

IBM InfoSphere Optim  
Version 9 Release 1

*Programming for Optim Service  
Interface*





IBM InfoSphere Optim  
Version 9 Release 1

*Programming for Optim Service  
Interface*



**Note**

Before using this information and the product it supports, read the information in "Notices" on page 41.

**Version 9 Release 1**

This edition applies to version 9, release 1, modification 0 of IBM InfoSphere Optim solution components and to all subsequent releases and modifications until otherwise indicated in new editions.

© Copyright IBM Corporation 2012, 2013.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

---

# Contents

<b>About this publication</b>	<b>v</b>
-------------------------------	----------

<b>Programming for the service interface</b>	<b>1</b>
--	----------

Starting the service interface on an application server	1
Service interface methods	2
Get Optim Directory Connections method	2
Service methods	3
Service execution methods	5
Service schedule methods	6
Service instance methods	12
Component methods	16
Work order methods	17
Command-line utility for service interface	27
Get connections (optimcmd -connection)	28
Get service (optimcmd -service)	28
Get servers (optimcmd -rsi)	29
Run service (optimcmd -run)	30
Monitor service instances (optimcmd -monitor)	31
Display work orders (optimcmd -workorder)	32
Display work order states (optimcmd -workorder display=states)	33

Display work order transitions (optimcmd -workorder display=transitions)	33
Display workflows (optimcmd -workorder display=workflows)	34
Create work order (optimcmd -workorder action=create)	35
Change work order state (optimcmd -workorder action=transition)	36
Change work order (optimcmd -workorder action=update)	36
Delete work order (optimcmd -workorder action=purge)	37
Run work order (optimcmd -workorder action=execute)	38

<b>Notices</b>	<b>41</b>
----------------	-----------

Trademarks	43
------------	----

<b>Index</b>	<b>45</b>
--------------	-----------



---

## About this publication

This document describes the functions of Optim<sup>™</sup> Service Interface. Use this information to create applications that use InfoSphere<sup>®</sup> Optim solution components to run and manage data management services.





---

## Programming for the service interface

Optim Service Interface is a web application that provides a public interface into the IBM® InfoSphere Optim data management solution environment. (Optim Service Interface is also known as the *service interface*.) The service interface can be used by other applications to run, monitor, and manage services.

### How the service interface works

The service interface accepts HTTP requests from other applications. Each HTTP request must follow a method to accomplish a specific task. Some methods require the submission of XML request payloads. The service interface processes each request that the service interface receives. When the request is complete, the service interface returns an HTTP response code and output document where applicable to the other application.

### Prerequisites

To use the service interface, you must first install the service interface on a computer. The service interface is installed as a part of the IBM InfoSphere Optim Web Applications package, which also contains the manager and the connection manager. You must then configure and deploy the connection manager to an application server before you can deploy and start the service interface. For information about how to install and deploy the service interface, see the manager installation information and the component configuration information for InfoSphere Optim solution components.

### Security

To secure the service interface from unauthorized use, use a firewall to control access to the application server on which the service interface is deployed. Also, ensure that any applications that use the service interface can authenticate users and log requests that are sent to the service interface.

---

## Starting the service interface on an application server

To use the service interface to run and manage test- or production-level services, you must first start the service interface on its application server. After the service interface is started on the application server, any application can access the service interface at any time.

Before you can start the service interface, you must first install the service interface on a computer. The service interface is installed as a part of the IBM InfoSphere Optim Web Applications package, which also contains the manager and the connection manager. You must then configure and deploy the connection manager to an application server before you can deploy and start the service interface.

To start the service interface on an application server:

1. Start the application server. If the application server is set to start the service interface web application automatically, then the service interface is started immediately after the application server. If you deployed the service interface to the version of WebSphere® Application Server Community Edition that is delivered with the manager, then complete the following step. In this step, *shared\_installation\_directory* is the installation directory that you specified for the manager.
  - Microsoft Windows computers: Click **Start > All Programs > IBM InfoSphere > Optim > Start WAS-CE**, or run the script *shared\_installation\_directory\WebSphere\AppServerCommunityEdition\bin\startup.bat*.
  - Linux or UNIX computers: Run the script *shared\_installation\_directory/WebSphere/AppServerCommunityEdition/bin/startup.sh*.

2. If the service interface is not started after a few minutes, start the service interface web application by using the application server console. If you deployed the service interface to the version of WebSphere Application Server Community Edition that is delivered with the manager, then complete the following steps:
  - a. Use a web browser to access and sign into the Administrative Console. The default location is at `http://hostname:port/console/`, where *hostname* is the host name or IP address of the WebSphere Application Server Community Edition computer and *port* is the port number. The default port number is 8080. Use user ID system and password manager to access the Administrative Console.
  - b. Click **Web App WARs**.
  - c. Click **Start** for the component with an URL of `/server`.

To automate the starting of the service interface after you restart the computer, configure the application server as a Windows service or Linux or UNIX daemon.

---

## Service interface methods

The service interface supports a set of methods that are implemented by using HTTP (Hypertext Transfer Protocol) services. Each method uses a distinct HTTP resource URL that indicates the method to use and any parameters to use with the method.

### Resource URLs

The format of each resource URL is `http://hostname:port/server/method_URI`.

- *hostname* is the host name or IP address of the application server.
- *port* is the port number of the application server.
- *method\_URI* is the URI that is used by the method. Depending on the method that is used, the method URI can contain user-specified parameters.

For example, if the application server uses port 8080 on a computer with host name `appserver`, and a method uses the URI `/monitor/`, the method resource URL is `http://appserver:8080/server/monitor/`.

### Supported encoding

Use UTF-8 encoding in any parameters resource URL and in all request payloads.

## Get Optim Directory Connections method

Use the Get Optim Directory Connections method to retrieve a list of all Optim directory connections in the connection manager.

HTTP method: GET

URI: `/connection/`

Request payload content type: None

Response payload content type: `application/xml`

Expected response: HTTP/1.1 200 OK

### Response payload example

The following example shows a sample XML response for this method.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:connections xmlns:ns2="http://www.ibm.com/optim/xsd/connection/9.1.0">
  <ns2:connection>
    <id>e1a93332-701b-48d7-bdc6-c20adb09ba51</id>
    <name>OPTIMDIR1</name>
    <description>Optim directory on local Oracle XE</description>
    <status>Complete</status>
    <vendor>Oracle</vendor>
    <version>11.2</version>
    <jdbcUrl>jdbc:oracle:thin:@localhost:1521:xe</jdbcUrl>
  </ns2:connection>
</ns2:connections>
```

## Response codes

The following response codes can be received when the method is complete.

Response code	Description
200	OK. The connection list was successfully returned.
400	Bad Request. A malformed request was submitted.
404	Not Found. No connections exist.

## Service methods

Use service methods to get a list of services and to get information about these services. Service methods are a prerequisite for most other tasks in the service interface.

### Get All Services method

Use the Get All Services method to receive an XML file that contains a list of all services in an Optim directory. You can then get more detailed information about a service by using the Get Service method.

HTTP method: GET

URI: */service/connection\_name*

Request payload content type: None

Response payload content type: application/xml

Expected response: HTTP/1.1 200 OK

### Parameters

The method URI contains the following parameters.

Name	Description	Required
<i>connection_name</i>	Enter the name that identifies the Optim directory connection whose information you want to view.	Y

### Response payload example

The following example shows a sample XML response for this method.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:services xmlns="http://www.w3.org/2005/Atom" xmlns:ns2="http://www.ibm.com/optim/xsd/srm/9.1.0">
  <link href="http://interface:8080/server/service/OPTIMDIR1/234d7245-aa47-44cb-b647-650734d200fd"/>
  <link href="http://interface:8080/server/service/OPTIMDIR1/79550682-e362-464d-9a0b-eae9a046083f"/>
  <link href="http://interface:8080/server/service/OPTIMDIR1/7c354125-4f2a-4d94-bfab-0711d2270af8"/>
</ns2:services>
```

## Response codes

The following response codes can be received when the method is complete.

Response code	Description
200	OK. The service list was successfully returned.
400	Bad Request. A malformed request was submitted.
404	Not Found. No services exist.

## Get Service method

Use the Get Service method to receive an XML file that contains information about a service.

HTTP method: GET

URI: /service/connection\_name/service\_id

Request payload content type: None

Response payload content type: application/xml

Expected response: HTTP/1.1 200 OK

## Parameters

The method URI contains the following parameters.

Name	Description	Required
<i>connection_name</i>	Enter the name that identifies the Optim directory connection that contains the service.	Y
<i>service_id</i>	Enter the ID that identifies the service whose information you want to view.	Y

## Response payload example

The following example shows an XML file that contains information about the requested service. The information includes the service name, path, type, and platform.

```
<?xml version="1.0" encoding="UTF-8"?>
<srm:service xmlns:atom="http://www.w3.org/2005/Atom"
  xmlns:srm="http://www.ibm.com/optim/xsd/srm/9.1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.ibm.com/optim/xsd/srm/9.1.0
  resource.xsd ">
  <serviceId>fbf1cdd5-bdf7-4682-96ad-722672af0001</serviceId>
  <serviceName>SIDEMO.ERGL23393T</serviceName>
```

```

<servicePath>OPTIMDIR/SMK_V910</servicePath>
<serviceType>com.ibm.nex.model.oim.distributed.ExtractRequest</serviceType>
<servicePlatform>com.ibm.nex.ois.runtime.productplatform.distributed</servicePlatform>
</srm:service>

```

## Response codes

The following response codes can be received when the method is complete.

