bbox="578 323 1373 426">■ <code>suspended</code>. Task is suspended and is not available in the user's inbox queue. Suspended tasks continue to appear in the Task Management Results list. <li data-bbox="578 447 1373 583">■ <code>completed</code>. Task has been successfully completed and is no longer available in the user's inbox queue. Completed tasks continue to appear in the Task Management Results list. <li data-bbox="578 604 1373 707">■ <code>cancelled</code>. Task has been canceled and is no longer available in the user's inbox queue. Canceled tasks continue to appear in the Task Management Results list. <li data-bbox="578 728 1373 831">■ <code>expired</code>. Task has expired and is no longer available in the user's inbox queue. Expired tasks continue to appear in the Task Management Results list. <li data-bbox="578 852 1373 1016">■ <code>error</code>. Task has failed or an unrecoverable error occurred during the processing of the task. This is often due to an incorrect task rule. The <code>errorCode</code> and <code>errorMessage</code> fields generally contain additional information about the error. <p data-bbox="578 1037 1373 1108">If you attempt to set <code>status</code> to any value other than the ones listed above, your value is ignored.</p>
<i>stepID</i>	<p data-bbox="578 1142 1373 1245">String Read-only. The StepID value from the <code>pub.prt:ProcessData</code> document. This value only appears in <code>TaskInfo</code> if the task was queued by a business process.</p>
<i>stepIteration</i>	<p data-bbox="578 1287 1373 1390">Integer Read-only. The TryCount value from the <code>pub.prt:ProcessData</code> document. This value only appears in <code>TaskInfo</code> if the task was queued by a business process.</p>
<i>taskScheduleDate</i>	<p data-bbox="578 1432 1373 1503">Date Specifies the date and time the task should start. The status of the task changes based on the specified value.</p> <ul style="list-style-type: none"> <li data-bbox="578 1524 1373 1593">■ If you specify a date that is later than the current date, the task status will be <code>scheduled</code>. <li data-bbox="578 1614 1373 1646">■ If you do not specify a date, the task will be <code>active</code>. <p data-bbox="578 1667 1373 1766">When the task starts or queues at the scheduled time, the global rule for task schedule changes the status of a task from <code>scheduled</code> to <code>active</code>.</p>
<i>taskID</i>	<p data-bbox="578 1808 1373 1881">String Read only. A unique identifier assigned to the task by the Task Engine when the task is queued.</p>

Element	Description
<i>taskTypeID</i>	String Read only. Specifies the task's type. Each task type that is deployed on the Task Engine has a unique ID. This ID is assigned by the developer when he or she creates a task application using Software AG Designer.
<i>taskURL</i>	String Read only. The relative URL for the task on My webMethods Server. This URL can be used to open the task instance in a browser.
<i>taskVersionNumber</i>	<p>Integer Optional. This is the current version number of the task record that is automatically incremented for each task update action.</p> <p>This property can be used in the updateTask() service. If a value different than 0 is passed in when executing updateTask(), then the Task Engine verifies this value against the current value of the task instance and throw an exception "Task is Out of Date" if they mismatch. For example, when the client of the API is holding a task record, but the task record has been since updated and thus "taskVersionNumber" was incremented.</p>

Task

Data object returned as output from the [getTask](#) and [searchTasks](#) operations. This data structure contains the TaskInfo and TaskData objects associated with a task.

Element	Description
<i>taskData</i>	Map Optional. Contains the business data associated with a task. See " TaskData " on page 156 for a description of this object.
<i>taskInfo</i>	TaskInfo Contains standard information that the Task Engine maintains about a task. See " TaskInfo " on page 158 for a description of this object.

TaskSearchQuery

Used as input to the [searchTasks](#) operation. The elements defined here are also present in [TaskSearchQueryV2](#).

