

# webMethods Task Engine API and Service Reference

Version 9.6

April 2014

This document applies to webMethods Task Engine Version 9.6 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 © 2006-2014 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://documentation.softwareag.com/legal/>.

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://documentation.softwareag.com/legal/> 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 and license terms, please refer to "License Texts, Copyright Notices and Disclaimers of Third Party Products". This document is part of the product documentation, located at <http://documentation.softwareag.com/legal/> and/or in the root installation directory of the licensed product(s).

# Table of Contents

<b>About this Guide.....</b>	<b>7</b>
Document Conventions.....	7
Documentation Installation.....	8
Online Information.....	8
<b>Using the Task Engine Built-in Services.....</b>	<b>11</b>
Task Engine Built-In Services Location.....	13
Summary of Available Services.....	13
pub.task.taskclient:addTaskAttachment.....	15
pub.task.taskclient:addTaskComment.....	16
pub.task.taskclient:countTasksIndexed.....	16
pub.task.taskclient:deleteTask.....	18
pub.task.taskclient:deleteTaskAttachment.....	19
pub.task.taskclient:deleteTaskComment.....	19
pub.task.taskclient:formFlowTaskNotify.....	20
pub.task.taskclient:getTask.....	21
pub.task.taskclient:getTaskAttachments.....	23
pub.task.taskclient:getTaskAudit.....	23
pub.task.taskclient:getTaskComments.....	23
pub.task.taskclient:queueTask.....	24
pub.task.taskclient:rollbackTask.....	26
pub.task.taskclient:searchTasks.....	26
pub.task.taskclient:searchTasksFields.....	28
pub.task.taskclient:searchTasksIndexed.....	30
pub.task.taskclient:updateTask.....	32
pub.task.taskclient:updateTaskAttachment.....	34
pub.task.taskclient:updateTaskComment.....	35
Summary of Data Structures.....	35
pub.task:Task.....	37
pub.task:TaskAttachment.....	37
pub.task:TaskAudit.....	38
pub.task:TaskCallbackService.....	39
pub.task:TaskComment.....	40
pub.task:TaskCompletionInfo.....	41
pub.task:TaskFields.....	42
pub.task:TaskInfo.....	42
pub.task:TaskQueueInfo.....	46
pub.task:TaskSearchQuery.....	47
pub.task:TaskSearchQueryTerm.....	49
pub.task.TaskSearchQueryV2.....	54

<b>Using Task Engine RESTful Web Services.....</b>	<b>57</b>
Overview.....	58
Summary of REST Resources and Requests.....	58
DELETE a Task Instance.....	60
GET Task Instance Information.....	61
POST a New Task Instance.....	63
PUT Information in a Task Instance.....	65
GET Tasks with a Simple Search Request.....	66
POST a Complex Task Search Request.....	68
GET Task Audit Information.....	74
DELETE Audit Entries to Rollback Task.....	74
GET or POST a Form Flow Notification.....	75
DELETE Task Comment.....	77
GET Task Comments.....	77
PUT Updates into a Task Comment.....	78
POST Task Comment.....	79
DELETE Task Attachment.....	80
GET Task Attachments.....	81
PUT Task Attachment.....	81
POST Task Attachment.....	82
<b>Using the Task Engine Web Service API.....</b>	<b>85</b>
Overview.....	87
Task Service URLs.....	87
Extracting the WSDL for the Task Web Service.....	87
Using the Task Web Service Operations.....	88
Summary of Available Operations.....	89
addTaskAttachment.....	90
addTaskComment.....	91
countTasksIndexed.....	92
deleteTask.....	93
deleteTaskAttachment.....	94
deleteTaskComment.....	94
formFlowTaskNotify.....	95
getTask.....	96
getTaskAttachments.....	97
getTaskAudit.....	97
getTaskComments.....	98
queueTask.....	98
rollbackTask.....	99
searchTaskFields.....	100
searchTasks.....	101
searchTasksIndexed.....	102
updateTask.....	104
updateTaskAttachment.....	106

---

updateTaskComment.....	106
Summary of Data Structures.....	107
TaskAttachment.....	108
TaskAudit.....	109
TaskComment.....	110
TaskData.....	111
TaskInfo.....	112
Task.....	117
TaskSearchQuery.....	117
TaskSearchQueryTerm.....	119
TaskSearchQueryV2.....	123
<b>Index.....</b>	<b>125</b>



---

## About this Guide

---

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

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

## Document Conventions

---

Convention	Description
<b>Bold</b>	Identifies elements on a screen.
Narrowfont	Identifies storage locations for services on webMethods Integration Server, using the convention <i>folder.subfolder:service</i> .
UPPERCASE	Identifies keyboard keys. Keys you must press simultaneously are joined with a plus sign (+).
<i>Italic</i>	Identifies variables for which you must supply values specific to your own situation or environment. Identifies new terms the first time they occur in the text.
Monospace font	Identifies text you must type or messages displayed by the system.
{ }	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.
[ ]	Indicates one or more options. Type only the information inside the square brackets. Do not type the [ ] symbols.

---

Convention	Description
...	Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...).

---

## Documentation Installation

---

You can download the product documentation using the Software AG Installer. The documentation is downloaded to a central directory named `_documentation` in the main installation directory (SoftwareAG by default).

## Online Information

---

You can find additional information about Software AG products at the locations listed below.

If you want to...	Go to...
Access the latest version of product documentation.	<b>Software AG Documentation website</b> <a href="http://documentation.softwareag.com">http://documentation.softwareag.com</a>
Find information about product releases and tools that you can use to resolve problems.  See the <a href="#">Knowledge Center</a> to: <ul style="list-style-type: none"> <li>■ Read technical articles and papers.</li> <li>■ Download fixes and service packs (9.0 SP1 and earlier).</li> <li>■ Learn about critical alerts.</li> </ul> See the <a href="#">Products area</a> to: <ul style="list-style-type: none"> <li>■ Download products.</li> <li>■ Download certified samples.</li> <li>■ Get information about product availability.</li> <li>■ Access older versions of product documentation.</li> </ul>	<b>Empower Product Support website</b> <a href="https://empower.softwareag.com">https://empower.softwareag.com</a>

---

If you want to...	Go to...
<ul style="list-style-type: none"> <li>■ Submit feature/enhancement requests.</li> </ul>	
<ul style="list-style-type: none"> <li>■ Access additional articles, demos, and tutorials.</li> <li>■ Obtain technical information, useful resources, and online discussion forums, moderated by Software AG professionals, to help you do more with Software AG technology.</li> <li>■ Use the online discussion forums to exchange best practices and chat with other experts.</li> <li>■ Expand your knowledge about product documentation, code samples, articles, online seminars, and tutorials.</li> <li>■ Link to external websites that discuss open standards and many web technology topics.</li> <li>■ See how other customers are streamlining their operations with technology from Software AG.</li> </ul>	<p data-bbox="873 405 1325 468"><b>Software AG Developer Community for webMethods</b></p> <p data-bbox="873 493 1263 556"><a href="http://communities.softwareag.com/">http://communities.softwareag.com/</a></p>

---



---

# 1 Using the Task Engine Built-in Services

---

■ Task Engine Built-In Services Location .....	13
■ Summary of Available Services .....	13
■ pub.task.taskclient:addTaskAttachment .....	15
■ pub.task.taskclient:addTaskComment .....	16
■ pub.task.taskclient:countTasksIndexed .....	16
■ pub.task.taskclient:deleteTask .....	18
■ pub.task.taskclient:deleteTaskAttachment .....	19
■ pub.task.taskclient:deleteTaskComment .....	19
■ pub.task.taskclient:formFlowTaskNotify .....	20
■ pub.task.taskclient:getTask .....	21
■ pub.task.taskclient:getTaskAttachments .....	23
■ pub.task.taskclient:getTaskAudit .....	23
■ pub.task.taskclient:getTaskComments .....	23
■ pub.task.taskclient:queueTask .....	24
■ pub.task.taskclient:rollbackTask .....	26
■ pub.task.taskclient:searchTasks .....	26
■ pub.task.taskclient:searchTasksFields .....	28
■ pub.task.taskclient:searchTasksIndexed .....	30
■ pub.task.taskclient:updateTask .....	32
■ pub.task.taskclient:updateTaskAttachment .....	34
■ pub.task.taskclient:updateTaskComment .....	35
■ Summary of Data Structures .....	35
■ pub.task:Task .....	37
■ pub.task:TaskAttachment .....	37
■ pub.task:TaskAudit .....	38
■ pub.task:TaskCallbackService .....	39

---

■ pub.task:TaskComment .....	40
■ pub.task:TaskCompletionInfo .....	41
■ pub.task:TaskFields .....	42
■ pub.task:TaskInfo .....	42
■ pub.task:TaskQueueInfo .....	46
■ pub.task:TaskSearchQuery .....	47
■ pub.task:TaskSearchQueryTerm .....	49
■ pub.task.TaskSearchQueryV2 .....	54

---

## Task Engine Built-In Services Location

---

The built-in services in this chapter are installed on the 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:

*IntegrationServerName* \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 the Integration Server is connected. folder:

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

## Summary of Available Services

---

The following elements are available in this folder:

Element	Package and Description
<a href="#">pub.task.taskclient:addTaskAttachment</a>	WmTaskClient. Adds an attachment to a task.
<a href="#">pub.task.taskclient:addTaskComment</a>	WmTaskClient. Adds a comment to a task.
<a href="#">pub.task.taskclient:countTasksIndexed</a>	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.
<a href="#">pub.task.taskclient:deleteTask</a>	WmTaskClient. Service that deletes an instance of a task on the Task Engine to which the Integration Server is connected.
<a href="#">pub.task.taskclient:deleteTaskAttachment</a>	WmTaskClient. Deletes the specified task attachment from a task.