Response code	Description
200	OK. The service information was successfully returned.
400	Bad Request. A malformed request was submitted.
404	Not Found. The service ID does not exist.

## Service execution methods

Use service execution methods to run services.

### Run Service method

Use the Run Service method to run a service. The service can be run without any changes. Alternatively, you can run the service with input values that are different from the values that are saved with the service.

HTTP method: POST

URI: `/execute/connection_name/service_id`

Request payload content type: application/xml

Response payload content type: application/xml

Expected response: HTTP/1.1 201 Created

### Parameters

The method URI contains the following parameters.

Name	Description	Required
<i>connection_name</i>	Enter the name that identifies the Optim directory connection that contains the service.	Y
<i>service_id</i>	Enter the service ID that identifies the service that you want to run.	Y

### Request payload example

The following example shows a request to run a service.

```

<?xml version="1.0" encoding="UTF-8"?>
<sem:serviceRequestExecutionInput
  xmlns:sem="http://www.ibm.com/optim/xsd/sem/9.1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.ibm.com/optim/xsd/sem/9.1.0
  resource.xsd">
  <serviceId>783015a5-5ad4-43a8-b3be-a45acaba081a</serviceId>
  <serviceName>SIDEMO.ERGL23393T</serviceName>

```

```

    <servicePath>OPTIMDIR/SMK_V910</servicePath>
    <rsiURL>http://servercomputer:12000/</rsiURL>
    <executedBy>jdoe</executedBy>
</sem:serviceRequestExecutionInput></p>

```

The following example shows a request to run the service from the first example with two changed input values (override values).

```

<p><?xml version="1.0" encoding="UTF-8"?>
<sem:serviceRequestExecutionInput
  xmlns:sem="http://www.ibm.com/optim/xsd/sem/9.1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.ibm.com/optim/xsd/sem/9.1.0
  resource.xsd">
  <serviceId>783015a5-5ad4-43a8-b3be-a45acaba081a</serviceId>
  <serviceName>SIDEMO.ERGL23393T</serviceName>
  <servicePath>OPTIMDIR/SMK_V910</servicePath>
  <rsiURL>http://servercomputer:12000/</rsiURL>
  <executedBy>jdoe</executedBy>
  <overrides>
    <override>
      <id>override-id-000001</id>
      <type>TypeOne</type>
      <value>Value One</value>
    </override>
    <override>
      <id>override-id-000002</id>
      <type>TypeTwo</type>
      <value>Value Two</value>
    </override>
  </overrides>
</sem:serviceRequestExecutionInput></p>

```

## Response payload example

The following example shows the response that confirms that the service was started.

```

<?xml version="1.0" encoding="UTF-8"?>
<sem:executionResults
  xmlns:sem="http://www.ibm.com/optim/xsd/sem/9.1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.ibm.com/optim/xsd/sem/9.1.0
  resource.xsd">
  <executionId>fbf1cdd5-bdf7-4682-96ad-722672af0001</executionId>
  <serviceId>783015a5-5ad4-43a8-b3be-a45acaba081a</serviceId>
  <serviceName>SIDEMO.ERGL23393T</serviceName>
  <servicePath>OPTIMDIR/SMK_V910</servicePath>
</sem:executionResults></p>

```

## Response codes

The following response codes can be received when the method is complete.

Response code	Description
202	Accepted. The service request was processed and accepted. This response code does not mean that the service request completed successfully.
400	Bad Request. A malformed request was submitted.

## Service schedule methods

Use service schedule methods to view, create, change, and delete service schedules. Each service can have one schedule at a time.

## Add Schedule method

Use the Add Schedule method to add a schedule to run a service.

HTTP method: POST

URI: `/scheduler/connection_name/schedule_id`

Request payload content type: application/xml

Response payload content type: None

Expected response: HTTP/1.1 201 Created

## Parameters

The method URI contains the following parameters.

Name	Description	Required
<i>connection_name</i>	Enter the name that identifies the Optim directory connection that is to contain the schedule.	Y
<i>schedule_id</i>	Enter the schedule ID that identifies the schedule to be added.	Y

## Request payload example

The following example shows a request to schedule a service to run on Thursday, August 1, 2013 at 17:30:00 local time (epoch time 1375378200000 in milliseconds).

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:schedule xmlns:ns2="http://www.ibm.com/optim/xsd/scheduler/9.1.0">
  <id>fbf1cdd5-bdf7-4682-96ad-722672af0001</id>
  <serviceId>783015a5-5ad4-43a8-b3be-a45acaba081a</serviceId>
  <startDate>1375378200000</startDate>
  <endDate>0</endDate>
  <repeatCount>0</repeatCount>
  <repeatInterval>0</repeatInterval>
  <schedulerTaskType>SOA_SERVICE</schedulerTaskType>
  <proxyURL>http://servercomputer:12000/</proxyURL>
  <creatorId>jdoe</creatorId>
</ns2:schedule>
```

The following example shows a request to schedule the service to run on Thursday, August 1, 2013 at 17:30:00 local time. The cron expression sets the schedule to repeat every Thursday afterward at 17:30:00 local time.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:schedule xmlns:ns2="http://www.ibm.com/optim/xsd/scheduler/9.1.0">
  <id>fbf1cdd5-bdf7-4682-96ad-722672af0002</id>
  <serviceId>783015a5-5ad4-43a8-b3be-a45acaba081a</serviceId>
  <startDate>1375378200000</startDate>
  <endDate>0</endDate>
  <cronExpression>0 30 17 ? * THU</cronExpression>
  <schedulerTaskType>SOA_SERVICE</schedulerTaskType>
  <proxyURL>http://servercomputer:12000/</proxyURL>
  <creatorId>jdoe</creatorId>
</ns2:schedule>
```

## Response codes

The following response codes can be received when the method is complete.

Response code	Description
201	Created. The service schedule was created and scheduled.
400	Bad Request. A malformed request was submitted.
409	Conflict. The service ID is already associated with a schedule and the schedule cannot be added, or the schedule ID is already defined in the scheduler.

## Update Schedule method

Use the Update Schedule method to update a schedule to run a service.

HTTP method: PUT

URI: `/scheduler/connection_name/schedule_id`

Request payload content type: application/xml

Response payload content type: None

Expected response: HTTP/1.1 200 OK

## Parameters

The method URI contains the following parameters.

Name	Description	Required
<i>connection_name</i>	Enter the name that identifies the Optim directory connection that contains the schedule.	Y
<i>schedule_id</i>	Enter the schedule ID that identifies the schedule to be updated.	Y

## Request payload example

The following example shows a request to schedule a service to run on Thursday, August 1, 2013 at 17:30:00 local time (epoch time 1375378200000 in milliseconds).

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:schedule xmlns:ns2="http://www.ibm.com/optim/xsd/scheduler/9.1.0">
  <id>fbf1cdd5-bdf7-4682-96ad-722672af0001</id>
  <serviceId>783015a5-5ad4-43a8-b3be-a45acaba081a</serviceId>
  <startDate>1375378200000</startDate>
  <endDate>0</endDate>
  <repeatCount>0</repeatCount>
  <repeatInterval>0</repeatInterval>
  <schedulerTaskType>SOA_SERVICE</schedulerTaskType>
  <proxyURL>http://servercomputer:12000/</proxyURL>
  <creatorId>jdoe</creatorId>
</ns2:schedule>
```



The following example shows a request to schedule the service to run on Thursday, August 1, 2013 at 17:30:00 local time. The cron expression sets the schedule to repeat every Thursday afterward at 17:30:00 local time.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:schedule xmlns:ns2="http://www.ibm.com/optim/xsd/scheduler/9.1.0">
  <id>fbf1cdd5-bdf7-4682-96ad-722672af0002</id>
  <serviceId>783015a5-5ad4-43a8-b3be-a45acaba081a</serviceId>
  <startDate>1375378200000</startDate>
  <endDate>0</endDate>
  <cronExpression>0 30 17 ? * THU</cronExpression>
  <schedulerTaskType>SOA_SERVICE</schedulerTaskType>
  <proxyURL>http://servercomputer:12000</proxyURL>
  <creatorId>jdoe</creatorId>
</ns2:schedule>
```

## Response codes

The following response codes can be received when the method is complete.

Response code	Description
200	OK. The service schedule was updated and scheduled.
400	Bad Request. A malformed request was submitted.
404	Not Found. The schedule ID is not found.
409	Conflict. The service ID is already associated with a different schedule.

## Get All Schedules method

Use the Get All Schedules method to list the schedules for services.

HTTP method: GET

URI: */scheduler/connection\_name/*

Request payload content type: None

Response payload content type: application/xml

Expected response: HTTP/1.1 200 OK

## Parameters

The method URI contains the following parameters.