Element	Description
<i>doNotShowAcceptedByOthers</i>	<p>Boolean Optional. Exclude tasks that are accepted by a user other than the user ID used in the search. Default is false.</p>
<i>maxResults</i>	<p>Integer Optional. The maximum number of tasks to return in the result set. If <i>MaxResults</i> is not specified, all results in the result set are returned.</p> <p>Note: This element does not apply to tasks with an indexed search provider. In this case, the specified value is ignored.</p>
<i>showNonActiveTasks</i>	<p>Boolean Optional. This field can be used when executing queries to fetch tasks from a user's inbox. This parameter is meant to override the default behavior of the <i>searchUserTasks</i> parameter and allow the return of non-active tasks.</p> <p>Set to:</p> <ul style="list-style-type: none"> ■ <code>true</code> to return non-active tasks. ■ <code>false</code> (default) to return active tasks only. <p>For example, by specifying <i>searchUserTasks</i> =<code>true</code> and <i>showNonActiveTasks</i> =<code>true</code>, the query returns all non-active tasks from a user's inbox (that is, those tasks already completed by user).</p> <p>The <i>showNonActiveTasks</i> parameter has no impact on non-user task searches. To construct a non-user task search to return only active tasks, you must set query terms of <i>searchUserTasks</i> =<code>false</code> and <i>status</i> =<code>active</code>.</p>
<i>terms</i>	<p>TaskSearchQueryTerm[] Specifies the TaskSearchQueryTerm object that contains an array of TaskSearchQueryTerm objects that specify the query criteria for the search. See “TaskSearchQueryTerm” on page 165 for a description of the fields in this object.</p> <p>When a search is performed, the criteria specified by the TaskSearchQueryTerm objects in <i>terms</i> are combined using the logical "AND" operator. Only tasks that satisfy the terms specified by all</p>

Element	Description
	TaskSearchQueryTerm objects in the <i>terms</i> array are returned in the result set.

Usage Notes

The *showNonActiveTasks* parameter overrides the default behavior of the *userTasks* parameter in any web service where that parameter occurs, such as:

- [countTasksIndexed](#)
- [searchTasks](#)
- [searchTasksIndexed](#)

TaskSearchQueryTerm

Contains search criteria for the [searchTasks](#) and [searchTasksIndexed](#) operations.

You use this data structure to specify 1) the field or fields that you want the service to search, 2) the value to which you want the field's contents to be compared, and 3) the type of comparison that you want the service to make.

If you include multiple TaskSearchQueryTerms in the *terms* array in [TaskSearchQuery](#), all terms in the array are combined using the logical "AND" operator. Only tasks that satisfy all TaskSearchQueryTerm objects in the *terms* array are returned in the result set.

Element	Description
<i>fields</i>	<p>String[] A list of fields that are to be searched. There are two different methods for specifying the field name, depending on whether you are working with standard task fields or indexed task fields. For more information about indexed fields, see the <i>webMethods BPM Task Development Help</i>.</p> <p>Working with Standard Task Fields</p> <p>You can search the following standard task fields in TaskInfo using these names:</p> <pre> acceptedByList assignedToList auditContext collaborationProcessID collaborationStepID createdBy createdDate customTaskID description expireDate lastModifiedDate lastModifiedBy </pre>

Element	Description
	<pre>lastAcceptedBy name priority taskID taskTypeID parentTaskID processInstanceID processModelID processModelVersion status stepID stepIteration</pre>
	<p>To search standard fields in TaskData (the task business data), specify the field using a binding expression in this format:</p>
	<pre>#{currentTask.taskData.pathToDataElement }</pre>
	<p>For example, this binding expression:</p>
	<pre>#{currentTask.taskData.travelReservation.reservationNo}</pre>
	<p>searches the "reservationNo" field within the "travelReservation" document in TaskData.</p>
	<p>If you specify multiple fields in this parameter, the comparison evaluates to true if <i>any</i> of the specified fields match the query defined by <i>operator</i> and <i>value</i> (that is, the service performs a logical "OR" when asked to evaluate multiple fields).</p>
	<p>This API returns the values for the specified data fields only.</p>
	<h3>Working with Indexed Fields</h3>
	<p>To search tasks with indexed fields, you must pass in the database name of the indexed field along with the desired operator and value. The indexed field name is set in the Designer task editor. To determine the database index field name:</p>
	<ol style="list-style-type: none"> 1. Open the task that contains the indexed data field. 2. On the Business Data tab, select the field you want to work with. 3. Click Edit. 4. In the Edit Business Data dialog box, obtain the indexed field name from the Name field in the Database field settings area.
	<p>For example, to search the database field "ZipCode" for matches to the value "90210":</p>
	<pre>SearchTerm.fields="ZipCode" SearchTerm.operator="=" SearchTerm.value="90210"</pre>

