Cisco Tidal Enterprise Scheduler 6.2 REST API Reference Guide

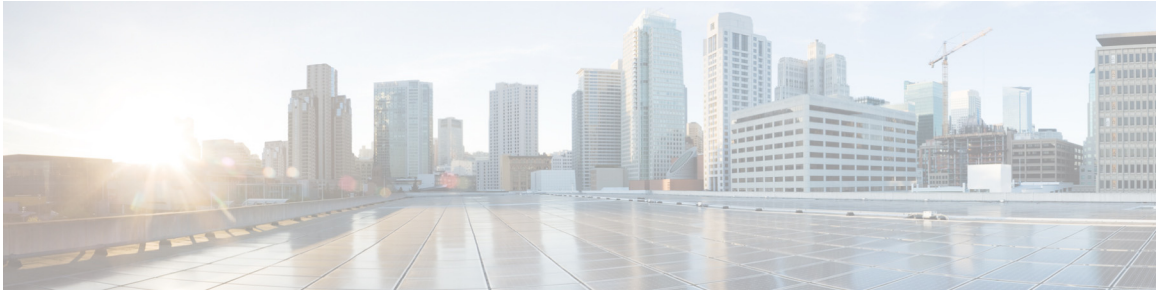
June 9, 2014

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)

Fax: 408 527-0883

Text Part Number: OL-32206-01

Cisco Tidal Enterprise Scheduler 6.2 REST API Reference Guide



Preface

This guide describes how to generate and use the Tidal Enterprise Scheduler 6.2 REST API.

Audience

This guide is for engineers who want to integrate their applications or systems with TES for Workload Management activities.

Related Documentation

See the *Cisco Tidal Enterprise Scheduler 6.2 Documentation Overview* for a list of all TES guides.

Note We sometimes update the documentation after original publication. Therefore, you should also review the documentation on Cisco.com for any updates.

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, submitting a service request, and gathering additional information, see What's New in Cisco Product Documentation at:

http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html.

Subscribe to What's New in Cisco Product Documentation, which lists all new and revised Cisco technical documentation, as an RSS feed and deliver content directly to your desktop using a reader application. The RSS feeds are a free service.

Introduction

Overview

TES (Tidal Enterprise Scheduler) is the premier enterprise job scheduling solution. In addition to the core functionality such as defining agents, agent lists, jobs, job groups, triggers, actions, events, and calendars, TES provides network support, monitoring and messaging services, security, flexibility, and fault tolerance.

TES 6.2 allows developers and integrators to customize and/or extend the features using REST API described in this guide.

Objects for REST API

Overview

The basic foundation of REST API is an ApiObject. The ApiObject provides an abstraction of TES objects (thereby hiding the implementation details).

ApiObject is the "root" class and is extended by the base objects such as **Node** and **Action**. The base classes are extended by additional objects such as **MasterNode** and **AlertAction**. A list of currently supported objects in REST API is described in Appendix (A).

ApiObject has fields for maintaining the integrity of the TES data Model. The current list includes: **lastChangeTime**, **revision**, and **revisionIndex**

ApiObject has a default behavior of CRUD operations. The list of operations currently includes:

- Create Api object – create()
- Retrieve Api object(s) – get(), getList()
- Update Api object – update()
- Delete Api object – delete()

Some REST API objects may override the basic behavior of the CRUD methods. They may also support additional methods (e.g. convenience methods).

A list of selected methods in REST API is described in Appendix (B).

Some commonly used objects and methods from REST API are covered in this guide.

Description of REST API Calls

Call Template

REST API calls can be described using the following template.

```
<?xml version="1.0" encoding="UTF-8"?>
<entry xmlns="http://purl.org/atom/ns#">
  <id>id_value</id>
  <tes:ApiObject.apiMethod xmlns:tes="http://www.tidalsoftware.com/client/teservlet">
    <tes:param1>param1_value</tes:param1>
          …
    <tes:paramN>paramN_value</tes:paramN>
  </tes:ApiObject.apiMethod>
</entry>
```

Note The standard header specifies the use of XML version 1.0 and UTF-8 encoding. The **<entry>** tag specifies the name space ("")for Atom. An object for REST API has a primary key (identifier) specified by the <id> tag. The xml name space **tes** is defined as .**tes:ApiObject.apiMethod** is the fully qualified name of the method. A method can have one or more parameters.

Calendar

Overview

TES 6.2 provides predefined calendars. A scheduled job has an associated calendar. TES 6.2 users may create custom calendars.

Calendar Type

Currently supported calendar types include:

- Fiscal Calendar
- Custom Calendar

Typical Operations

- Get a list of available calendars
- Compile a single calendar
- Compile all calendars

REST API Examples

From the browser, issue and then click **Manual Commands (Post)** to take you to the API post screen of the client. You can test the REST API examples from this screen. For valid parameters and successful execution of a request, you will get a response/result. For invalid parameters and unsuccessful execution of a request, you will get error messages and/or exceptions.