Name	Description	Required
<i>connection_name</i>	Enter the name that identifies the Optim directory connection that contains the schedules.	Y

## Response payload example

The following example shows a sample XML response for this method.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:schedules xmlns="http://www.w3.org/2005/Atom"
  xmlns:ns2="http://www.ibm.com/optim/xsd/scheduler/9.1.0">
  <link href="http://interface:8080/server/scheduler/OPTIMDIR1/fbf1cdd5-bdf7-4682-96ad-722672af0001"/>
  <link href="http://interface:8080/server/scheduler/OPTIMDIR1/fbf1cdd5-bdf7-4682-96ad-722672af0002"/>
</ns2:schedules>
```

## Response codes

The following response codes can be received when the method is complete.

Response code	Description
200	OK. The schedule list was successfully returned.
400	Bad Request. A malformed request was submitted.
404	Not Found. No schedules were found.

## Get Schedule method

Use the Get Schedule method to get the specified schedule for a service.

HTTP method: GET

URI: */scheduler/connection\_name/schedule\_id*

Request payload content type: None

Response payload content type: application/xml

Expected response: HTTP/1.1 200 OK

## Parameters

The method URI contains the following parameters.

Name	Description	Required
<i>connection_name</i>	Enter the name that identifies the Optim directory connection that contains the schedule.	Y
<i>schedule_id</i>	Enter the schedule ID that identifies the schedule you want to see.	Y

## Response payload example

The following example shows a schedule to run a service on Thursday, August 1, 2013 at 17:30:00 UTC time (epoch time 1375378200000 in milliseconds). The format of the response payload for the Get Schedule method is similar to the request payload for the Add Schedule or Update Schedule methods.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:schedule xmlns:ns2="http://www.ibm.com/optim/xsd/scheduler/9.1.0">
  <id>fbf1cdd5-bdf7-4682-96ad-722672af0001</id>
  <serviceId>783015a5-5ad4-43a8-b3be-a45acaba081a</serviceId>
  <startDate>1375378200000</startDate>
  <endDate>0</endDate>
  <repeatCount>0</repeatCount>
  <repeatInterval>0</repeatInterval>
  <schedulerTaskType>SOA_SERVICE</schedulerTaskType>
  <proxyURL>http://servercomputer:12000/</proxyURL>
</ns2:schedule>
```

The following example shows a request to schedule the service to run on Thursday, August 1, 2013 at 17:30:00 UTC time. The cron expression sets the schedule to repeat every Thursday afterward at 17:30:00 local time.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:schedule xmlns:ns2="http://www.ibm.com/optim/xsd/scheduler/9.1.0">
  <id>fbf1cdd5-bdf7-4682-96ad-722672af0002</id>
  <serviceId>783015a5-5ad4-43a8-b3be-a45acaba081a</serviceId>
  <startDate>1375378200000</startDate>
  <endDate>0</endDate>
  <cronExpression>0 30 17 ? * THU</cronExpression>
  <schedulerTaskType>SOA_SERVICE</schedulerTaskType>
  <proxyURL>http://servercomputer:12000</proxyURL>
</ns2:schedule>
```

## Response codes

The following response codes can be received when the method is complete.

Response code	Description
200	OK. The schedule was successfully returned.
400	Bad Request. A malformed request was submitted.
404	Not Found. The schedule does not exist.

## Delete Schedule method

Use the Delete Schedule method to delete the schedule for a service.

HTTP method: DELETE

URI: */scheduler/connection\_name/schedule\_id*

Request payload content type: None

Response payload content type: None

Expected response: HTTP/1.1 204 No Content

## Parameters

The method URI contains the following parameters.

Name	Description	Required
<i>connection_name</i>	Enter the name that identifies the Optim directory connection that contains the schedule.	Y
<i>schedule_id</i>	Enter the schedule ID whose schedule is to be deleted.	Y

## Response codes

The following response codes can be received when the method is complete.

Response code	Description
204	No Content. The service schedule was removed.
400	Bad Request. A malformed request was submitted.

Response code	Description
404	Not Found. The schedule ID does not exist.

## Service instance methods

Use service instance methods to view information about specific instances when a service is run. When you run a service, a service instance record is created to store information about how the service was run. Service instance information includes the start and end times of the service, the return code, and artifacts that contain additional information.

### Get All Service Instances method

Use the Get All Service Instances method to receive an XML file that contains a list of all service instances. You can then get more detailed information about a service instance by using the Get Service Instance method.

HTTP method: GET

URI: */monitor/connection\_name*

Request payload content type: None

Response payload content type: application/xml

Expected response: HTTP/1.1 200 OK

### Parameters

The method URI contains the following parameters.

Name	Description	Required
<i>connection_name</i>	Enter the name that identifies the Optim directory connection that contains the service instance.	Y

### Response payload example

The following example shows a sample XML response for this method.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:serviceInstances xmlns="http://www.w3.org/2005/Atom"
  xmlns:ns2="http://www.ibm.com/optim/xsd/sim/9.1.0">
  <link href="http://interface:8080/server/monitor/OPTIMDIR1/892ea9ab-af33-40a1-acbb-110dc6cd74f3"/>
  <link href="http://interface:8080/server/monitor/OPTIMDIR1/0f5f5bf3-5f29-4d53-8575-2ff2cbab1c08"/>
  <link href="http://interface:8080/server/monitor/OPTIMDIR1/8630618c-3c08-4b76-b3cf-4dd6fb06fad3"/>
</ns2:serviceInstances>
```

### Response codes

The following response codes can be received when the method is complete.

Response code	Description
200	OK. The service instance list was successfully returned.
400	Bad Request. A malformed request was submitted.
404	Not Found. No service instances exist.

## Get Service Instance method

Use the Get Service Instance method to receive an XML file that contains information about a service instance. The file includes the service ID and the service instance ID, start time, end time, and return code. The file also lists any associated artifacts, which can be retrieved by using the Get Service Instance Artifact method.

HTTP method: GET

URI: `/monitor/connection_name/execution_id`

Request payload content type: None

Response payload content type: `application/xml`

Expected response: HTTP/1.1 200 OK

## Parameters

The method URI contains the following parameters.

Name	Description	Required
<code>connection_name</code>	Enter the name that identifies the Optim directory connection that contains the service instance.	Y
<code>execution_id</code>	Enter the execution ID that identifies the service instance whose information you want to view.	Y

## Response payload example

The following example shows a service instance with execution ID 34770e5c-e282-47bf-9467-160cda2a1e06. Service 783015a5-5ad4-43a8-b3be-a45acaba081a was started on Thursday, August 1, 2013 at 17:30:00 UTC time (epoch time 1375378200230 in milliseconds). The service completed successfully at 17:33:07 UTC time (epoch time 1375378387206 in milliseconds). The service instance contains the artifacts `execution.properties`, `overrides.txt`, `run.log`, and `svc_request.xml`.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<ns2:serviceInstance
  xmlns:ns2="http://www.ibm.com/optim/xsd/sim/9.1.0">
  <id>34770e5c-e282-47bf-9467-160cda2a1e06</id>
  <serviceId>783015a5-5ad4-43a8-b3be-a45acaba081a</serviceId>
  <startTime>1375378200230</startTime>
  <endTime>1375378387206</endTime>
  <returnCode>0</returnCode>
  <artifactList>
    <artifactName>run.log</artifactName>
    <artifactName>svc_request.xml</artifactName>
    <artifactName>execution.properties</artifactName>
    <artifactName>svc_response.xml</artifactName>
    <artifactName>overrides.txt</artifactName>
  </artifactList>
</ns2:serviceInstance>
```

## Response codes

The following response codes can be received when the method is complete.

Response code	Description
200	OK. The service instance information was successfully returned.
400	Bad Request. A malformed request was submitted.
404	Not Found. The execution ID does not exist.

## Get Service Instance Artifact Names method

Use the Get Service Instance Artifact Names method to receive an XML file that contains a list of artifact names for a service instance. For example, you can determine whether a service instance has an override file. You can then use the Get Service Instance Artifact method to receive the content that is contained in each artifact.

HTTP method: GET

URI: `/monitor/connection_name/artifacts/execution_id`

Request payload content type: None

Response payload content type: application/xml

Expected response: HTTP/1.1 200 OK

## Parameters

The method URI contains the following parameters.

Name	Description	Required
<i>connection_name</i>	Enter the name that identifies the Optim directory connection that contains the service instance.	Y
<i>execution_id</i>	Enter the execution ID that identifies the service instance whose artifacts you want to view.	Y

## Response payload example

The following example shows a list of artifacts for a service instance with the execution ID fbflcdd5-bdf7-4682-96ad-722672af0001. The artifacts are named execution.properties, overrides.txt, run.log, svc\_request.xml, and svc\_response.xml.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:serviceInstanceArtifacts xmlns="http://www.w3.org/2005/Atom" xmlns:ns2="http://www.ibm.com/optim/xsd/sim/9.1.0">
  <link href="http://interface:8080/server/monitor/OPTIMDIR1/fbflcdd5-bdf7-4682-96ad-722672af0001/run.log"/>
  <link href="http://interface:8080/server/monitor/OPTIMDIR1/fbflcdd5-bdf7-4682-96ad-722672af0001/svc_request.xml"/>
  <link href="http://interface:8080/server/monitor/OPTIMDIR1/fbflcdd5-bdf7-4682-96ad-722672af0001/execution.properties"/>
  <link href="http://interface:8080/server/monitor/OPTIMDIR1/fbflcdd5-bdf7-4682-96ad-722672af0001/svc_response.xml"/>
  <link href="http://interface:8080/server/monitor/OPTIMDIR1/fbflcdd5-bdf7-4682-96ad-722672af0001/overrides.txt"/>
</ns2:serviceInstanceArtifacts>
```

## Response codes

The following response codes can be received when the method is complete.