Element	Package and Description
<a href="#">pub.task.taskclient:deleteTaskComment</a>	WmTaskClient. Deletes the specified task comment from a task.
<a href="#">pub.task.taskclient:formFlowTaskNotify</a>	WmTaskClient. Service that notifies a waiting step in a task workflow that the workflow is completed.
<a href="#">pub.task.taskclient:getTask</a>	WmTaskClient. Service that returns the TaskInfo object, and optionally the TaskData object, for a specified task.
<a href="#">pub.task.taskclient:getTaskAttachments</a>	WmTaskClient. Returns all the attachments in a task.
<a href="#">pub.task.taskclient:getTaskAudit</a>	WmTaskClient. Returns the audit log of all of the operations performed on a task.
<a href="#">pub.task.taskclient:getTaskComments</a>	WmTaskClient. Returns all the comments in a task.
<a href="#">pub.task.taskclient:queueTask</a>	WmTaskClient. Service that adds a task instance to the Task Engine's queue.
<a href="#">pub.task.taskclient:rollbackTask</a>	WmTaskClient. Enables you to roll back the task to any available audit point in the task's audit history.
<a href="#">pub.task.taskclient:searchTasks</a>	WmTaskClient. Service that is used to retrieve tasks that match specified search criteria. This service applies only to tasks that use standard business data.
<a href="#">pub.task.taskclient:searchTasksFields</a>	WmTaskClient. Service that returns a collection of named fields requested in each service that matches specified search criteria.
<a href="#">pub.task.taskclient:searchTasksIndexed</a>	WmTaskClient. Service that returns a subset of tasks that match the specified search criteria on the Task

Element	Package and Description
	Engine connected to the Integration Server.
<a href="#">pub.task.taskclient:updateTask</a>	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. By default, this service enables you to update only tasks in Active, Error, and Suspended state; for information about updating tasks with other statuses, see <i>Usage Notes</i> .
<a href="#">pub.task.taskclient:updateTaskAttachment</a>	WmTaskClient. Updates the specified task attachment.
<a href="#">pub.task.taskclient:updateTaskComment</a>	WmTaskClient. Updates the specified task comment.

## pub.task.taskclient:addTaskAttachment

WmTaskClient. Adds the specified attachment to a task. This service applies only to standard task attachments, and is not intended for use with Content Service Platform attachments.

### Input Parameters

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

### Output Parameters

<i>commentID</i>	<b>String</b> 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

---

<i>businessData</i>	<p><b>Boolean</b> 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> <p>Set to:</p> <ul style="list-style-type: none"><li>■ <code>true</code> to include business data in the search query processing. An error occurs if <i>businessData</i> is set to <code>true</code> and the task type does not contain any indexed fields.</li><li>■ <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.</li></ul>
<i>user</i>	<p><b>String</b> 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.</p>
<i>TaskSearchQuery</i>	<p><b>TaskSearchQueryV2</b> 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 <a href="#">pub.task.TaskSearchQueryV2</a>.</p>
<i>searchUserTasks</i>	<p><b>Boolean</b> 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"><li>■ <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.</li><li>■ <code>false</code> (default) to search all tasks to which <i>user</i> has access.</li></ul> <p>By default, only active tasks are returned. This can be overridden with the <i>showNonActiveTasks</i> parameter in <a href="#">pub.task:TaskSearchQuery</a>.</p>
<i>retryOnFailure Count</i>	<p><b>String</b> 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><b>String</b> 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</p>

---

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.

### See Also

[pub.task.taskclient:searchTasksIndexed](#)

[pub.task.TaskSearchQueryV2](#)

---

## 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.

### See Also

[pub.task.taskclient:searchTasks](#)

---

## pub.task.taskclient:deleteTaskAttachment

WmTaskClient. Deletes the specified task attachment. This service applies only to standard task attachments, and is not intended for use with Content Service Platform attachments.

### Input Parameters

---

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

---

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

<i>localOnly</i>	<b>String</b> Optional. Set to: <ul style="list-style-type: none"> <li>■ <code>true</code> if the service is not being used in a cluster environment.</li> </ul>
------------------	--

- 
- `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>	<b>String</b> The ID that identifies the task that you want to retrieve.
<i>includeTaskData</i>	<p><b>String</b> 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"> <li>■ <code>true</code> to retrieve TaskData in addition to TaskInfo.</li> <li>■ <code>false</code> (default) to get TaskInfo only.</li> </ul>
<i>retryOnFailureCount</i>	<b>String</b> 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>	<b>String</b> 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>	<b>Document</b> TaskInfo document containing standard information about the task. See <a href="#">pub.task:TaskInfo on page 42</a> for a description of the fields in this document.
<i>TaskData</i>	<p><b>Document</b> 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.

## See Also

[pub.task:TaskInfo](#)

---

[pub.task.taskclient:searchTasks](#)

---

## pub.task.taskclient:getTaskAttachments

WmTaskClient. Returns all the attachments in a task. This service applies only to standard task attachments, and is not intended for use with Content Service Platform attachments.

### 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:queueTask

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

---

## 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 the PDF publication *webMethods Task Engine User's Guide*.

*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 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.

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

*TaskQueueInfo*        **Document** Optional. A TaskQueueInfo document containing basic information about the task (e.g., name, priority, list of users to which the task is assigned). See [pub.task:TaskQueueInfo](#) for a description of the fields in this document.

---

<i>TaskData</i>	<b>Document</b> Optional. The business data, if any, associated with this task. TaskData does not have a specified structure. Its content varies according to the task.
<i>CallbackServiceName</i>	<p><b>String</b> 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> <p>The signature of the callback service must match the specification described in <a href="#">pub.task.TaskCallbackService on page 39</a>.</p>
<i>CallbackData</i>	<b>Document</b> Optional. Any data that you want the task to pass back to the callback service.
<i>ruleSet</i>	<b>String</b> 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>	<b>String</b> 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>	<b>String</b> 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>taskID</i>	<b>String</b> 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.

---

## 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

---

*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.

### 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

---

*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.

---

<i>TaskSearchQuery</i>	<p><b>Document</b> Optional. TaskSearchQuery document, which specifies the search criteria. See <a href="#">pub.task:TaskSearchQuery on page 47</a> 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 <a href="#">pub.task:TaskSearchQuery on page 47</a>.</p>
<i>searchUserTasks</i>	<p><b>String</b> Optional. Specifies whether the service searches all tasks or just the user's inbox. Set to:</p> <ul style="list-style-type: none"> <li>■ <code>true</code> to search only the inbox for <i>user</i>.</li> <li>■ <code>false</code> (default) to search all tasks to which <i>user</i> has access.</li> </ul> <p>By default, only active tasks are returned. This can be overridden with the <i>showNonActiveTasks</i> parameter in <a href="#">pub.task:TaskSearchQuery</a>.</p>
<i>retryOnFailureCount</i>	<p><b>String</b> 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><b>String</b> 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><b>Document List</b> Result set containing the tasks 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"> <li>■ <i>TaskID</i> <b>String</b> The identifier assigned to the task.</li> <li>■ <i>TaskInfo</i> <b>Document</b> A TaskInfo document containing standard information about the task. See <a href="#">pub.task:TaskInfo</a> for a description of the fields in this document.</li> <li>■ <i>TaskData</i> <b>Document</b> 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 <i>includeTaskData</i> is set to <code>true</code>.</li> </ul>
-------------	---

---

## 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 30](#).

## See Also

[pub.task:TaskSearchQuery](#)

[pub.task:TaskSearchQueryTerm](#)

[pub.task.taskclient:searchTasksFields](#)

---

## 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

*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.

---

<i>user</i>	<p><b>String</b> 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> <div style="background-color: #f0f0f0; padding: 5px; border: 1px solid #ccc;"> <p><b>Note:</b> Only tasks to which <i>user</i> has access are searched.</p> </div>
<i>TaskSearchQuery</i>	<p><b>Document</b> Optional. TaskSearchQuery document, which specifies the search criteria. See <a href="#">pub.task:TaskSearchQuery on page 47</a> 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 <a href="#">pub.task:TaskSearchQuery on page 47</a>.</p>
<i>searchUserTasks</i>	<p><b>String</b> 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"> <li>■ <code>true</code> to search only the inbox for <i>user</i>.</li> <li>■ <code>false</code> (default) to search all tasks to which <i>user</i> has access.</li> </ul> <p>By default, only active tasks are returned. This can be overridden with the <i>showNonActiveTasks</i> parameter in <a href="#">pub.task:TaskSearchQuery</a>.</p>
<i>fields</i>	<p><b>String</b> 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 <a href="#">pub.task:TaskSearchQueryTerm on page 49</a>. For example, suppose you want to return the following fields as result of the query:</p> <ul style="list-style-type: none"> <li>■ <code>taskID</code></li> <li>■ The business data field <i>orderID</i> from task data documents.</li> </ul> <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>retryOnFailure Count</i>	<p><b>String</b> Optional. Specifies the number of times to attempt to invoke the service. The default is 0 times. When specified,</p>

---

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 42.

## 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.

## See Also

[pub.task:TaskSearchQuery](#)

[pub.task:TaskSearchQueryTerm](#)

[pub.task.taskclient:searchTasks](#)

---

## 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 26.

---

## 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 contains no 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.

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

*user*

**String** Optional. Specifies the user ID of the Integration 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 [pub.task.TaskSearchQueryV2](#) document type that provides the search criteria. If *TaskSearchQueryV2* is null, an error occurs.

*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** Contains *TaskData* documents 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** 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 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:searchTasks](#)

[pub.task:TaskSearchQueryTerm](#)

[pub.task.TaskSearchQueryV2](#)

**See Also**

---

## 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. By default, this service enables you to update only tasks in Active,

---

Error, and Suspended state; for information about updating tasks with other statuses, see *Usage Notes*, below.