This API returns the values for the specified indexed fields only. For more information about indexed fields, see the *webMethods BPM Task Development Help*.

operator

String The comparison that the service makes when evaluating the contents of *Value* against the contents of the specified field(s).

Operator must be one of the following:

- = Is equal to.
- <> Is not equal to.
- < Is less than. Valid only for numbers or date fields.
- > Is greater than. Valid only for numbers or date fields.
- <= Is less than or equal to. Valid only for numbers or date fields.
- >= Is greater than or equal to. Valid only for numbers or date fields.
- contains Compares *Value* to the individual elements in a String array. The comparison evaluates to true if any element in the list matches *Value*. Valid only with string array fields.

For example, if you have a string array with three elements, this operator evaluates to true if any one of the three elements contains a string that exactly matches the value in *Value*.

- in Matches if the single string value of the task field can be found inside a string list of values passed in the term. Such as: *TaskFieldValue* in (*value1*, *value2*, *value3*). Valid for all types.
- is empty Matches if the value of the field is either null or an empty string ("").
- is not empty Matches if the value of the field is not null and not an empty string ("").
- is null Matches if the current term field is null.
- is not null Matches if current term field is not null.
- like Matches the pattern string specified in *Value*. See *Value* for pattern string information. Valid only with string fields.
- not in Matches if the single string value of the task field cannot be found in a string list of values passed in the term. Such as: *TaskFieldValue* not in (*value1*, *value2*, *value3*). Valid for all types.
- not like Matches for all instances where the pattern string does not match the text specified in *Value*. See *Value* for information about pattern strings; valid only with string fields.

value

anyType The value to which the contents of the field specified in *Field* is compared.

Value can also specify a pattern string that can include the * wildcard character. For example:

- A *Value* of `abc` would return tasks whose field contents contain only the character sequence `abc`.
- A *Value* of `abc*` would return tasks whose field contents start with the character sequence `abc`.
- A *Value* of `*abc` would return tasks whose field contents end with the character sequence `abc`.
- A *Value* of `*abc*` would return tasks whose field contents include the character sequence `abc` anywhere within the field.

Note: *Value* is case-sensitive. That is, `abc` will not match `ABC` or `Abc`.

TaskSearchQueryV2

Use the `TaskSearchQueryV2` data type as input to [searchTasksIndexed](#).

The `TaskSearchQueryV2` data type includes the same fields as [TaskSearchQuery](#) plus the additional fields listed in the table below to support queries on indexed business data.

Field Name	Description
<i>fromIndex</i>	Integer Specifies the starting index of the page of search results. Zero represents the first index of the result set.
<i>toIndex</i>	Integer Indicates the end of the search results indexed on the page. The <i>toIndex</i> must be greater than the <i>fromIndex</i> and less than or equal to the total result count.
<i>sortBy</i>	String Optional. Specifies the comma-separated list of field names used to sort the search results. The first entry is the primary sort key, second entry is the secondary sort key, matching the results to the order of the sort key sequence. When specifying indexed field names, you must provide the database index field name. To determine the value of the database index field name, see " TaskSearchQueryTerm " on page 165.