Response code	Description
200	OK. The artifact list was successfully returned.
400	Bad Request. A malformed request was submitted.
404	Not Found. The execution ID does not exist.

## Get Service Instance Artifact method

Use the Get Service Instance Artifact method to receive a service instance artifact. For example, you can use this method to request the process report for a service instance.

HTTP method: GET

URI: */monitor/connection\_name/execution\_id/artifact\_name*

Request payload content type: None

Response payload content type: application/xml

Expected response: HTTP/1.1 200 OK

## Parameters

The method URI contains the following parameters.

Name	Description	Required
<i>connection_name</i>	Enter the name that identifies the Optim directory connection that contains the service instance.	Y
<i>execution_id</i>	Enter the execution ID that identifies the service instance whose artifacts you want to view.	Y
<i>artifact_name</i>	Enter the name of the artifact whose content you want to view.	Y

## Response payload example

The following example shows possible contents of an artifact. Artifacts are text-based files that are either in XML format or text format. Use artifacts to troubleshoot issues with a service.

```
/OUTPUT PSTDIR=OPTDIRORA TYPE=Extract REQUEST=TESTDATA.ERCUST11K STOP=None ERRORLEVEL=0
Extract Process Report
```

Request Name	TESTDATA.ERCUST11K
Server Name	(Local)
Extract File	C:\IBM\InfoSphere\Optim\data\ERCUST11K.xf
Access Definition	TESTDATA.CUST11K
File Attachments	Processed
Client User ID	optadmin
Server User ID	optadmin
Teradata Character Set	WE8MSWIN1252
Time Started	5/16/2013 13:03:52

Time Finished 5/16/2013 13:03:54  
Elapsed Time 00:00:02  
Extract File Data Byte Count 0.001 MB  
Process Status no errors, no warnings

## Response codes

The following response codes can be received when the method is complete.

Response code	Description
200	OK. The artifact was successfully returned.
400	Bad Request. A malformed request was submitted.
404	Not Found. The execution ID does not exist, or the artifact name does not exist for the execution ID.

## Delete Service Instance method

Use the Delete Service Instance method to purge or delete a service instance.

HTTP method: DELETE

URI: */monitor/connection\_name/execution\_id*

Request payload content type: None

Response payload content type: application/xml

Expected response: HTTP/1.1 204 No Content

## Parameters

The method URI contains the following parameters.

Name	Description	Required
<i>connection_name</i>	Enter the name that identifies the Optim directory connection that contains the service instance.	Y
<i>execution_id</i>	Enter the execution ID that identifies the service instance to be deleted.	Y

## Response codes

The following response codes can be received when the method is complete.

Response code	Description
204	No Content. The service instance was successfully purged.
400	Bad Request. A malformed request was submitted.
404	Not Found. The execution ID does not exist.

## Component methods

Use component methods to view and interact directly with registered components. For example, you can use a component method to get a list of registered servers.



## Get Server List method

Use the Get Server List method to receive an XML file that contains a list of registered servers.

HTTP method: GET

URI: /runtime

Request payload content type: None

Response payload content type: application/xml

Expected response: HTTP/1.1 200 OK

## Response payload example

The following example shows a list of three servers with host names server1, server2, and server3.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:servers xmlns:ns2="http://www.ibm.com/optim/xsd/rr/9.1.0">
  <rsiURL>http://server1:12000/</rsiURL>
  <rsiURL>http://server2:12000/</rsiURL>
  <rsiURL>http://server3:12000/</rsiURL>
</ns2:servers>
```

## Response codes

The following response codes can be received when the method is complete.

Response code	Description
200	OK. The server list was successfully returned.
400	Bad Request. A malformed request was submitted.
404	Not Found. No servers were found.

## Work order methods

A *work order* is a request to create or change a service or service set for a specific purpose. Use work order methods to create and manage work orders.

### Create Work Order method

Use the Create Work Order method to create a work order. A work order number that identifies the work order is automatically assigned to the work order.

HTTP method: POST

URI: /workorder/connection\_name/transition\_id

Request payload content type: application/xml

Response payload content type: None (a link to the newly created work order is returned in the Location field of the response header; for example, http://interface:8080/server/workorder/OPTIMDIR1/10047)

Expected response: HTTP/1.1 201 Created

## Parameters

The method URI contains the following parameters.

Name	Description	Required
<i>connection_name</i>	Enter the name that identifies the Optim directory connection that is to contain the work order.	Y
<i>transition_id</i>	<p>Enter the ID that identifies the state transition of the work order to be created. A <i>state transition</i> is a valid change of state for a work order. The manager and the service interface allow only state transitions that are defined in the workflow. To get the valid state transitions for your workflow:</p> <ol style="list-style-type: none"><li>1. Use the Get Active States method to get a list of all active states for the current workflow, find the state with a name of NEW, and obtain the state ID for that state.</li><li>2. Use the Get Active Reachable Transitions method to get a list of state transitions for the NEW state. There is only one state transition possible for the NEW state. Use this state transition when you create the work order.</li></ol>	Y

## Request payload example

The following example shows a request by user TEST1 to create a subset of customer information and to mask the personally identifiable information within the subset.

```
<?xml version="1.0" encoding="UTF-8"?>
<workorder:workorder xmlns:atom="http://www.w3.org/2005/Atom"
  xmlns:workorder="http://www.ibm.com/optim/xsd/workorder/9.1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.ibm.com/optim/xsd/workorder/9.1.0 resource.xsd ">
  <wfId>1</wfId>
  <stateId>100</stateId>
  <description>Create a subset of customers with personally identifiable information masked</description>
  <priority>Within 8 Hours</priority>
  <comments>Mask names, addresses, phone numbers, email addresses</comments>
  <createdBy>TEST1</createdBy>
</workorder:workorder>
```

## Response codes

The following response codes can be received when the method is complete.

Response code	Description
201	Created. The work order was successfully created.
400	Bad Request. A malformed request was submitted.

## Update Work Order method

Use the Update Work Order method to update a work order. If a state transition is specified, the service interface changes the state of the work order based upon the specified state transition. If email addresses are associated with users in the manager, the service interface also notifies users by email.

HTTP method: PUT

URI: */workorder/connection\_name/work\_order\_id/transition\_id*

Request payload content type: application/xml

Response payload content type: None

Expected response: HTTP/1.1 200 OK

## Parameters

The method URI contains the following parameters.

Name	Description	Required
<i>connection_name</i>	Enter the name that identifies the Optim directory connection that contains the work order.	Y
<i>work_order_id</i>	Enter the ID that identifies the work order to be updated.	Y
<i>transition_id</i>	Enter the ID that identifies the state transition of the work order to be updated. A <i>state transition</i> is a valid change of state for a work order. The manager and the service interface allow only state transitions that are defined in the workflow. Use the Get Active Reachable Transitions method to get a list of state transitions for the current state of the work order.	N

## Request payload example

The following example shows a request to update a work order. The request is made by user TEST1 to create a subset of customer information and to mask the personally identifiable information within the subset.

```
<?xml version="1.0" encoding="UTF-8"?>
<workorder:workorder xmlns:atom="http://www.w3.org/2005/Atom"
  xmlns:workorder="http://www.ibm.com/optim/xsd/workorder/9.1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.ibm.com/optim/xsd/workorder/9.1.0 resource.xsd">
  <description>Create a subset of customers with personally identifiable information masked</description>
  <priority>Within 8 hours</priority>
  <comments>Mask names, addresses, phone numbers, email addresses</comments>
  <createdBy>TEST1</createdBy>
</workorder:workorder>
```

## Response codes

The following response codes can be received when the method is complete.

Response code	Description
200	OK. The work order was updated.
400	Bad Request. A malformed request was submitted.
404	Not Found. The work order ID does not exist.

## Delete Work Order method

Use the Delete Work Order method to delete a work order.

HTTP method: DELETE

URI: */workorder/connection\_name/work\_order\_id*

Request payload content type: None

Response payload content type: None

Expected response: HTTP/1.1 204 No Content

## Parameters

The method URI contains the following parameters.

Name	Description	Required
<i>connection_name</i>	Enter the name that identifies the Optim directory connection that contains the work order.	Y
<i>work_order_id</i>	Enter the ID that identifies the work order to be deleted.	Y

## Response codes

The following response codes can be received when the method is complete.

Response code	Description
204	No Content. The work order was successfully deleted.
400	Bad Request. A malformed request was submitted.
404	Not Found. The work order ID does not exist.

## Get Work Orders method

Use the Get Work Orders method to receive an XML file that contains links to the work orders.

HTTP method: GET

URI: */workorder/connection\_name*

Request payload content type: None

Response payload content type: application/xml

Expected response: HTTP/1.1 200 OK

## Parameters

The method URI contains the following parameters.

Name	Description	Required
<i>connection_name</i>	Enter the name that identifies the Optim directory connection that contains the work orders.	Y

## Response payload example

The following example shows a response that contains a list of three work orders.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:workorders xmlns="http://www.w3.org/2005/Atom"
  xmlns:ns2="http://www.ibm.com/optim/xsd/workorder/9.1.0">
  <link href="http://interface:8080/server/workorder/OPTIMDIR1/10008"/>
  <link href="http://interface:8080/server/workorder/OPTIMDIR1/10011"/>
  <link href="http://interface:8080/server/workorder/OPTIMDIR1/10015"/>
</ns2:workorders>
```

## Response codes

The following response codes can be received when the method is complete.