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>	<b>String</b> 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 the PDF publication <i>webMethods Task Engine User's Guide</i> .
<i>taskID</i>	<b>String</b> The ID that identifies the task you want to modify.
<i>TaskInfo</i>	<b>Document</b> TaskInfo object that specifies the changes you want to make to the standard information maintained for the task. See <a href="#">pub.task:TaskInfo on page 42</a> for a description of the fields in this document.
<i>TaskData</i>	<b>Document</b> 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>	<b>String</b> 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>	<b>String</b> 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.

By default, this operation enables you to update only tasks in Active, Error, and Suspended state. Tasks with other statuses (such as Completed, Cancelled, or Expired) cannot be updated.

However, a Task Engine optional setting, `#Update.completed.task=true`, is available to apply update actions to a task regardless of its current status. When this option is not present or set to `= false`, task updates apply only to tasks in Active, Error, or Suspended status.

For information about setting this option, see “Specifying Task Update Behavior” and “Applying Task Engine Optional Settings” in Chapter 6 of the PDF publication, *webMethods Task Engine User's Guide*.

---

## pub.task.taskclient:updateTaskAttachment

WmTaskClient. Updates the specified task attachment. This service applies only to standard task attachments, and is not intended for use with Content Service Platform attachments.

### Input Parameters

---

<i>attachment</i>	<b>TaskAttachment</b> The <a href="#">pub.task:TaskAttachment</a> object representing the updated attachment to be used to update the existing attachment.
<i>attachmentID</i>	<b>String</b> The unique identifier of the attachment to be updated.
<i>encoding</i>	<b>String</b> Specify the encoding type for the file. Valid values include: <ul style="list-style-type: none"><li>■ binary</li><li>■ text</li></ul>
<i>taskID</i>	<b>String</b> The unique identifier that the Task Engine assigns to the task.
<i>user</i>	<b>String</b> 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>	<b>TaskAttachment[]</b> An array of <a href="#">pub.task:TaskAttachment</a> objects representing the attachments to be added to the comment being updated.
<i>commentID</i>	<b>String</b> The unique identifier of the comment to be deleted.
<i>deleteAttachments</i>	<b>String[]</b> An array of IDs of the attachments to be deleted from the comment
<i>taskID</i>	<b>String</b> The unique identifier that the Task Engine assigns to the task.
<i>text</i>	<b>String</b> Optional. The updated text for this comment.
<i>user</i>	<b>String</b> Optional. The user ID of the My webMethods Server user on whose behalf this operation executes.

### Output Parameters

None.

---

## Summary of Data Structures

The following elements are available in this folder:

Element	Package and Description
<a href="#">pub.task:Task</a>	WmTaskClient. A document type that identifies the characteristics of a particular task.
<a href="#">pub.task:TaskAttachment</a>	WmTaskClient. Document type representing an attachment.
<a href="#">pub.task:TaskAudit</a>	WmTaskClient. Document type containing information returned as output from <a href="#">GetTaskAudit</a> .
<a href="#">pub.task:TaskCallbackService</a>	WmTaskClient. Specification that describes the inputs and outputs required for a callback service.
<a href="#">pub.task:TaskComment</a>	WmTaskClient. Document type representing a comment in a task.
<a href="#">pub.task:TaskCompletionInfo</a>	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.
<a href="#">pub.task:TaskFields</a>	WmTaskClient. Document type that contains information about task fields returned by <a href="#">pub.task.taskclient:searchTasksFields</a> .
<a href="#">pub.task:TaskInfo</a>	WmTaskClient. Document type that contains standard information about a task. You specify certain fields in <a href="#">TaskInfo</a> when you queue a task using <a href="#">pub.task.taskclient:queueTask</a> .
<a href="#">pub.task:TaskQueueInfo</a>	WmTaskClient. Document type used by <a href="#">pub.task.taskclient:queueTask</a> to set basic values in <a href="#">TaskInfo</a> .
<a href="#">pub.task:TaskSearchQuery</a>	WmTaskClient. Document type used as input to the

Element	Package and Description
	<a href="#">pub.task.taskclient:searchTasks</a> service. This service applies only to tasks that use standard business data.
<a href="#">pub.task:TaskSearchQueryTerm</a>	WmTaskClient. Document type that you use to specify search criteria for the <a href="#">pub.task.taskclient:searchTasks</a> service.
<a href="#">pub.task.TaskSearchQueryV2</a>	WmTaskClient. Indicates the document type that you use to specify the search criteria for the <a href="#">pub.task.taskclient:searchTasksIndexed</a> service.

## pub.task:Task

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

### Parameters

<i>TaskID</i>	<b>String</b> The value of the identifier that the Task Engine assigns to the task.
<i>TaskInfo</i>	<b>Document</b> TaskInfo document containing standard information about the task. See <a href="#">pub.task:TaskInfo on page 42</a> for a description of the fields in this document.
<i>TaskData</i>	<b>Document</b> 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>	<b>String</b> A Base64 encoded string representation of the data for the attachment.
-------------------	--

---

<i>contentType</i>	<b>String[ ]</b> Content-type of the data in the attachment.
<i>downloadLink</i>	<b>String</b> A link from My webMethods Server for downloading the attachment.
<i>fileName</i>	<b>String</b> Name of the file in the attachment.
<i>iconURL</i>	<b>String</b> URL of the icon to be used to represent the attachment in the task user interface.
<i>id</i>	<b>String</b> Read only. Unique ID of an existing attachment.
<i>lastModifiedDate</i>	<b>Date</b> The date and time of last modification of the attachment.
<i>name</i>	<b>String</b> Name of the attachment.
<i>contentLength</i>	<b>Long</b> Length of the data in the attachment.
<i>deletable</i>	<b>Boolean</b> Set to: <ul style="list-style-type: none"> <li>■ <code>true</code> if the attachment can be deleted by a user with appropriate permissions. Default.</li> <li>■ <code>false</code> if the attachment cannot be deleted by a user.</li> </ul>
<i>updatable</i>	<b>Boolean</b> Set to: <ul style="list-style-type: none"> <li>■ <code>true</code> if the attachment can be updated by a user with appropriate permissions. Default.</li> <li>■ <code>false</code> if the attachment cannot be updated by a user.</li> </ul>

---

## pub.task:TaskAudit

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

### Parameters

---

<i>createdDate</i>	<b>Date</b> The date and time the operation was applied to the audited task.
<i>difference</i>	<b>String[ ]</b> An array of strings representing the changes made as part of an audit log entry.

---

<i>sourceURI</i>	<b>String</b> URI of the source of this change.
<i>TaskURI</i>	<b>String</b> URI of the audited task.
<i>id</i>	<b>Integer</b> Unique ID of this audit log entry.
<i>operation</i>	<p><b>Integer</b> 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 <a href="#">Software AG Documentation web site</a> under My webMethods Server in the corresponding webMethods Product Suite release number.</p>
<i>taskVersionNumber</i>	<b>Integer</b> 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>	<b>Document</b> 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>	<b>Document</b> TaskCompletionInfo document, which contains information about the completion of the task See <a href="#">pub.task:TaskCompletionInfo on page 41</a> for a description of the fields in this document.
<i>CallbackData</i>	<b>Document</b> 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 <a href="#">pub.task.taskclient:queueTask on page 24</a> 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>	<b>TaskAttachment[]</b> An array of attachments to be enclosed inside the comment.
<i>id</i>	<b>String</b> Unique ID of the comment.
<i>displayText</i>	<b>String</b> 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>	<b>String</b> Text associated with the comment.
<i>timeStamp</i>	<b>Date</b> The date and time of last modification of the comment.

---

*userDisplayName*     **String** Displayable representation of the name of user associated with the comment.

*userID*                **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

---

*taskID*                **String** The identifier of the task that ended.

*status*                **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 still appear in the Task Management Results list.
- `cancelled`. Task has been cancelled and is no longer available in the user's inbox queue. Cancelled tasks still appear in the Task Management Results list.
- `expired`. Task has expired and is no longer available in the user's inbox queue. Expired tasks still 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. If no user accepted the task, this field is 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 24.

### 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

---

*acceptedByList*              **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.

*assignedToList*              **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>	<b>String</b> Read-only. Value of AuditContext from <code>pub.prt.ProcessData</code> document. This value appears only when a task is queued by business process
<i>collaborationProcessID</i>	<b>String</b> Read-only. The unique ID of the collaboration process flow created when a task is used for collaboration workflow.
<i>collaborationStepID</i>	<b>String</b> Read-only. The unique ID of the task step in the process flow when task is used for collaboration workflow.
<i>createdDate</i>	<b>java.util.Date</b> Read-only. Date and time when the task was queued.
<i>createdBy</i>	<b>String</b> Read-only. The user ID (on My webMethods Server) of the user that initially queued the task.
<i>customTaskID</i>	<p><b>String</b> 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 <a href="#">pub.task.taskclient:searchTasks</a> service.</p> <p><b>Note:</b> 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>delegatedFromList</i>	<b>String</b> Optional. List of user IDs who delegated this task.
<i>delegatedToList</i>	<b>String</b> Optional. List of user IDs to whom the task was delegated
<i>description</i>	<p><b>String</b> 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 <b>Details View</b> tab in My Inbox and in the Task Management Results list.</p> <p>Maximum length is 255 characters.</p>
<i>errorCode</i>	<b>String</b> 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>	<b>String</b> 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>	<b>java.util.Date</b> 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>	<b>String</b> Read-only. The user ID (on My webMethods Server) of the last user to accept the task. If no user has accepted the task, this field is null.
<i>lastAcceptedDate</i>	<b>java.util.Date</b> Read-only. Date and time when the task was last accepted.
<i>lastModifiedBy</i>	<b>String</b> 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.
<i>lastModifiedDate</i>	<b>java.util.Date</b> Read-only. Date and time when the task was last updated.
<i>name</i>	<b>String</b> Optional. The name of the task. This name appears in various places in the My webMethods user interface, such as on the <b>Details View</b> 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.
<i>parentTaskID</i>	<b>String</b> . 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><b>String</b> 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>	<p><b>String</b> 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.</p>
<i>processIteration</i>	<p><b>String</b> 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.</p>
<i>processModelID</i>	<p><b>String</b> 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.</p>
<i>processModelVersion</i>	<p><b>String</b> 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.</p>
<i>status</i>	<p><b>String</b> Optional. The state of the task. Must contain one of the following values:</p> <ul style="list-style-type: none"> <li>■ <code>active</code>. Task is active and is available in the user's inbox queue. Only active tasks can be modified by the user.</li> <li>■ <code>suspended</code>. Task is suspended and is not available in the user's inbox queue. Suspended tasks still appear in the Task Management Results list.</li> <li>■ <code>completed</code>. Task has been successfully completed and is no longer available in the user's inbox queue. Completed tasks still appear in the Task Management Results list.</li> <li>■ <code>cancelled</code>. Task has been cancelled and is no longer available in the user's inbox queue. Cancelled tasks still appear in the Task Management Results list.</li> <li>■ <code>expired</code>. Task has expired and is no longer available in the user's inbox queue. Expired tasks still appear in the Task Management Results list.</li> <li>■ <code>error</code>. Task has failed or an unrecoverable error occurred during the processing of the task. This is often due to an</li> </ul>