Field Name	Description
<i>sortOrder</i>	<p>String Optional. Specifies the comma-separated list of values ascending (ASC) or descending (DESC) that match the <i>sortBy</i> list of fields. <i>sortOrder</i> indicates the sort order used to list the fields, ascending or descending order.</p> <p>Set <i>sortOrder</i> to:</p> <ul style="list-style-type: none"> ■ ASC to display the results in ascending order. ■ DESC to display the results in descending order. <p>The default <i>sortOrder</i> is ascending (ASC).</p>
<i>searchEngineType</i>	<p>String Optional. Specifies the search engine to use when searching for tasks. Valid values are:</p> <ul style="list-style-type: none"> ■ <code>db</code> - to search task data, stored in the indexed table of the My webMethods Server database. ■ <code>hpstra</code> - to search task data, indexed in an Elasticsearch store by the HPSTRA module in Task Engine. <p>You can also specify the name of any custom engine that you develop for task searches and register as an OSGi service in the My webMethods Server runtime. For more information and examples about adding custom task search engines, see the Software AG TECHcommunity website.</p>
<i>principalID</i>	<p>String Optional. The ID of the principal whose inbox is to be searched. If not specified, the default value is the Task Server User, as defined on the WmTaskClient package's home page.</p>
<i>checkPermissions</i>	<p>Boolean Optional. Specifies if the current user's permission are checked prior to the task search. Default is <code>true</code>.</p>
<i>invocationID</i>	<p>This parameter is for internal use only. Do not use.</p>

Index

A

- addTagsToTask
 - REST service 102
 - web service 134
- addTaskAttachment 16
 - REST service 105
 - web service 135
- addTaskComment
 - built-in service 17
 - REST service 100
 - web service 136
- attachment
 - deleting
 - with MWS restful service 126
- attachments
 - get
 - with MWS restful service 126
 - posting
 - with MWS restful service 128
 - put
 - with MWS restful service 127
- audit
 - get
 - with MWS restful service 121

B

- built-in service 16, 21, 27, 45
- business data
 - structure of for Web service 156
 - updating with built-in service 42

C

- comment
 - deleting
 - with MWS restful service 123
 - posting
 - with MWS restful service 125
 - put
 - with MWS restful service 124
- comments
 - get
 - with MWS restful service 124
- countTasksIndexed 18, 19

D

- data structures
 - Task 163
 - TaskData 156
 - TaskInfo 158
 - TaskSearchQuery 163
 - TaskSearchQueryTerm 165
- delete audit entries
 - with MWS REST service 121
- deleteTagsFromTask 21
 - REST service 101
 - web service 138
- deleteTask operation 138
- deleteTask service 21
- deleteTaskAttachment
 - built-in service 22
 - REST service 103
 - web service 139
- deleteTaskComment 22
 - REST service 98
 - web service 139
- document types
 - Guidance 47
 - Task 47
 - TaskAttachment
 - built-in service 48
 - web service 153
 - TaskAudit 49
 - web service 154
 - TaskComment 51
 - web service 156
 - TaskCompletionInfo 51
 - TaskFields 52
 - TaskInfo 53
 - TaskQueueInfo 58
 - TaskSearchQuery 60

E

- endpoint for task Web service 131

F

- formFlowTaskNotify
 - built-in service 23
 - MWS REST request 122
 - REST request 96
 - web service 140

G

- getTagsByTask

REST service 101
 getTask operation 141
 getTask service 24
 getTaskAttachments
 built-in service 26
 REST service 104
 web service 142
 getTaskAudit
 built-in service 26
 REST request 95
 web service 142
 getTaskComments
 built-in service 26
 RESTS service 98
 web service 143
 Guidance
 built-in service document type 47

L

listTagsByTask 27
 web service 143

M

mapping
 TaskData array 157
 user delegation 159
 MWS REST services
 _delete attachment from task service 126
 _delete audit entries service 121
 _delete comment service 123
 _delete task service 110
 _get task attachments service 126
 _get task audit service 121
 _get task comments service 124
 _get task search service 115
 _get task service 111
 _post task attachments service 128
 _post task comment service 125
 _post task search service 116
 _post task service 112
 _put task attachments service 127
 _put task comment service 124
 _put task information service 113