Response code	Description
200	OK. The work orders links were successfully returned.
400	Bad Request. A malformed request was submitted.
404	Not Found. No work orders exist.

## Get Work Order method

Use the Get Work Order method to receive an XML file that contains the properties and state of a specified work order. If the work order is associated with a service or a service set, the XML file also specifies the service or service set.

HTTP method: GET

URI: /workorder/*connection\_name*/*work\_order\_id*

Request payload content type: None

Response payload content type: application/xml

Expected response: HTTP/1.1 200 OK

## Parameters

The method URI contains the following parameters.

Name	Description	Required
<i>connection_name</i>	Enter the name that identifies the Optim directory connection that contains the work order.	Y

Name	Description	Required
<i>work_order_id</i>	Enter the ID that identifies the work order that you want to receive.	Y

## Response payload example

The following example shows work order 10047, which was created on Thursday, August 1, 2013 at 17:30:00 local time (epoch time 1375378200000 in milliseconds).

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<ns2:workorder xmlns:ns2="http://www.ibm.com/optim/xsd/workorder/9.1.0">
  <id>10047</id>
  <wfId>1</wfId>
  <stateId>300</stateId>
  <description>Create a subset of customers with personally identifiable information masked</description>
  <priority>Within 8 hours</priority>
  <comments>Mask names, addresses, phone numbers, email addresses</comments>
  <createdBy>TEST1</createdBy>
  <createdDate>1375378200000</createdDate>
</ns2:workorder>
```

The following example shows work order 10047 after it is associated with a service with service ID fbflcdd5-bdf7-4682-96ad-722672af0001.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<ns2:workorder xmlns:ns2="http://www.ibm.com/optim/xsd/workorder/9.1.0">
  <id>10047</id>
  <wfId>1</wfId>
  <stateId>500</stateId>
  <description>Create a subset of customers with personally identifiable information masked</description>
  <priority>Within 8 hours</priority>
  <comments>Mask names, addresses, phone numbers, email addresses</comments>
  <createdBy>TEST1</createdBy>
  <createdDate>1375378200000</createdDate>
  <serviceId>fbflcdd5-bdf7-4682-96ad-722672af0001</serviceId>
</ns2:workorder>
```

The following example shows work order 10047 after its association is changed to version 1.0.0 of service set ExtractSet1.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<ns2:workorder xmlns:ns2="http://www.ibm.com/optim/xsd/workorder/9.1.0">
  <id>10047</id>
  <wfId>1</wfId>
  <stateId>500</stateId>
  <description>Create a subset of customers with personally identifiable information masked</description>
  <priority>Within 8 hours</priority>
  <comments>Mask names, addresses, phone numbers, email addresses</comments>
  <createdBy>TEST1</createdBy>
  <createdDate>1375378200000</createdDate>
  <sSetName>ExtractSet1</sSetName>
  <sSetVersion>1.0.0</sSetVersion>
</ns2:workorder>
```

## Response codes

The following response codes can be received when the method is complete.

Response code	Description
200	OK. The work order was successfully returned.
400	Bad Request. A malformed request was submitted.

Response code	Description
404	Not Found. The work order ID was not found.

## Get Active States method

Use the Get Active States method to receive an XML file that contains the possible work order states for a specified workflow.

HTTP method: GET

URI: `/workorder/connection_name/states/workflow_id`

Request payload content type: None

Response payload content type: application/xml

Expected response: HTTP/1.1 200 OK

## Parameters

The method URI contains the following parameters.

Name	Description	Required
<i>connection_name</i>	Enter the name that identifies the Optim directory connection that contains the work order.	Y
<i>workflow_id</i>	Enter the ID that identifies the workflow whose active states you want to receive. Enter 1 for the default workflow that is used by the manager and the service interface.	Y

## Response payload example

The following example shows the response for the default workflow (1), which contains the possible work order states for the workflow.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<ns2:states xmlns:ns2="http://www.ibm.com/optim/xsd/workorder/9.1.0">
  <ns2:state>
    <stateId>100</stateId>
    <wfId>1</wfId>
    <stateName>NEW</stateName>
  </ns2:state>
  <ns2:state>
    <stateId>200</stateId>
    <wfId>1</wfId>
    <stateName>REQUESTED</stateName>
  </ns2:state>
  <ns2:state>
    <stateId>250</stateId>
    <wfId>1</wfId>
    <stateName>DENIED</stateName>
  </ns2:state>
  <ns2:state>
    <stateId>300</stateId>
    <wfId>1</wfId>
    <stateName>SUBMITTED</stateName>
  </ns2:state>
</ns2:states>
```

```

<ns2:state>
  <stateId>301</stateId>
  <wfId>1</wfId>
  <stateName>INCOMPLETE</stateName>
</ns2:state>
<ns2:state>
  <stateId>350</stateId>
  <wfId>1</wfId>
  <stateName>REJECTED</stateName>
</ns2:state>
<ns2:state>
  <stateId>400</stateId>
  <wfId>1</wfId>
  <stateName>READY</stateName>
</ns2:state>
<ns2:state>
  <stateId>450</stateId>
  <wfId>1</wfId>
  <stateName>RETURNED</stateName>
</ns2:state>
<ns2:state>
  <stateId>999</stateId>
  <wfId>1</wfId>
  <stateName>TERMINATED</stateName>
</ns2:state>
</ns2:states>

```

## Response codes

The following response codes can be received when the method is complete.

Response code	Description
200	OK. The active state list was successfully returned.
400	Bad Request. A malformed request was submitted.
404	Not Found. The workflow ID is valid but no active states exist for the workflow ID. This is a configuration issue.

## Get Active Reachable Transitions method

Use the Get Active Reachable Transitions method to receive an XML file that contains the state transitions for a specified work order state within a specified workflow. The XML file contains only the state transitions that are directly reachable from the specified work order state.

HTTP method: GET

URI: */workorder/connection\_name/transitions/workflow\_id/state\_id*

Request payload content type: None

Response payload content type: application/xml

Expected response: HTTP/1.1 200 OK



## Parameters

The method URI contains the following parameters.

Name	Description	Required
<i>connection_name</i>	Enter the name that identifies the Optim directory connection that contains the work order.	Y
<i>workflow_id</i>	Enter the ID that identifies the workflow whose reachable transitions you want to receive. Enter 1 for the default workflow that is used by the manager.	Y
<i>state_id</i>	Enter the ID that identifies the state whose reachable transitions you want to receive.	Y

## Response payload example

The following example shows a sample XML response for the default workflow and state 200 (REQUESTED).

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<ns2:transitions xmlns:ns2="http://www.ibm.com/optim/xsd/workorder/9.1.0">
  <ns2:transition>
    <id>2</id>
    <wfId>1</wfId>
    <oldStateId>200</oldStateId>
    <transitionName>DENY</transitionName>
    <newStateId>250</newStateId>
    <owner>2</owner>
  </ns2:transition>
  <ns2:transition>
    <id>4</id>
    <wfId>1</wfId>
    <oldStateId>200</oldStateId>
    <transitionName>TERMINATE</transitionName>
    <newStateId>999</newStateId>
    <owner>1</owner>
  </ns2:transition>
  <ns2:transition>
    <id>5</id>
    <wfId>1</wfId>
    <oldStateId>200</oldStateId>
    <transitionName>TERMINATE</transitionName>
    <newStateId>999</newStateId>
    <owner>2</owner>
  </ns2:transition>
  <ns2:transition>
    <id>6</id>
    <wfId>1</wfId>
    <oldStateId>200</oldStateId>
    <transitionName>TERMINATE</transitionName>
    <newStateId>999</newStateId>
    <owner>3</owner>
  </ns2:transition>
  <ns2:transition>
    <id>10</id>
    <wfId>1</wfId>
    <oldStateId>200</oldStateId>
    <transitionName>APPROVE</transitionName>
```

```

        <newStateId>300</newStateId>
        <owner>2</owner>
    </ns2:transition>
</ns2:transitions>

```

## Response codes

The following response codes can be received when the method is complete.

Response code	Description
200	OK. The active reachable transitions list was successfully returned.
400	Bad Request. A malformed request was submitted.
404	Not Found. No active reachable transitions exist for the workflow ID and state ID.

## Run Work Order method

Use the Run Work Order method to run the service that is associated with a work order. The service can be run without any changes. Alternatively, you can run the service with input values that are different from the values that are saved with the service.

HTTP method: POST

URI: `/workorder/connection_name/execute/work_order_id`

Request payload content type: application/xml

Response payload content type: None (a link to the newly created service instance is returned in the Location field of the response header; for example, `http://interface:8080/server/monitor/OPTIMDIR1/fbf1cdd5-bdf7-4682-96ad-722672af0001`)

Expected response: HTTP/1.1 202 Accepted

## Parameters

The method URI contains the following parameters.

Name	Description	Required
<i>connection_name</i>	Enter the name that identifies the Optim directory connection that contains the work order.	Y
<i>work_order_id</i>	Enter the ID that identifies the work order whose service you want to run.	Y

## Request payload example

The following example shows a request to run the service that is associated with work order 10047.