---

incorrect task rule. The *errorCode* and *errorMessage* fields generally contain additional information about the error.

<i>stepID</i>	<b>String</b> Read-only. The StepID 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>stepIteration</i>	<b>String</b> Read-only. The TryCount 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>taskTypeID</i>	<b>String</b> Read-only. The task type identifier (GUID) for this task instance.
<i>taskUrl</i>	<b>String</b> 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>	<b>Integer</b> 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 <a href="#">pub.task.taskclient:updateTask</a> . If a value different than 0 is passed when executing <a href="#">pub.task.taskclient:updateTask</a> , 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 <b>taskVersionNumber</b> 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

---

<i>name</i>	<b>String</b> Optional. The name of the task. This name appears in various places in the My webMethods user interface, such as on the <b>Details View</b> 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*.

---

## pub.task:TaskSearchQuery

WmTaskClient. Document type used as input to the [pub.task.taskclient:searchTasks](#) and [pub.task.taskclient:searchTasksFields](#) 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.

This parameter is ignored when the *searchUserTasks* parameter in [pub.task.taskclient:searchTasks](#) is set to `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`.

### *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](#)

### See Also

[pub.task:TaskSearchQueryTerm](#)

---

## 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* 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`

*Continued on next page*

- `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}
```

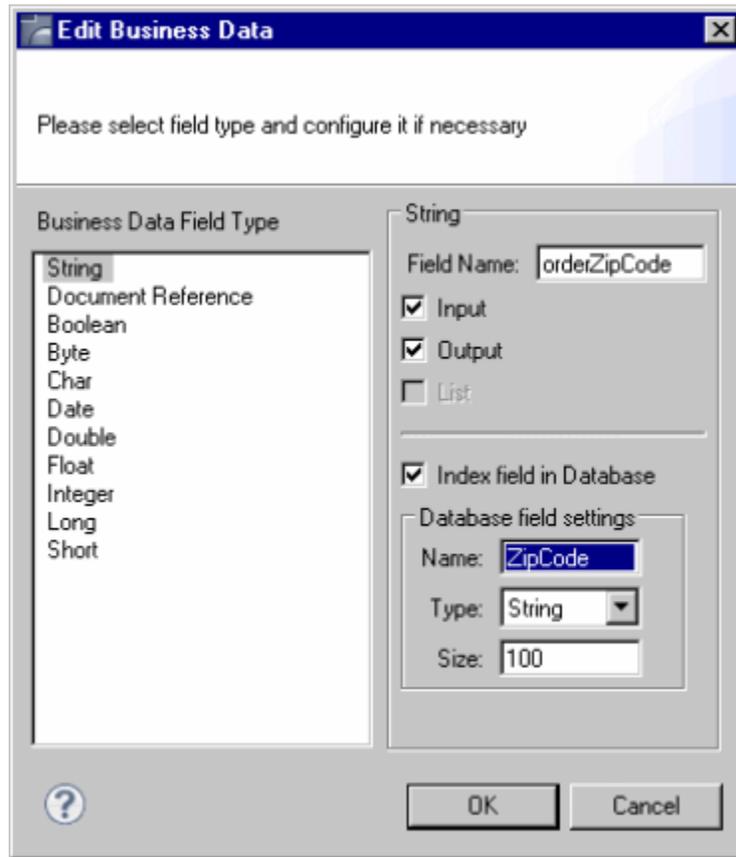
would search the "reservationNo" field within the "travelReservation" document in TaskData.

### **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.

*Continued on the next page*



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 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. When the can specify a valid regular expression pattern that will be used to match to a field value being tested. For example, it can be used to implement a case-insensitive comparison (otherwise, the = operator uses case-sensitive comparison for strings), or for any other pattern matching supported by `java.util.regex.Pattern` regular expressions.

*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`.

### Usage Notes

To search for all accepted tasks, you must specify the field using a binding expression in the following format:

```
TaskSearchQueryTerm.Fields= #{not empty currentTask.taskInfo.acceptedByList}
TaskSearchQueryTerm.Operator= =
TaskSearchQueryTerm.Value= true
```

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

### See Also

[pub.task:TaskSearchQuery](#)

[pub.task.taskclient:searchTasks](#)

[pub.task.taskclient:searchTasksFields](#)

---

## pub.task.TaskSearchQueryV2

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

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>	<b>Integer</b> Specifies the starting index of the page of search results. Zero represents the first index of the result set.
<i>toIndex</i>	<b>Integer</b> 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>	<b>String</b> 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 <a href="#">pub.task:TaskSearchQueryTerm</a> .

---

Parameters	Description
<i>sortOrder</i>	<p><b>String</b> 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"><li>■ ASC to display the results in ascending order.</li><li>■ DESC to display the results in descending order.</li></ul> <p>The default <i>sortOrder</i> is ascending (ASC).</p>

### See Also

[pub.task:TaskSearchQuery](#)

[pub.task:TaskSearchQueryTerm](#)

[pub.task.taskclient:searchTasksIndexed](#)



---

## 2 Using Task Engine RESTful Web Services

---

■ Overview .....	58
■ Summary of REST Resources and Requests .....	58
■ DELETE a Task Instance .....	60
■ GET Task Instance Information .....	61
■ POST a New Task Instance .....	63
■ PUT Information in a Task Instance .....	65
■ GET Tasks with a Simple Search Request .....	66
■ POST a Complex Task Search Request .....	68
■ GET Task Audit Information .....	74
■ DELETE Audit Entries to Rollback Task .....	74
■ GET or POST a Form Flow Notification .....	75
■ DELETE Task Comment .....	77
■ GET Task Comments .....	77
■ PUT Updates into a Task Comment .....	78
■ POST Task Comment .....	79
■ DELETE Task Attachment .....	80
■ GET Task Attachments .....	81
■ PUT Task Attachment .....	81
■ POST Task Attachment .....	82

---

## 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 35.

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.

## 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
<a href="#">DELETE a Task Instance</a>	Deletes an instance of a task on the Task Engine connected to the Integration Server.
<a href="#">GET Task Instance Information</a>	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.
<a href="#">POST a New Task Instance</a>	Creates a new task instance and adds it to the Task Engine queue.
<a href="#">PUT Information in a Task Instance</a>	Modifies a task instance on the Task Engine connected to the Integration Server.

Request	Description
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 Attachments

Request	Description
<a href="#">DELETE Task Attachment</a>	Deletes the specified task attachment from a task instance.
<a href="#">GET Task Attachments</a>	Returns all the attachments in a task instance.
<a href="#">PUT Task Attachment</a>	Updates the specified task attachment in a task instance.
<a href="#">POST Task Attachment</a>	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>	<b>String</b> 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>	<b>String</b> 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> .

---

### 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 61](#) 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>	<b>String</b> Optional. Specifies whether you want to retrieve the <code>TaskData</code> document as well as the <code>TaskInfo</code> document. Set to: <ul style="list-style-type: none"><li>■ <code>true</code> to retrieve <code>TaskData</code> in addition to <code>TaskInfo</code>.</li><li>■ <code>false</code> (default) to get <code>TaskInfo</code> only.</li></ul>
------------------------	--

---

<i>searchUserTasks</i>	<p><b>String</b> Optional. Specifies whether the service searches all tasks or only the user's inbox. Set to:</p> <ul style="list-style-type: none"> <li>■ <code>true</code> to search only the inbox for user.</li> <li>■ <code>false</code> (default) to search all tasks to which user has access.</li> </ul>
<i>user</i>	<p><b>String</b> 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>. Note: Only tasks to which <i>user</i> has access are searched.</p>

## Output Parameters

---

<i>Tasks</i>	<p><b>Document list</b> 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"> <li>■ <i>TaskID</i> <b>String</b> The identifier assigned to the task instance.</li> <li>■ <i>TaskInfo</i> <b>Document</b> The TaskInfo document containing standard information about the task. See <a href="#">pub.task:TaskInfo on page 42</a> for a description of the fields in this document.</li> <li>■ <i>TaskData</i> <b>Document</b> The TaskData document containing the business data associated with the task. Returned only if <i>includeTaskData</i> is set to <code>true</code>.</li> </ul>
--------------	---

## 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><b>String</b> 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"> <li>■ <code>true</code> to retrieve TaskData in addition to TaskInfo.</li> <li>■ <code>false</code> (default) to get TaskInfo only.</li> </ul>
------------------------	---

---

<i>taskID</i>	<b>String</b> 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.
<i>user</i>	<b>String</b> 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

---

<i>TaskID</i>	<b>String</b> The ID that identifies the task.
<i>TaskInfo</i>	<b>Document</b> The TaskInfo document containing standard information about the task. See <a href="#">pub.task:TaskInfo on page 42</a> for a description of the fields in this document.
<i>TaskData</i>	<b>Document</b> 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> .

## 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>	<b>Document</b> Optional. Any data that you want the task to pass back to the callback service.
<i>CallbackServiceName</i>	<b>String</b> 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 <a href="#">pub.task:TaskCallbackService on page 39</a> .
<i>ruleSet</i>	<b>String</b> 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>	<b>Document</b> Optional. 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>taskQueueInfo</i> or <i>TaskQueueInfo</i>	<b>Document</b> Optional. A TaskQueueInfo document containing basic information about the task (for example, name, priority, list of users to which the task is assigned). See <a href="#">pub.task:TaskQueueInfo on page 46</a> for a description of the fields in this document.
<i>taskTypeID</i>	<b>String</b> 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 <b>Task Configuration</b> panel on the Task Engine Administration page in My webMethods.
<i>user</i>	<b>String</b> 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 *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. By default, this service enables you to update only tasks in Active, Error, and Suspended state. For information about updating tasks with other statuses, see [Usage Notes on page 66](#), below. 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

---

*taskID*                      **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.

*taskInfo* or  
*TaskInfo*                      **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 42](#) for a description of the fields in this document.

*taskData* or  
*TaskData*                      **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

*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

---

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.

By default, this operation enables you to update only tasks in Active, Error, and Suspended state. Tasks with other statuses (such as Completed, Cancelled, or Expired) cannot be updated.

However, a Task Engine optional setting, `#Dupdate.completed.task=true`, is available to apply update actions to a task regardless of its current status. When this option is not present or set to `false`, task updates apply only to tasks in Active, Error, or Suspended status.

For information about setting this option, see "Specifying Task Update Behavior" and "Applying Task Engine Optional Settings" in Chapter 6 of the PDF publication, *webMethods Task Engine User's Guide*.

---

## 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><b>String</b> Optional. Specifies whether the service will return the TaskData document as well as the <a href="#">pub.task:TaskInfo</a> document for each task matching the search criteria. Set to:</p> <ul style="list-style-type: none"><li>■ <code>true</code> to include the TaskData document in the result set.</li><li>■ <code>false</code> (default) to include only the TaskInfo document in the result set.</li></ul>
<i>maxRes</i>	<p><b>String</b> 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><b>String</b> 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"><li>■ <code>true</code> to include non-active tasks from a user's inbox (that is, those tasks already completed by user).</li><li>■ <code>false</code> (default) to return only active tasks.</li></ul>
<i>notOthers</i>	<p><b>String</b> 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"><li>■ <code>true</code> to exclude tasks that are accepted by a user other than <i>user</i>.</li><li>■ <code>false</code> (default) to search all tasks to which <i>user</i> has access.</li></ul>
<i>searchUserTasks</i>	<p><b>String</b> Optional. Specifies whether the service searches all tasks or just the user's inbox. Set to:</p> <ul style="list-style-type: none"><li>■ <code>true</code> to search only the inbox for user.</li><li>■ <code>false</code> (default) to search all tasks to which user has access.</li></ul>
<i>user</i>	<p><b>String</b> 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 42](#) 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 68](#), 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

---

*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 47](#) 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 42](#) 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 49](#). 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 47](#) 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 42](#).

### 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 69](#).

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

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

## 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.

---

	<ul style="list-style-type: none"> <li>■ <code>false</code> (default) to include only the <code>TaskInfo</code> document in the result set.</li> </ul>
<i>searchUserTasks</i>	<p><b>String</b> Optional. Specifies whether the service searches all tasks or only the user's inbox. Set to:</p> <ul style="list-style-type: none"> <li>■ <code>true</code> to search only the inbox for <i>user</i>.</li> <li>■ <code>false</code> (default) to search all tasks to which <i>user</i> has access.</li> </ul>
<i>taskSearchQueryV2</i> or <i>TaskSearchQueryV2</i>	<p><b>Document</b> Optional. The <code>TaskSearchQueryV2</code> document that specifies the search criteria. See <a href="#">pub.task.TaskSearchQueryV2 on page 54</a> for a description of the fields in this document. If <code>TaskSearchQueryV2</code> is null, an error occurs.</p>
<i>user</i>	<p><b>String</b> 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><b>Document list</b> Result set containing the tasks that match the search criteria in <a href="#">pub.task.TaskSearchQueryV2</a>. Each document in the result set has the following structure:</p> <ul style="list-style-type: none"> <li>■ <i>TaskID</i> <b>String</b> The identifier assigned to the task instance.</li> <li>■ <i>TaskInfo</i> <b>Document</b> The <code>TaskInfo</code> document containing standard information about the task. See <a href="#">pub.task:TaskInfo on page 42</a> for a description of the fields in this document.</li> <li>■ <i>TaskData</i> <b>Document</b> 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>.</li> </ul>
--------------	---

### 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

---

- 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 *businessData* 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 54](#) 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:

- **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 54](#).

---

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