N

notifying a task workflow component
 with MWS REST service 122
 with REST service 96

notifying a task workflow component with built-in
 service 23
 notifying a task workflow component with
 webservice 140

O

operations, Web service 133

P

pub.opentask
 _delete service 73
 _get service 74
 _post service 77
 _put service 78
 MWS
 _delete attachment from task service 126
 _delete audit entries service 121
 _delete comment service 123
 _delete service 110
 _get task attachments service 126
 _get task audit service 121
 _get task comments service 124
 _get task instance service 111
 _post task attachments service 128
 _post task comment service 125
 _post task service 112
 _put task attachments service 127
 _put task comment service 124
 _put task information service 113
 pub.opentasksearch
 _get service 80, 81
 _post service 83, 89
 MWS
 _get task search service 115
 _post task search service 116
 pub.task.taskclient
 deleteTask 21
 pub.task
 Guidance 47
 Task 47
 TaskAttachment 48
 TaskAudit 49
 TaskCallbackService 50
 TaskComment 51
 TaskCompletionInfo 51
 TaskFields 52
 TaskInfo 53
 TaskQueueInfo 58

- TaskSearchQuery 60
- pub.task.taskclient
 - addTaskAttachment 16
 - addTaskComment 17
 - countTasksIndexed 18, 19
 - deleteTagsFromTask 21
 - deleteTaskAttachment 22
 - deleteTaskComment 22
 - formFlowTaskNotify 23
 - getTask 24
 - getTaskAttachments 26
 - getTaskAudit 26
 - getTaskComments 26
 - listTagsByTask 27
 - queueTask 28
 - rollbackTask 29
 - searchTaskFields 34, 36
 - searchTasks 30, 32
 - updateTask 42
 - updateTaskAttachment 44
 - updateTaskComment 44
 - updateTaskTags 45
- pub.task.TaskSearchQueryV2 66

Q

- queueTask operation 144
- queueTask service 28

R

- REST requests
 - formFlowTaskNotify 96
 - getTaskAudit 95
 - rollbackTask 96
- REST services
 - addTagsToTask 102
 - addTaskAttachment 105
 - addTaskComment 100
 - deleteTagsFromTask 101
 - deleteTaskAttachment 103
 - deleteTaskComment 98
 - getTagsByTask 101
 - getTaskAttachments 104
 - getTaskComments 98
 - updateTaskAttachment 105
 - updateTaskComment 99
 - updateTaskTags 103
- RESTful services API
 - described 71

- retrieving a task
 - with built-in service 24
 - with Web service operation 141
- rollbackTask
 - built-in service 29
 - REST request 96
 - web service 145

S

- searchTasks operation 147
- searchTasks service 30, 32
- searchTasksFields operation 145
- services, built-in
 - TaskSearchQuery document type 60
- services, built-in
 - _delete service 73
 - _get search service 80, 81
 - _get service 74
 - _post search service 83, 89
 - _post service 77
 - _put service 78
 - deleteTask service 21
 - formFlowTaskNotify service 23
 - getTask service 24
 - Guidance document type 47
 - queueTask service 28
 - searchTasks service 30, 32
 - Task document type 47
 - TaskAttachment document type 48
 - TaskAudit document type 49
 - TaskCallbackService specification 50
 - TaskComment document type 51
 - TaskCompletionInfo document type 51
 - TaskFields document type 52
 - TaskInfo document type 53
 - TaskQueueInfo document type 58
 - updateTask service 42
- services, MWS REST
 - _delete attachment from task service 126
 - _delete audit entries service 121
 - _delete comment service 123
 - _delete service 110
 - _get task attachments service 126
 - _get task audit service 121
 - _get task comments service 124
 - _get task search service 115
 - _get task service 111
 - _post task attachments service 128