```

<?xml version="1.0" encoding="UTF-8"?>
<workorder:workOrderExecution xmlns:atom="http://www.w3.org/2005/Atom"
  xmlns:workorder="http://www.ibm.com/optim/xsd/workorder/9.1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.ibm.com/optim/xsd/workorder/9.1.0 resource.xsd">

```

```

    <workOrderId>10047</workOrderId>
    <proxyURL>http://servercomputer:12000/</proxyURL>
    <executedBy>TEST1</executedBy>
</workorder:workOrderExecution>

```

The following example shows a request to run the service that is associated with work order 10047, and to use the input values in an override file at C:\override\override.txt on the server computer.

```

<?xml version="1.0" encoding="UTF-8"?>
<workorder:workOrderExecution xmlns:atom="http://www.w3.org/2005/Atom"
  xmlns:workorder="http://www.ibm.com/optim/xsd/workorder/9.1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.ibm.com/optim/xsd/workorder/9.1.0 resource.xsd ">
  <workOrderId>10047</workOrderId>
  <proxyURL>http://servercomputer:12000/</proxyURL>
  <overrideFilePath>C:\override\override.txt</overrideFilePath>
  <executedBy>TEST1</executedBy>
</workorder:workOrderExecution>

```

The following example shows a request to run the service that is associated with work order 10047 with 2 changed input values (override values).

```

<?xml version="1.0" encoding="UTF-8"?>
<workorder:workOrderExecution xmlns:atom="http://www.w3.org/2005/Atom"
  xmlns:workorder="http://www.ibm.com/optim/xsd/workorder/9.1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.ibm.com/optim/xsd/workorder/9.1.0 resource.xsd ">
  <workOrderId>10047</workOrderId>
  <proxyURL>http://servercomputer:12000/</proxyURL>
  <executedBy>TEST1</executedBy>
  <overrides>
    <workorder:override>
      <id>override-id-000001</id>
      <value>Value One</value>
    </workorder:override>
    <workorder:override>
      <id>override-id-000002</id>
      <value>Value Two</value>
    </workorder:override>
  </overrides>
</workorder:workOrderExecution>

```

## Response codes

The following response codes can be received when the method is complete.

Response code	Description
202	Accepted. The service that is associated with the work order was processed and accepted. This response code does not mean that the service request completed successfully.
400	Bad Request. A malformed request was submitted.

## Command-line utility for service interface

By default, the service interface is installed with a command-line utility. The command-line utility is an example of an application that uses the service interface to run and manage services.

## Command-line utility location

The command-line utility uses the **optimcmd** tool. The location of the **optimcmd** tool depends on the operating system. *shared\_installation\_directory* is the installation directory that you specified for the service interface.

- Microsoft Windows computers: *shared\_installation\_directory\tools\optimcmd\optimcmd.bat*.
- Linux or UNIX computers: *shared\_installation\_directory/tools/optimcmd/optimcmd.sh*.

## Configuration file location

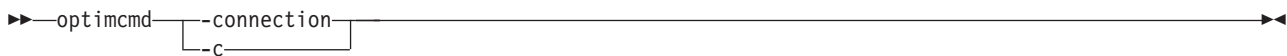
The *optimcmd.properties* file contains the URLs that the command-line utility uses to locate the service interface, the server, and the connection manager. Before you use the command-line utility, confirm that the *optimcmd.properties* file contains the URLs that you want the command-line utility to use. The *optimcmd.properties* file is in the *shared\_installation\_directory/tools/optimcmd/* directory, where *shared\_installation\_directory* is the installation directory that you specified for the service interface.

## Get connections (optimcmd -connection)

Use the Get connections command to get a list of Optim directory connections. Use the connection names to identify Optim directories in other commands.

### Command structure

The following diagram indicates the structure of the command and its parameters.



### Input

The following information can be entered with the command.

**optimcmd**

The optimcmd tool.

**-connection or -c**

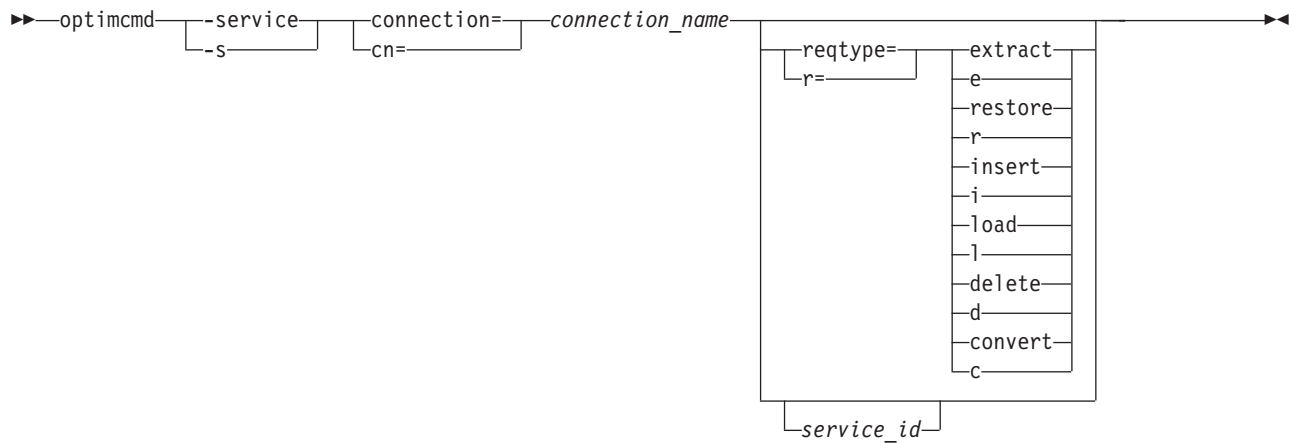
The Get connections command.

## Get service (optimcmd -service)

Use the Get service command to get a list of services or to get detailed information about a specific service. A list of services contains the service ID, the service name, the path of the service, and the service type.

### Command structure

The following diagram indicates the structure of the command and its parameters.



## Input

The following information can be entered with the command.

### **optimcmd**

The optimcmd tool.

### **-service or -s**

The Get service command.

### **connection=connection\_name or cn=connection\_name**

The connection name of the Optim directory in which the services are located.

### **reqtype= or r=**

The service type whose services you want to list. Use any of the following values:

- extract or e: Extract
- restore or r: Restore
- insert or i: Insert
- load or l: Load
- delete or d: Delete
- convert or c: Convert

### **service\_id**

The ID for the service that you want to view.

## Output

The output depends on whether you specify a service ID.

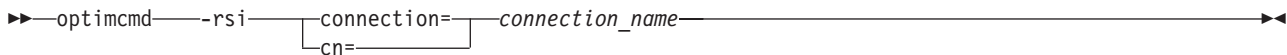
- If you specify a service ID, the command returns detailed information about the service.
- If you do not specify a service ID, the command returns a list of services. Each line contains information for a service (the service ID, the service name, the path of the service, and the service type). You can optionally limit the services in this list to services with a specified service type.

## Get servers (optimcmd -rsi)

Use the Get servers command to get a list of registered servers. The output of this command contains the full URL of the server with the host name or IP address and the port (for example, `http://server:12000/`).

## Command structure

The following diagram indicates the structure of the command and its parameters.



## Input

The following information can be entered with the command.

**optimcmd**

The `optimcmd` tool.

**-rsi**

The Get servers command.

**connection=connection\_name or cn=connection\_name**

The connection name of the Optim directory in which the servers are registered.

## Output

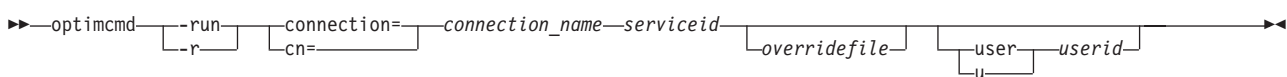
The command returns a list of registered servers.

## Run service (optimcmd -run)

Use the Run service command to run a service by using the server to which the service is assigned. You can optionally specify an override file that contains input values for the service. If you specify an override file, the service is run by using the input values that are specified in the override file.

## Command structure

The following diagram indicates the structure of the command and its parameters.



## Input

The following information can be entered with the command.

**optimcmd**

The `optimcmd` tool.

**-run or -r**

The Run service command.

**connection=connection\_name or cn=connection\_name**

The connection name of the Optim directory in which the service is located.

**serviceid**

The service ID for the service that you want to run.

**overridefile**

The name and path of the override file that contains the input values that you want to use to run the service. The override file that is used by the Run service command is a text file. Each line in the override file contains a parameter or override name, an equals sign, and the value to use for the parameter or override.

**-user or -u**

The user parameter for the Run service command.

*userid* The user ID to use to run the service.

## Output

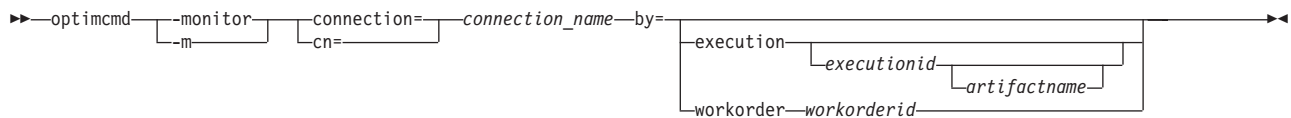
When successful, the command returns the execution ID of the service instance that is started when you run the service.

## Monitor service instances (optimcmd -monitor)

Use the Monitor service instances command to view a list of service instances or to view detailed information about a specific service instance. You can get information about all service instances, a specific service instance, or service instances that are associated with a specific work order.

### Command structure

The following diagram indicates the structure of the command and its parameters.



## Input

The following information can be entered with the command.

**optimcmd**

The optimcmd tool.

**-monitor or -m**

The Monitor service command.

**connection=connection\_name or cn=connection\_name**

The connection name of the Optim directory in which the service instances are located.

**by=execution or by=workorder**

Indicate whether to view service instances for a specific work order (`by=workorder`) or without regard to work order (`by=execution`).

*executionid*

The execution ID for the service instance that you want to view.

*artifactname*

The name of the artifact that you want to view. Artifacts are XML or text files that contain more information about the service instance.

*workorderid*

The work order ID that is associated with the service instances that you want to view.

## Output

The output depends on whether you specify an execution ID, work order ID, or artifact name.

- If you do not specify an execution ID or a work order ID, the command returns a list of service instances. Each line contains information for a service instance (the execution ID, the service ID, and the return code).

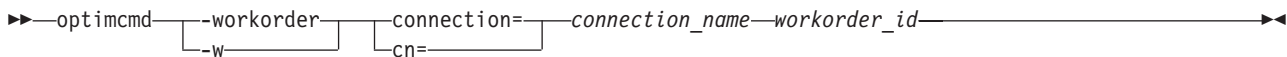
- If you specify an execution ID without an artifact name, the command returns detailed information about the service instance. The information includes the execution ID, the service ID, the start and end date, and the return code. The information also contains a list of artifacts that contain detailed information about the service instance.
- If you specify an execution ID with an artifact name, the command returns the contents of the artifact. The artifact can be text output or XML code.
- If you specify a work order ID, the command returns detailed information about the service instances that are associated with the work order. The information includes the work order ID, the execution ID, the service ID, the start and end date, and the return code. The information also contains a list of artifacts that contain detailed information about the service instance.

## Display work orders (optimcmd -workorder)

Use the Display work orders command to display work orders. You can display a list of all work orders, or you can display detailed information about a specific work order.

### Command structure

The following diagram indicates the structure of the command and its parameters.



### Input

The following information can be entered with the command.

#### **optimcmd**

The optimcmd tool.

#### **-workorder or -w**

The Work order command.

#### **connection=connection\_name or cn=connection\_name**

The connection name of the Optim directory in which the work orders are located.

#### **workorder\_id**

The work order ID for the work order that you want to view. If you do not specify a work order ID, a list of all work orders is returned.

### Example output

The following example output shows a list of all work orders.

ID	WorkflowID	StateID	State	Priority	Description
10001	1	100	New	Within 8 hours	Create a subset of customers
10002	1	300	Submitted	Within 16 hours	Describing free text
10003	1	400	Ready	Within 8 hours	Ready for you

The following example output shows work order 10001.