### Output Parameters

<i>taskAudits</i>	<b>TaskAudit[]</b> An array of <code>TaskAudit</code> objects representing the audit logs of the task. See <a href="#">pub.task:TaskAudit on page 38</a> 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

---

*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. 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.

**Important:** The 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 at:

<http://communities.softwareag.com/ecosystem/communities/codesamples/webmethods/caf/webmethods-caf-codesamples.html>

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

## Output Parameters

<i>taskComments</i>	<b>TaskComment[]</b> An array of TaskComment objects that represent the comments retrieved from the task instance. See <a href="#">pub.task:TaskComment on page 40</a> 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>	<b>TaskAttachment[]</b> Optional. An array of TaskAttachment objects representing any attachments to be added to the comment being updated. See <a href="#">pub.task:TaskAttachment on page 37</a> for the structure of this data.
<i>commentID</i>	<b>String</b> The unique identifier of a comment to be deleted.

---

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

This REST request deletes the specified task attachment from a task instance. This service applies only to standard task attachments, and is not intended for use with Content Service Platform attachments. 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

---

*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. 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. This service applies only to standard task attachments, and is not intended for use with Content Service Platform attachments. 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

---

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

### Output Parameters

<i>taskAttachments</i>	<b>TaskAttachment[]</b> An array of TaskAttachment objects that represent the attachments in the task instance. See <a href="#">pub.task:TaskAttachment on page 37</a> for the structure of this data.
------------------------	--

---

## PUT Task Attachment

This REST request updates the specified task attachment in a task instance. This service applies only to standard task attachments, and is not intended for use with Content Service Platform attachments. 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>	<b>TaskAttachment</b> The TaskAttachment object representing the new attachment to be used to update the existing attachment. See <a href="#">pub.task.TaskAttachment on page 37</a> for the structure of this data.
<i>attachmentID</i>	<b>String</b> The unique identifier of the attachment to be updated.
<i>encoding</i>	<b>String</b> Optional. Specify the encoding type for the file. Set to: <ul style="list-style-type: none"><li>■ binary</li><li>■ text</li></ul>
<i>taskID</i>	<b>String</b> The unique identifier that the Task Engine assigns to the task instance.
<i>user</i>	<b>String</b> 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 Attachment

This REST request adds the specified attachment to a task instance. This service applies only to standard task attachments, and is not intended for use with Content Service Platform attachments. 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>	<b>TaskAttachment</b> The TaskAttachment object representing the new attachment to be used to update the existing attachment. See <a href="#">pub.task:TaskAttachment on page 37</a> for the structure of this data.
<i>encoding</i>	<b>String</b> Optional. Specify the encoding type for the file. Set to: <ul style="list-style-type: none"><li>■ binary</li><li>■ text</li></ul>
<i>taskID</i>	<b>String</b> The unique identifier that the Task Engine assigns to the task instance.
<i>user</i>	<b>String</b> 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 Web Service API

---

■ Overview .....	87
■ Task Service URLs .....	87
■ Extracting the WSDL for the Task Web Service .....	87
■ Using the Task Web Service Operations .....	88
■ Summary of Available Operations .....	89
■ addTaskAttachment .....	90
■ addTaskComment .....	91
■ countTasksIndexed .....	92
■ deleteTask .....	93
■ deleteTaskAttachment .....	94
■ deleteTaskComment .....	94
■ formFlowTaskNotify .....	95
■ getTask .....	96
■ getTaskAttachments .....	97
■ getTaskAudit .....	97
■ getTaskComments .....	98
■ queueTask .....	98
■ rollbackTask .....	99
■ searchTaskFields .....	100
■ searchTasks .....	101
■ searchTasksIndexed .....	102
■ updateTask .....	104
■ updateTaskAttachment .....	106
■ updateTaskComment .....	106
■ Summary of Data Structures .....	107
■ TaskAttachment .....	108

---

■ TaskAudit .....	109
■ TaskComment .....	110
■ TaskData .....	111
■ TaskInfo .....	112
■ Task .....	117
■ TaskSearchQuery .....	117
■ TaskSearchQueryTerm .....	119
■ TaskSearchQueryV2 .....	123

---

## 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 here:  
`http://hostName :port /services/bizPolicy/task`
- For .NET clients, the task Web service is hosted here:`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 WSDL, log on to My webMethods Server as sysadmin and navigate to following URL:

---

Client Type	WSDL File Location
.NET	<code>http://server:port/services/bizPolicy/taskDotNet?wsdl</code>
Other	<code>http://server:port/services/bizPolicy/task?wsdl</code>

## 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 operations are provided by the task service.

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

Operation	Description
<a href="#">queueTask</a>	Adds a task instance to the Task Engine.
<a href="#">rollbackTask</a>	Enables you to roll back the task to any available audit point in the task's audit history.
<a href="#">searchTasks</a>	Returns an array of tasks that match specified search criteria.
<a href="#">searchTaskFields</a>	Returns a list of tasks that match specified field search criteria.
<a href="#">searchTasksIndexed</a>	Returns a subset of tasks that match the specified search criteria against indexed business data.
<a href="#">updateTask</a>	Modifies an existing task instance on the Task Engine. You use this operation to change information in the task's TaskInfo and TaskData objects. By default, this operation enables you to update only tasks in Active, Error, and Suspended state; for information about updating tasks with other statuses, see <i>Usage Notes</i> .
<a href="#">updateTaskAttachment</a>	Updates the specified task attachment.
<a href="#">updateTaskComment</a>	Updates the specified task comment.

## addTaskAttachment

Adds the specified attachment to a task. This service applies only to standard task attachments, and is not intended for use with Content Service Platform attachments.

### Input Parameters

<i>attachment</i>	<b>TaskAttachment</b> The <a href="#">TaskAttachment</a> object representing the attachment to be added.
<i>encoding</i>	<b>String</b> Specify the encoding type for the file. Valid values include: <ul style="list-style-type: none"> <li>■ binary</li> <li>■ text</li> </ul>

---

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

### Output Parameters

<i>commentID</i>	<b>String</b> 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

---

*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.

*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 117.

## Output Parameters

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

## See Also

[searchTasksIndexed](#)

[TaskSearchQueryV2](#)

[Using the Task Engine Web Service API](#)

---

# deleteTask

Deletes an instance of a task from the Task Engine.

## Input

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

*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.

## 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.

---

## See Also

[searchTasks](#)

[Using the Task Engine Web Service API](#)

---

## deleteTaskAttachment

Deletes the specified task attachment. This service applies only to standard task attachments, and is not intended for use with Content Service Platform attachments.

### Input Parameters

---

<i>attachmentID</i>	<b>String</b> The unique identifier of the attachment to be deleted.
<i>taskID</i>	<b>String</b> The unique identifier that the Task Engine assigns to the task.
<i>user</i>	<b>String</b> 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>	<b>String</b> The unique identifier of the comment to be deleted.
<i>taskID</i>	<b>String</b> The unique identifier that the Task Engine assigns to the task.
<i>user</i>	<b>String</b> 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>	<b>Boolean</b> Optional. Specifies whether you want to retrieve the <a href="#">TaskData</a> object as well as the <a href="#">TaskInfo</a> object. Set to true to retrieve <a href="#">TaskData</a> in addition to <a href="#">TaskInfo</a> . Default is false.
<i>taskId</i>	<b>String</b> The ID that identifies the task that you want to retrieve.
<i>user</i>	<b>String</b> 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

---

*Task*                    **Task** A Task object containing the task's TaskInfo object and, optionally, its TaskData object. See [Task on page 117](#) 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.

### See Also

[searchTasks](#)

[Using the Task Engine Web Service API](#)

---

## getTaskAttachments

Returns all the attachments in a task. This service applies only to standard task attachments, and is not intended for use with Content Service Platform 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.

---

## 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

---

## 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 111](#) 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 112](#) 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

---

*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.

## 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>	<p><b>String</b> 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"><li>■ <code>true</code> to include the TaskData in the result set.</li><li>■ <code>false</code> (default) to include only the TaskInfo in the result set.</li></ul>
<i>taskSearchQuery</i>	<p><b>TaskSearchQuery</b> The TaskSearchQuery object, which specifies the search criteria. See <a href="#">TaskSearchQuery on page 117</a> 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><b>String</b> 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> <p><b>Note:</b> Only tasks to which <i>user</i> has access are searched.</p>
<i>fields</i>	<p><b>String</b> 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 <a href="#">TaskSearchQueryTerm</a>. For example, suppose you want to return the following fields as result of the query:</p> <ul style="list-style-type: none"><li>■ <code>taskID</code></li><li>■ The business data field <code>orderID</code> from task data documents.</li></ul> <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>

---

## 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.

### See Also

[TaskSearchQuery](#)

[TaskSearchQueryTerm](#)

[Using the Task Engine Web Service API](#)

---

## searchTasks

Returns an array of tasks that match specified search criteria.

### Input Message

---

*includeTaskData*      **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:

- `true` to include the `TaskData` in the result set.

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

*taskSearchQuery*

**TaskSearchQuery** The `TaskSearchQuery` object, which specifies the search criteria. See [TaskSearchQuery on page 117](#) 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.

*userTasks*

**Boolean** Optional. Specifies whether the operation searches all tasks or just 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 117](#).

## Output Parameters

---

*tasks*

**Task[ ]** An array of `Task` objects representing the tasks that matched the search criteria in *TaskSearchQuery*. See [Task on page 117](#) 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 101](#).

## 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. 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.

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 117](#).

---

## 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.

## See Also

[TaskSearchQueryV2](#)

[TaskSearchQuery](#)

[Using the Task Engine Web Service API](#)

---