- _post task comment service 125
- _post task search service 116
- _post task service 112
- _put task attachments service 127
- _put task comment service 124
- _put task information service 113
- formFlowTaskNotify request 122
- services, REST
 - formFlowTaskNotify request 96
- services, web
 - formFlowTaskNotify service 140
 - TaskAudit document type 154
 - TaskComment document type 156
- services,web
 - TaskAttachment document type 153
- specification, TaskCallbackService 50

T

- Task
 - built-in service document type 47
- Task data structure 163
- task Web service URL 131
- task workflow notify
 - built-in service 23
 - MWS REST request 122
 - REST request 96
 - web service 140
- TaskAttachment
 - built-in service document type 48
 - web service document type 153
- TaskAudit
 - built-in service document type 49
 - web service document type 154
- TaskCallbackService specification 50
- TaskComment
 - built-in service document type 51
 - web service document type 156
- TaskCompletionInfo
 - built-in service document type 51
- TaskData array, mapping 157
- TaskData data structure 156
 - for .NET Web service clients 157
- TaskFields
 - built-in service document type 52
- TaskInfo
 - built-in service document type 53
 - Web service data structure 158
- TaskQueueInfo document type 58

- tasks
 - deleting
 - with built-in restful service 73
 - with built-in service 21
 - with MWS restful service 110
 - with Web service operation 138
 - getting
 - with built-in restful search service 80, 81
 - with built-in service 24
 - with MWS restful search service 115
 - with Web service 141
 - listing
 - with built-in restful service 74
 - with MWS restful service 111
 - notifying
 - with built-in service 23
 - with MWS REST service 122
 - with REST service 96
 - with web service 140
 - posting
 - with built-in restful service 77
 - with MWS REST search service 116
 - with MWS restful service 112
 - with REST search service 83, 89
 - putting
 - with built-in restful service 78
 - with MWS restful service 113
 - queuing
 - with built-in service 28
 - with Web service 144
 - rollback
 - with MWS REST service 121
 - searching for
 - with built-in service 30, 32
 - with Web service 145, 147
 - updating
 - with built-in service 42
 - with Web service 149
- Web service data structures
 - Task 163
 - TaskData 156
 - TaskInfo 158
 - TaskSearchQuery 163
 - TaskSearchQueryTerm 165
- TaskSearchQuery
 - built-in service document type 60
 - Web service data structure 163
- TaskSearchQueryTerm
 - Web service data structure 165

TaskSearchQueryV2 66

U

updateTask operation 149

updateTask service 42

updateTaskAttachment

built-in service 44

REST service 105

web service 151

updateTaskComment

built-in service 44

REST service 99

web service 151

updateTaskTags 45

REST service 103

web service 152

URL for task Web service 131

user parameter, for Web service operations 132

W

Web service

data structures

Task 163

TaskData 156

TaskInfo 158

TaskSearchQuery 163

TaskSearchQueryTerm 165

URLs for 131

Web service API

overview of 131

summary of services 133

Web service operations

deleteTask 138

getTask 141

queueTask 144

searchTasks 145, 147

updateTask 149

user input parameter 132

using 132

web services

addTagsToTask 134

addTaskAttachment 135

addTaskComment 136

deleteTagsFromTask 138

deleteTaskAttachment 139

deleteTaskComment 139

formFlowTaskNotify 140

getTaskAttachments 142

getTaskAudit 142

getTaskComments 143

listTagsByTask 143

rollbackTask 145

TaskAttachment document 153

TaskAudit 154

TaskComment document 156

updateTaskAttachment 151

updateTaskComment 151

updateTaskTags 152

WSDL extraction, for task Web service 131

Symbols

.NET client

description of 131

TaskData structure 157

URL for task Web-service 131

_delete task service 73

_get task search service 80, 81

_get task service 74

_post task search service 83, 89

_post task service 77

_put task service 78