```

ID=10001
WFID=1
STATEID=100
Description=Create a subset of customers
Priority=Within 8 Hours
Comments=Mask names, addresses, phone numbers, email addresses
CreatedBy=TEST1
CreatedDate=2014-12-12
  
```

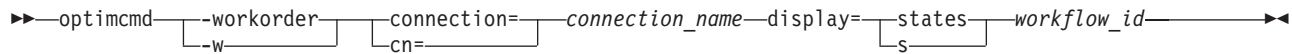


## Display work order states (optimcmd -workorder display=states)

Use the Display work orders states command to display a list of active work order states for a specified workflow.

### Command structure

The following diagram indicates the structure of the command and its parameters.



### Input

The following information can be entered with the command.

#### **optimcmd**

The optimcmd tool.

#### **-workorder or -w**

The Work order command.

#### **connection=connection\_name or cn=connection\_name**

The connection name of the Optim directory in which the workflow is located.

#### **display=states or display=s**

Indicate that the command is to display a list of active work order states for a specified workflow.

#### **workflow\_id**

The ID that identifies the workflow whose active states you want to view. Enter 1 for the default workflow that is used by the manager and the service interface.

### Example output

The following example output shows the states for workflow ID 1.

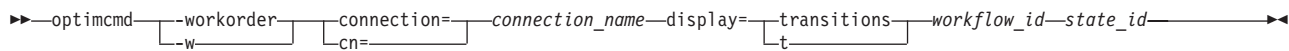
WFID	State ID	State Name
1	100	NEW
1	200	REQUESTED
1	250	DENIED
1	300	SUBMITTED
1	301	INCOMPLETE
1	350	REJECTED
1	400	READY
1	450	RETURNED
1	999	TERMINATED

## Display work order transitions (optimcmd -workorder display=transitions)

Use the Display work order transitions command to display the state transitions for a specified work order state within a specified workflow. The output contains only the state transitions that are directly reachable from the specified work order state.

### Command structure

The following diagram indicates the structure of the command and its parameters.



## Input

The following information can be entered with the command.

**optimcmd**

The optimcmd tool.

**-workorder or -w**

The Work order command.

**connection=connection\_name or cn=connection\_name**

The connection name of the Optim directory in which the workflow is located.

**display=transitions or display=t**

Indicate that the command is to display a list of transitions for a specified workflow.

*workflow\_id*

The ID that identifies the workflow whose transitions you want to view. Enter 1 for the default workflow that is used by the manager and the service interface.

*state\_id*

The ID that identifies the state whose transitions you want to view.

## Example output

The following example output shows the transitions for workflow ID 1 and state ID 200 (REQUESTED).

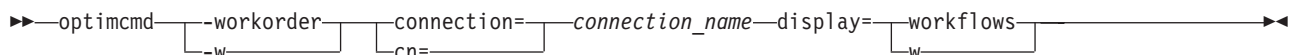
TransitionID	TransName	WFID	OldStateName	State ID	StateName
2	DENY	1	REQUESTED	250	DENIED
10	APPROVE	1	REQUESTED	300	SUBMITTED
6	TERMINATE	1	REQUESTED	999	TERMINATED

## Display workflows (optimcmd -workorder display=workflows)

Use the Display workflows command to display a list of active work order states for all workflows.

## Command structure

The following diagram indicates the structure of the command and its parameters.



## Input

The following information can be entered with the command.

**optimcmd**

The optimcmd tool.

**-workorder or -w**

The Work order command.

**connection=connection\_name or cn=connection\_name**

The connection name of the Optim directory in which the workflows are located.

**display=workflows or display=w**

Indicate that the command is to display a list of work order workflows.

## Example output

The following example output shows the states for workflow IDs 1 and 2.

WFID	State ID	State Name
1	100	NEW
1	200	REQUESTED
1	250	DENIED
1	300	SUBMITTED
1	301	INCOMPLETE
1	350	REJECTED
1	400	READY
1	450	RETURNED
1	999	TERMINATED

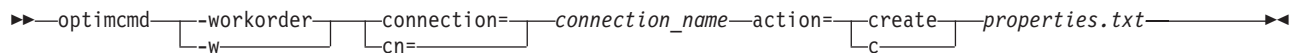
WFID	State ID	State Name
2	100	NEW
2	300	SUBMITTED
2	301	INCOMPLETE
2	350	REJECTED
2	400	READY
2	450	RETURNED
2	999	TERMINATED

## Create work order (optimcmd -workorder action=create)

Use the Create work order command to create a work order. The work order is created based on the properties in a specified text file.

### Command structure

The following diagram indicates the structure of the command and its parameters.



## Input

The following information can be entered with the command.

### **optimcmd**

The `optimcmd` tool.

### **-workorder or -w**

The Work order command.

### **connection=connection\_name or cn=connection\_name**

The connection name of the Optim directory in which the work order is to be created.

### **action=create or action=c**

Indicates that the command is to create a work order.

### **properties.txt**

The name of the text file that specifies the initial properties of the work order. The text file must contain the following properties. Specify each property on a separate line of the text file.

#### **wfID=workflow\_id**

The ID of the workflow that is to be used by the work order.

#### **stated=initial\_state**

The ID of the initial state for the work order. Use the Display work order states command to view a list of all active states for the workflow, find the state with a name of NEW, and use the ID for that state here.

**description=description**

A short description of what the work order is to do. For example, Create a subset of customers.

**priority=priority\_id**

The time in which the work order is to be completed. Valid priorities are:

- Immediately
- Within 8 Hours
- Within 16 Hours
- Within 24 Hours
- Within 48 Hours

**comments=comments**

Additional information to add to the work order. This information is available to others who work with the work order.

**createdBy=creator\_id**

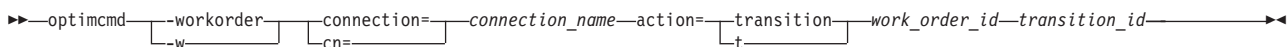
The user ID of the user who is creating the work order.

## Change work order state (optimcmd -workorder action=transition)

Use the Change work order state command to change the state of a work order. The state of a work order indicates who is currently responsible for processing the work order and what is to be done with the work order. To change the work order state, you must specify a transition ID, which indicates the reason why the work order state is being changed.

### Command structure

The following diagram indicates the structure of the command and its parameters.



### Input

The following information can be entered with the command.

**optimcmd**

The optimcmd tool.

**-workorder or -w**

The Work order command.

**connection=connection\_name or cn=connection\_name**

The connection name of the Optim directory in which the work order is located.

**action=transition or action=t**

Indicates that the command is to change the state of a work order.

**work\_order\_id**

The ID of the work order whose state is to be changed.

**transition\_id**

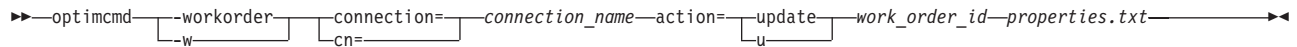
The ID of the transition that indicates why the work order state is being changed.

## Change work order (optimcmd -workorder action=update)

Use the Change work order command to change the properties in a work order other than the work order state. The changes to the work order are based on the properties in a specified text file.

## Command structure

The following diagram indicates the structure of the command and its parameters.



## Input

The following information can be entered with the command.

### **optimcmd**

The optimcmd tool.

### **-workorder or -w**

The Work order command.

### **connection=connection\_name or cn=connection\_name**

The connection name of the Optim directory in which the work order is located.

### **action=update or action=u**

Indicates that the command is to change the properties of a work order.

### **work\_order\_id**

The ID of the work order whose state is to be changed.