# 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. By default, this operation enables you to update only tasks in Active, Error, and Suspended state; for information about updating tasks with other statuses, see *Usage Notes*, below.

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

---

*taskData*                **Map** The Map object that contains the business data that you want to assign to the task. See [TaskData on page 111](#) for a description of the Map object used by *taskData* .

---

<i>taskID</i>	<b>String</b> The ID that identifies the task that you want to modify.
<i>taskInfo</i>	<b>TaskInfo</b> Object that specifies the changes you want to make to the data in the task's TaskInfo object. See <a href="#">TaskInfo on page 112</a> for a description of the fields in this object.
<i>user</i>	<b>String</b> 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.

By default, this operation enables you to update only tasks in Active, Error, and Suspended state. Tasks with other statuses such as Completed, Cancelled, or Expired cannot be updated.

However, a Task Engine optional setting, `#Update.completed.task=true`, is available to apply update actions to a task regardless of its current status. When this option is not present or set to `= false`, task updates apply only to tasks in Active, Error, or Suspended status.

For information about setting this option, see “Specifying Task Update Behavior” and “Applying Task Engine Optional Settings” in Chapter 6 of the PDF publication, *webMethods Task Engine User’s Guide*.

---

## See Also

[searchTasks](#)

[Using the Task Engine Web Service API](#)

---

## updateTaskAttachment

Updates the specified task attachment. This service applies only to standard task attachments, and is not intended for use with Content Service Platform attachments.

### Input Parameters

---

<i>attachment</i>	<b>TaskAttachment</b> The <a href="#">TaskAttachment</a> object representing the updated attachment to be used to update the existing attachment.
<i>attachmentID</i>	<b>String</b> The unique identifier of the attachment to be updated.
<i>encoding</i>	<b>String</b> Specify the encoding type for the file. Valid values include: <ul style="list-style-type: none"><li>■ binary</li><li>■ text</li></ul>
<i>taskID</i>	<b>String</b> The unique identifier that the Task Engine assigns to the task.
<i>user</i>	<b>String</b> 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>	<b>TaskAttachment[]</b> An array of <a href="#">TaskAttachment</a> objects representing any attachments to be added to the comment being updated.
--------------------	---

---

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

### Output Parameters

None.

## Summary of Data Structures

---

The following complex data structures are used by the task service.

Data Structure	Description
<a href="#">TaskAttachment</a>	Document type representing an attachment.
<a href="#">TaskAudit</a>	Document type containing information returned as output from <a href="#">getTaskAudit</a> .
<a href="#">TaskComment</a>	Document type representing a comment in a task.
<a href="#">TaskData</a>	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.
<a href="#">TaskInfo</a>	Contains standard information about a task. You specify certain fields in TaskInfo when you start a task using the <a href="#">queueTask</a> operation. After a task is queued, you can modify fields in TaskInfo using the <a href="#">updateTask</a> operation.

---

Data Structure	Description
<a href="#">Task</a>	Data object returned as output from the <a href="#">getTask</a> and <a href="#">searchTasks</a> operations. This data structure contains the TaskInfo and TaskData objects associated with a task.
<a href="#">TaskSearchQuery</a>	Used as input to the <a href="#">searchTasks</a> operation.
<a href="#">TaskSearchQueryTerm</a>	Contains search criteria for the <a href="#">searchTasks</a> operation.
<a href="#">TaskSearchQueryV2</a>	Use the TaskSearchQueryV2 data type as input to <a href="#">searchTasks</a> .

## TaskAttachment

Document type representing an attachment.

### Parameters

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

---

*deletable*      **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.

---

## TaskAudit

Document type containing information returned as output from [getTaskAudit](#).

---

### Parameters

*createdDate*      **Date** The date and time the operation was applied to the audited task.

*difference*      **String[ ]** An array of strings representing the changes made as part of an audit log entry.

*sourceURI*      **String** URI of the source of this change.

*TaskURI*      **String** URI of the audited task.

*id*      **Integer** Unique ID of this audit log entry.

*operation*      **Integer** Numerical representation of the operation performed. See Usage Notes below for more information.

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 *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.

*taskVersionNumber*      **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>	<b>TaskAttachment [ ]</b> An array of attachments to be enclosed inside the comment.
<i>id</i>	<b>String</b> Unique ID of the comment.
<i>displayText</i>	<b>String</b> Displayable representation of the text associated with the comment.
<i>text</i>	<b>String</b> Text associated with the comment.#
<i>timeStamp</i>	<b>Date</b> The date and time of last modification of the comment.
<i>userDisplayName</i>	<b>String</b> Displayable representation of the name of user associated with the comment.
<i>userID</i>	<b>String</b> 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.

<b>Key</b>	<b>Value</b>
<i>firstName</i>	<b>String</b> Anna
<i>lastName</i>	<b>String</b> Pareo
<i>emailAddress</i>	<b>String</b> apareo@msx.com
<i>acctNum</i>	<b>String</b> J5468-6268-508
<i>TransactionCodes</i>	<b>String [ ]</b> 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:

<b>Field</b>	<b>Description</b>
<i>key</i>	<b>String</b> A name given to the field.
<i>value</i>	<b>AnyType</b> 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>	<b>String</b> Anna
1	<i>lastName</i>	<b>String</b> Pareo
2	<i>emailAddress</i>	<b>String</b> apareo@msx.com
3	<i>acctNum</i>	<b>String</b> J5468-6268-508
4	<i>TransactionCodes</i>	<b>String[ ]</b> 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>	<b>String[ ]</b> 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>	<b>String[ ]</b> 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>	<b>Map</b> Optional. Contains data that is used by the process runtime. For internal use only.

---

<b>Element</b>	<b>Description</b>
<i>auditContext</i>	<b>String</b> 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>	<b>String</b> Read-only. The unique ID of the collaboration process flow created when a task is used for collaboration workflow.
<i>collaborationStepID</i>	<b>String</b> Read-only. The unique ID of the task step in the process flow when task is used for collaboration workflow.
<i>createdBy</i>	<b>String</b> Read-only. The user ID (on My webMethods Server) of the user that initially queued the task.
<i>createdDate</i>	<b>DateTime</b> Read-only. The date and time when the task was queued.
<i>currentUserAccepted</i>	<b>Boolean</b> Read-only. Returns true if the task is accepted by the current user.
<i>customTaskID</i>	<p><b>String</b> 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 <a href="#">searchTasks</a> operation.</p> <p><b>Note:</b> 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>	<b>String</b> 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>	<b>String[ ]</b> Optional. List of user IDs who delegated this task.
<i>delegatedToList</i>	<b>String[ ]</b> Optional. List of user IDs to whom the task was delegated
<i>delegationMap</i>	<b>Map</b> 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.

---

Element	Description
	<ul style="list-style-type: none"> <li>■ <i>key</i> <b>String</b> User ID of delegating user.</li> <li>■ <i>value</i> <b>String</b> User ID of target user.</li> </ul>
<i>description</i>	<p><b>String</b> 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 <b>Details View</b> tab in My Inbox and in the Task Management Results list.</p> <p>Maximum length is 255 characters.</p>
<i>errorCode</i>	<p><b>String</b> 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.</p>
<i>errorMessage</i>	<p><b>String</b> 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.</p>
<i>expireDate</i>	<p><b>DateTime</b> 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>lastAcceptedBy</i>	<p><b>String</b> Read-only. The user ID (on My webMethods Server) of the last user to accept the task. If no user has accepted the task, this field is null.</p>
<i>lastAcceptedDate</i>	<p><b>DateTime</b> Read-only. The date and time the task was last accepted</p>
<i>lastModifiedBy</i>	<p><b>String</b> 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.</p>

---

<b>Element</b>	<b>Description</b>
<i>lastModifiedDate</i>	<b>DateTime</b> Read-only. Date and time when the task was last updated.
<i>name</i>	<p><b>String</b> Optional. The name of the task. This name appears in various places in the My webMethods user interface, such as on the <b>Details View</b> tab in My Inbox and in the Task List Management Results list.</p> <p>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>	<p><b>String</b> 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.</p>
<i>priority</i>	<p><b>String</b> Optional. The priority of the task. Must contain one of the following values:</p> <p>none low medium high critical</p> <p>Values are case-sensitive.</p>
<i>processInstanceID</i>	<b>String</b> 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>	<b>Integer</b> 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>	<b>String</b> 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>	<b>String</b> 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><b>String</b> Optional. The state of the task. Must contain one of the following values:</p> <ul style="list-style-type: none"> <li>■ <b>active.</b> Task is active and is available in the user's inbox queue. Only active tasks can be modified by the user.</li> </ul>

---

Element	Description
	<ul style="list-style-type: none"> <li data-bbox="540 323 1364 426">■ <code>suspended</code>. Task is suspended and is not available in the user's inbox queue. Suspended tasks still appear in the Task Management Results list.</li> <li data-bbox="540 447 1364 550">■ <code>completed</code>. Task has been successfully completed and is no longer available in the user's inbox queue. Completed tasks still appear in the Task Management Results list.</li> <li data-bbox="540 571 1364 674">■ <code>cancelled</code>. Task has been cancelled and is no longer available in the user's inbox queue. Cancelled tasks still appear in the Task Management Results list.</li> <li data-bbox="540 695 1364 798">■ <code>expired</code>. Task has expired and is no longer available in the user's inbox queue. Expired tasks still appear in the Task Management Results list.</li> <li data-bbox="540 819 1364 961">■ <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.</li> </ul> <p data-bbox="540 968 1364 1035">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="540 1077 1364 1180"><b>String</b> 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="540 1224 1364 1327"><b>Integer</b> 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>taskID</i>	<p data-bbox="540 1371 1364 1438"><b>String</b> Read only. A unique identifier assigned to the task by the Task Engine when the task is queued.</p>
<i>taskTypeID</i>	<p data-bbox="540 1482 1364 1623"><b>String</b> 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.</p>
<i>taskURL</i>	<p data-bbox="540 1667 1364 1770"><b>String</b> 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.</p>
<i>taskVersionNumber</i>	<p data-bbox="540 1814 1364 1908"><b>Integer</b> Optional. This is the current version number of the task record that is automatically incremented for each task update action.</p>

---

Element	Description
	This property can be used in the <code>updateTask()</code> service. If a value different than 0 is passed in when executing <code>updateTask()</code> , 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.

---