### **properties.txt**

The name of the text file that specifies the properties to change in the work order. Specify each property on a separate line of the text file.

### **description=description**

A short description of what the work order is to do. For example, Create a subset of customers.

### **priority=priority\_id**

The time in which the work order is to be completed. Valid priorities are:

- Immediately
- Within 8 Hours
- Within 16 Hours
- Within 24 Hours
- Within 48 Hours

### **comments=comments**

Additional information to add to the work order. This information is available to others who work with the work order.

### **createdBy=creator\_id**

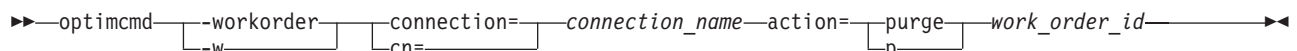
The user ID of the user who created the work order.

## Delete work order (optimcmd -workorder action=purge)

Use the Delete work order command to delete a work order.

## Command structure

The following diagram indicates the structure of the command and its parameters.



## Input

The following information can be entered with the command.

**optimcmd**

The optimcmd tool.

**-workorder or -w**

The Work order command.

**connection=connection\_name or cn=connection\_name**

The connection name of the Optim directory in which the work order is located.

**action=purge or action=p**

Indicates that the command is to delete a work order.

*work\_order\_id*

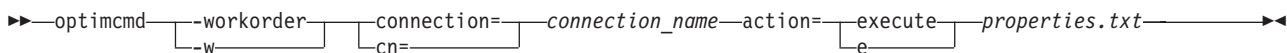
The ID of the work order that is to be deleted.

## Run work order (optimcmd -workorder action=execute)

Use the Run work order command to run the service that is associated with a work order.

### Command structure

The following diagram indicates the structure of the command and its parameters.



## Input

The following information can be entered with the command.

**optimcmd**

The optimcmd tool.

**-workorder or -w**

The Work order command.

**connection=connection\_name or cn=connection\_name**

The connection name of the Optim directory in which the work order is located.

**action=execute or action=e**

Indicates that the command is to run a work order.

*properties.txt*

The name of the text file that specifies the initial properties of the work order. The text file must contain the following properties. Specify each property on a separate line of the text file.

**wfID=workflow\_id**

The ID of the workflow that is to be used by the work order.

**stated=initial\_state**

The ID of the initial state for the work order. Use the Display work order states command to view a list of all active states for the workflow, find the state with a name of NEW, and use the ID for that state here.

**description=description**

A short description of what the work order is to do. For example, Create a subset of customers with personally identifiable information masked.

**priority**=*priority\_id*

The time in which the work order is to be completed. Valid priorities are:

- Immediately
- Within 8 Hours
- Within 16 Hours
- Within 24 Hours
- Within 48 Hours

**comments**=*comments*

Additional information to add to the work order. This information is available to others who work with the work order.

**createdBy**=*creator\_id*

The user ID of the user who is creating the work order.





---

## Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan Ltd.  
1623-14, Shimotsuruma, Yamato-shi  
Kanagawa 242-8502 Japan

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation  
Software Interoperability Coordinator  
Director of Engineering, Information Management (Office 16)  
111 Campus Drive  
Princeton, NJ 08540  
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

#### COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. \_enter the year or years\_. All rights reserved.

If you are viewing this information in softcopy, the photographs and color illustrations may not appear.

---

## Trademarks

IBM, the IBM logo, and [ibm.com](http://ibm.com)<sup>®</sup> are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.



---

# Index

## A

Add Schedule  
    method 7  
    service schedule methods 7  
Add Schedule method 7  
artifact 31

## C

Change work order command 37  
Change work order state command 36  
command-line utility 28  
    configuring 28  
    location 28  
component methods 17  
configuring  
    command-line utility 28  
Create Work Order  
    method 17  
    service instance methods 17  
Create work order command 35, 38  
Create Work Order method 17

## D

Delete Schedule  
    method 11  
    service schedule methods 11  
Delete Schedule method 11  
Delete Service Instance  
    method 16  
    service instance methods 16  
Delete Service Instance method 16  
Delete Work Order  
    method 20  
    service instance methods 20  
Delete work order command 37  
Delete Work Order method 20  
Display work order states command 33  
Display work order transitions  
    command 33  
Display work orders command 32  
Display workflows command 34

## E

encoding 2

## F

format  
    URLs 2  
format of resource URLs 2

## G

Get Active Reachable Transitions  
    method 24

Get Active Reachable Transitions  
    (*continued*)  
        service instance methods 24  
Get Active Reachable Transitions  
    method 24  
Get Active States  
    method 23  
    service instance methods 23  
Get Active States method 23  
Get All Schedules  
    method 9  
    service schedule methods 9  
Get All Schedules method 9  
Get All Service Instances  
    method 12  
    service instance methods 12  
Get All Service Instances method 12  
Get All Services  
    method 3  
    service methods 3  
Get All Services method 3  
Get connections command 28  
Get Optim Directory Connections  
    method 2  
    service methods 2  
Get Optim Directory Connections  
    method 2  
Get Schedule  
    method 10  
    service schedule methods 10  
Get Schedule method 10  
Get Server List  
    component methods 17  
    method 17  
Get Server List method 17  
Get servers command 30  
Get Service  
    method 4  
    service methods 4  
Get service command 28  
Get Service Instance  
    method 13  
    service instance methods 13  
Get Service Instance Artifact  
    method 15  
    service instance methods 15  
Get Service Instance Artifact method 15  
Get Service Instance Artifact Names  
    method 14  
    service instance methods 14  
Get Service Instance Artifact Names  
    method 14  
Get Service Instance method 13  
Get Service method 4  
Get Work Order  
    method 21  
    service instance methods 21  
Get Work Order method 21  
Get Work Orders  
    method 20  
    service instance methods 20

Get Work Orders method 20

## M

Monitor service instances command 31

## O

optimcmd  
    -connection 28  
    -monitor 31  
    -proxy 30  
    -run 30  
    -service 28  
    -workorder 32  
    -workorder action=create 35  
    -workorder action=execute 38  
    -workorder action=purge 37  
    -workorder action=transition 36  
    -workorder action=update 37  
    -workorder display=states 33  
    -workorder display=transitions 33  
    -workorder display=workflows 34

## P

prerequisites 1

## R

Run Service  
    method 5  
    service execution methods 5  
Run service command 30  
Run Service method 5  
Run Work Order  
    method 26  
    service instance methods 26  
Run Work Order method 26

## S

server 30  
service execution methods 5  
service instance 31  
service instance methods 12  
service interface  
    security 1  
service interface methods 2  
service interface security 1  
service methods 3  
service schedule methods 7

## U

Update Schedule  
    method 8  
    service schedule methods 8

Update Schedule method 8  
Update Work Order  
    method 19  
    service instance methods 19  
Update Work Order method 19

## W

work order methods 17





Printed in USA