## 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>	<b>Map</b> Optional. Contains the business data associated with a task. See <a href="#">TaskData on page 111</a> for a description of this object.
<i>taskInfo</i>	<b>TaskInfo</b> Contains standard information that the Task Engine maintains about a task. See <a href="#">TaskInfo on page 112</a> 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>	<b>Boolean</b> Optional. Exclude tasks that are accepted by a user other than the user ID used in the search. Default is false.  This parameter is ignored if you set the <i>userTasks</i> parameter of <a href="#">searchTasks</a> to <code>false</code> .
<i>maxResults</i>	<b>Integer</b> 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.

Element	Description
<i>showNonActiveTasks</i>	<p><b>Note:</b> This element does not apply to tasks with an indexed search provider. In this case, the specified value is ignored.</p> <p><b>Boolean</b> 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"> <li>■ <code>true</code> to return non-active tasks.</li> <li>■ <code>false</code> (default) to return active tasks only.</li> </ul> <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><b>TaskSearchQueryTerm[]</b> Specifies the <a href="#">TaskSearchQueryTerm</a> object that contains an array of <a href="#">TaskSearchQueryTerm</a> objects that specify the query criteria for the search. See <a href="#">TaskSearchQueryTerm on page 119</a> for a description of the fields in this object.</p> <p>When a search is performed, the criteria specified by the <a href="#">TaskSearchQueryTerm</a> objects in <i>terms</i> are combined using the logical "AND" operator. Only tasks that satisfy the terms specified by all <a href="#">TaskSearchQueryTerm</a> objects in the <i>terms</i> array are returned in the result set.</p>

### 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><b>String[ ]</b> 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><b>Working with Standard Task Fields</b></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 lastAcceptedBy name priority taskID taskTypeID parentTaskID processInstanceID processModelID processModelVersion status stepID stepIteration </pre>

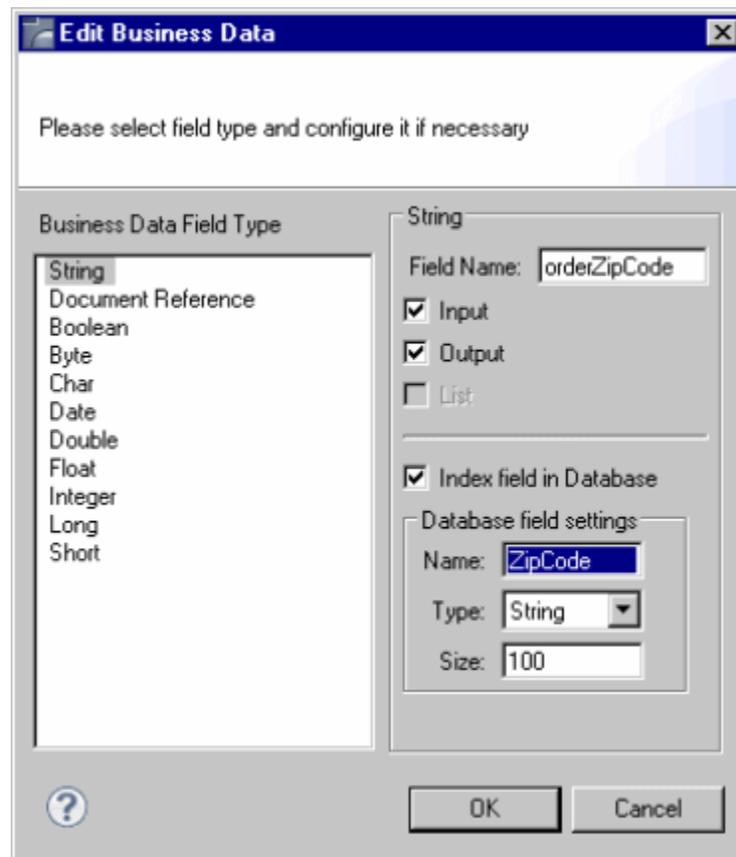
---

Element	Description
	<p>To search standard fields in TaskData (the task business data), specify the field using a binding expression in this format:</p> <pre data-bbox="474 407 1065 432"># {currentTask.taskData.pathToDataElement }</pre> <p>For example, this binding expression:</p> <pre data-bbox="474 499 1260 525"># {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> <p><b>Working with Indexed Fields</b></p> <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 data-bbox="474 1075 1346 1297" style="list-style-type: none"> <li>1. Open the task that contains the indexed data field.</li> <li>2. On the <b>Business Data</b> tab, select the field you want to work with.</li> <li>3. Click <b>Edit</b>.</li> <li>4. In the Edit Business Data dialog box, obtain the indexed field name from the <b>Name</b> field in the <b>Database field settings</b> area.</li> </ol>

---

**Element****Description**

---



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

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

*Continued on the next page*

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 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.

When the `like` or `not like` operator is used, *Value* can specify a valid regular expression pattern that is used to match to a field value being tested. For example, it can be used to implement a case-insensitive comparison (otherwise, the `=` operator uses case-sensitive comparison for strings), or for any other pattern matching supported by `java.util.regex.Pattern` regular expressions.

*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>	<b>Integer</b> Specifies the starting index of the page of search results. Zero represents the first index of the result set.
<i>toIndex</i>	<b>Integer</b> 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>	<p><b>String</b> 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 <a href="#">TaskSearchQueryTerm</a> on page 119.</p>
<i>sortOrder</i>	<p><b>String</b> 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"> <li>■ ASC to display the results in ascending order.</li> <li>■ DESC to display the results in descending order.</li> </ul> <p>The default <i>sortOrder</i> is ascending (ASC).</p>

---

<b>Field Name</b>	<b>Description</b>
<i>principalID</i>	<b>String</b> 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.
<i>checkPermissions</i>	<b>Boolean</b> Optional. Specifies if the current user's permission are checked prior to the task search. Default is <code>true</code> .
<i>invocationID</i>	This parameter is for internal use only. Do not use.

### **See Also**

[searchTasksIndexed](#)

[TaskSearchQuery](#)

[Using the Task Engine Web Service API](#)

# Index

## A

- addTaskAttachment 15
  - REST service 82
  - web service 90
- addTaskComment
  - built-in service 16
  - REST service 79
  - web service 91

## B

- built-in service 15
- built-in services API
  - summary of services 13
- business data
  - structure of for Web service 111
  - updating with built-in service 32

## C

- countTasksIndexed 16

## D

- data structures
  - Task 117
  - TaskData 111
  - TaskInfo 112
  - TaskSearchQuery 117
  - TaskSearchQueryTerm 119
- deleteTask operation 93
- deleteTask service 18
- deleteTaskAttachment
  - built-in service 19
  - REST service 80
  - web service 94
- deleteTaskComment 19
  - REST service 77
  - web service 94
- document types
  - Task 37
  - TaskAttachment
    - built-in service 37
    - web service 108
  - TaskAudit 38
    - web service 109
  - TaskComment 40
    - web service 110

- TaskCompletionInfo 41
- TaskFields 42
- TaskInfo 42
- TaskQueueInfo 46
- TaskSearchQuery 47
- TaskSearchQueryTerm 49
- documentation
  - using effectively 7

## E

- endpoint for task Web service 87

## F

- formFlowTaskNotify
  - built-in service 20
  - REST request 75
  - web service 95

## G

- getTask operation 96
- getTask service 21
- getTaskAttachments
  - built-in service 23
  - REST service 81
  - web service 97
- getTaskAudit
  - built-in service 23
  - REST request 74
  - web service 97
- getTaskComments
  - built-in service 23
  - RESTS service 77
  - web service 98

## M

- mapping
  - TaskData array 111
  - user delegation 113

## N

- notifying a task workflow component
  - with REST service 75
- notifying a task workflow component with built-in service 20
- notifying a task workflow component with webservice 95

**O**

operations, Web service 89

**P**

pub.opentask  
  \_delete service 60  
  \_get service 61  
  \_post service 63  
  \_put service 65  
pub.opentasksearch  
  \_get service 66  
  \_post service 68  
pub.taskclient  
  deleteTask 18  
pub.task  
  Task 37  
  TaskAttachment 37  
  TaskAudit 38  
  TaskCallbackService 39  
  TaskComment 40  
  TaskCompletionInfo 41  
  TaskFields 42  
  TaskInfo 42  
  TaskQueueInfo 46  
  TaskSearchQuery 47  
  TaskSearchQueryTerm 49  
pub.task.taskclient  
  addTaskAttachment 15  
  addTaskComment 16  
  countTasksIndexed 16  
  deleteTaskAttachment 19  
  deleteTaskComment 19  
  formFlowTaskNotify 20  
  getTask 21  
  getTaskAttachments 23  
  getTaskAudit 23  
  getTaskComments 23  
  queueTask 24  
  rollbackTask 26  
  searchTaskFields 28  
  searchTasks 26  
  updateTask 32  
  updateTaskAttachment 34  
  updateTaskComment 35  
pub.task.TaskSearchQueryV2 54

**Q**

queueTask operation 98  
queueTask service 24

**R**

REST requests  
  formFlowTaskNotify 75  
  getTaskAudit 74  
  rollbackTask 74  
REST services  
  addTaskAttachment 82  
  addTaskComment 79  
  deleteTaskAttachment 80  
  deleteTaskComment 77  
  getTaskAttachments 81  
  getTaskComments 77  
  updateTaskAttachment 81  
  updateTaskComment 78  
RESTful services API  
  described 58  
retrieving a task  
  with built-in service 21  
  with Web service operation 96  
rollbackTask  
  built-in service 26  
  REST request 74  
  web service 99

**S**

searchTasks operation 101  
searchTasks service 26  
searchTasksFields operation 100  
services, built-in  
  TaskSearchQuery document type 47  
services, built-in  
  \_delete service 60  
  \_get search service 66  
  \_get service 61  
  \_post search service 68  
  \_post service 63  
  \_put service 65  
  deleteTask service 18  
  formFlowTaskNotify service 20  
  getTask service 21  
  queueTask service 24  
  searchTasks service 26  
  Task document type 37  
  TaskAttachment document type 37  
  TaskAudit document type 38

- TaskCallbackService specification 39
- TaskComment document type 40
- TaskCompletionInfo document type 41
- TaskFields document type 42
- TaskInfo document type 42
- TaskQueueInfo document type 46
- TaskSearchQueryTerm document type 49
- updateTask service 32
- services, REST
  - formFlowTaskNotify request 75
- services, web
  - formFlowTaskNotify service 95
  - TaskAudit document type 109
  - TaskComment document type 110
- services,web
  - TaskAttachment document type 108
- specification, TaskCallbackService 39

## T

- Task
  - built-in service document type 37
- Task data structure 117
- task Web service URL 87
- task workflow notify
  - built-in service 20
  - REST request 75
  - web service 95
- TaskAttachment
  - built-in service document type 37
  - web service document type 108
- TaskAudit
  - built-in service document type 38
  - web service document type 109
- TaskCallbackService specification 39
- TaskComment
  - built-in service document type 40
  - web service document type 110
- TaskCompletionInfo
  - built-in service document type 41
- TaskData array, mapping 111
- TaskData data structure 111
  - for .NET Web service clients 111
- TaskFields
  - built-in service document type 42
- TaskInfo
  - built-in service document type 42
  - Web service data structure 112
- TaskQueueInfo document type 46

- tasks
  - deleting
    - with built-in restful service 60
    - with built-in service 18
    - with Web service operation 93
  - getting
    - with built-in restful search service 66
    - with built-in service 21
    - with Web service 96
  - listing
    - with built-in restful service 61
  - notifying
    - with built-in service 20
    - with REST service 75
    - with web service 95
  - posting
    - with built-in restful service 63
    - with REST search service 68
  - putting
    - with built-in restful service 65
  - queuing
    - with built-in service 24
    - with Web service 98
  - searching for
    - with built-in service 26
    - with Web service 100, 101
  - updating
    - with built-in service 32
    - with Web service 104
- Web service data structures
  - Task 117
  - TaskData 111
  - TaskInfo 112
  - TaskSearchQuery 117
  - TaskSearchQueryTerm 119
- TaskSearchQuery
  - built-in service document type 47
  - Web service data structure 117
- TaskSearchQueryTerm
  - built-in service document type 49
  - Web service data structure 119
- TaskSearchQueryV2 54

## U

- updateTask operation 104
- updateTask service 32
- updateTaskAttachment
  - built-in service 34

- REST service 81
- web service 106
- updateTaskComment
  - built-in service 35
  - REST service 78
  - web service 106
- URL for task Web service 87
- user parameter, for Web service operations 88

## W

- Web service
  - data structures
    - Task 117
    - TaskData 111
    - TaskInfo 112
    - TaskSearchQuery 117
    - TaskSearchQueryTerm 119
  - URLs for 87
- Web service API
  - overview of 87
  - summary of services 89
- Web service operations
  - deleteTask 93
  - getTask 96
  - queueTask 98
  - searchTasks 100, 101
  - updateTask 104
  - user input parameter 88
  - using 88
- web services
  - addTaskAttachment 90
  - addTaskComment 91
  - deleteTaskAttachment 94
  - deleteTaskComment 94
  - formFlowTaskNotify 95
  - getTaskAttachments 97
  - getTaskAudit 97
  - getTaskComments 98
  - rollbackTask 99
  - TaskAttachment document 108
  - TaskAudit 109
  - TaskComment document 110
  - updateTaskAttachment 106
  - updateTaskComment 106
- WSDL extraction, for task Web service 87
  - description of 87
  - TaskData structure 111
  - URL for task Web-service 87
  - WSDL extraction 88
  - \_delete task service 60
  - \_get task search service 66
  - \_get task service 61
  - \_post task search service 68
  - \_post task service 63
  - \_put task service 65

## Symbols

- .NET client