Example CAL–1: Get a list of available calendars

The **Calendar** object provides a method called **getList**. It has no parameters. The following REST API call will return a list of calendars.

```
<?xml version="1.0" encoding="UTF-8"?>
<entry xmlns="http://purl.org/atom/ns#">
  <id>xxx</id>
```

```
  <tes:Calendar.getList  xmlns:tes="http://www.tidalsoftware.com/client/teservlet">
  </tes:Calendar.getList>
</entry>
```

Example CAL–2: Get a specific calendar

The **Calendar** object provides a method called **getList**. The parameter called **id** (e.g. 30) is required, and specifies the calendar to be retrieved. The following REST API call will return the specified calendar if it exists and an exception if the calendar cannot be found.

```
<?xml version="1.0" encoding="UTF-8"?>
<entry xmlns="http://purl.org/atom/ns#">
  <id>xxx</id>
  <tes:Calendar.get xmlns:tes="http://www.tidalsoftware.com/client/teservlet">
    <tes:id>30</tes:id>
  </tes:Calendar.get>
</entry>
```

Example CAL–3: Compile a single calendar

The **Calendar** object provides a method called **compileSingleCalendar**. The parameter called **id** (e.g. 30) specifies the calendar instance to be compiled. The following REST API call will compile the calendar with **id** `12`.

```
<?xml version="1.0" encoding="UTF-8"?>
<entry xmlns="http://purl.org/atom/ns#">
  <id>xxx</id>
  <tes:Calendar.compileSingleCalendar
xmlns:tes="http://www.tidalsoftware.com/client/teservlet">
    <tes:id>30</tes:id>
  </tes:Calendar.compileSingleCalendar>
</entry>
```

Example CAL–4: Compile all calendars

The **Calendar** object also provides a method called **compileCalendars**.The parameter called **id** (e.g. 1) is required, but ignored. The following REST API call will compile all calendars.

```
<?xml version="1.0" encoding="UTF-8"?>
<entry xmlns="http://purl.org/atom/ns#">
  <id>xxx</id>
  <tes:Calendar.compileCalendars
xmlns:tes="http://www.tidalsoftware.com/client/teservlet">
    <tes:id>1</tes:id>
  </tes:Calendar.compileCalendars>
</entry>
```

Job

Overview

A job is a set of instructions about how, when and where to perform an automated task. In the job rule definition, one can specify an alias (short name) for the job, a command or script to run, an agent or a list of agents needed to run the job, the days and times to run the job, the dependencies (job, file, variable, time) and constraints that need to be satisfied before the job can be run, and other runtime criteria (such as parameter overrides, interception and job control).

A job can be defined and saved. The saved job definition can be scheduled for running as scheduled job or unscheduled (ad hoc) job. Jobs may be grouped into a hierarchy, where each job in the group can inherit properties from its parent group. Job groups can belong to other job groups.

A job's or job group's definition can be added to the production schedule either manually on demand or automatically through a calendar.

Job Type

Currently supported job types include:

- FTPJob
- JobGroup
- MPEJob
- OS400Job
- OSJob
- OVMSBatch
- OVMSCommand
- OVMSJob
- ServiceJob

Job is the base of TES 6.2 job types. The FTPJob, JobGroup, MPEJob, OS400Job, OSJob, OVMSBatch, OVMSCommand, OVMSJob and ServiceJob objects are derived from Job and share a common base structure.

Job Instance

A job instance has its own life cycle, with the associated statuses, in the production schedule.

A typical life cycle is one where the job:

- waits in the production schedule for its dependencies to be met (Waiting On Dependencies)
- enters a queue and waits for an execution slot to become available (Waiting On Resource)
- launches on its designated agent (Launched)
- starts execution successfully on its designated agent (Active)
- Completes normally (Completed Normally)

Other statuses (e.g. Completed Abnormally) are also possible depending on certain conditions and exceptions.

Typical Operations

- Create a job or a job group
- Insert a job or a job group into the production schedule
- Update a job
- List all job definitions
- Get a job definition
- Delete a job

REST API Examples

From the browser, issue and then click **Manual Commands (Post)** to take you to the API post screen of the Client Manager. You can test the REST API examples from that screen. For valid parameters and successful execution of a request, you will get a response/result. For invalid parameters and unsuccessful execution of a request, you will get error messages and/or exceptions.

Example JOB–1: Get a list of available job definitions

The **Job** object provides a method called **getList**. This method has no parameters. The following REST API call will return a list of job definitions.

```
<?xml version="1.0" encoding="UTF-8"?>
<entry xmlns="http://purl.org/atom/ns#">
  <id>xxx</id>
  <tes:Job.getList  xmlns:tes="http://www.tidalsoftware.com/client/teservlet">
  </tes:Job.getList>
</entry>
```

Example JOB-2: Get a specific job definition

The **Job** object provides a method called **get**. The parameter called **id** (e.g. 732) is required, and specifies the job definition to be retrieved. The following REST API call will return the specified job definition if it exists and an exception if the job definition cannot be found.

```
<?xml version="1.0" encoding="UTF-8"?>
<entry xmlns="http://purl.org/atom/ns#">
  <id>xxx</id>
  <tes:Job.get xmlns:tes="http://www.tidalsoftware.com/client/teservlet">
    <tes:id>732</tes:id>
  </tes:Job.get>
</entry>
```

Example JOB–3: Create a Job

The **Job** object provides the create method. The method has many parameters, but not all are required. Default values can be used by TES 6.2 for optional parameters.

```
<?xml version="1.0" encoding="UTF-8">
<entry xmlns="http://purl.org/atom/ns#">
  <id>xxx</id>
  <tes:Job.create xmlns:tes="http://www.tidalsoftware.com/client/tesservlet">
    <tes:job>
     <tes:normalexitfromrange>0</tes:normalexitfromrange>
     <tes:inheritoptions>Y</tes:inheritoptions>
     <tes:parentid>3</tes:parentid>
     <tes:type>2</tes:type>
     <tes:repeatcount>0</tes:repeatcount>
     <tes:nearoutage>3</tes:nearoutage>
     <tes:allowunscheduled>Y</tes:allowunscheduled>
     <tes:exitcodenormaloperator>1</tes:exitcodenormaloperator>
     <tes:description> Description </tes:description>
     <tes:priority>50</tes:priority>
     <tes:concurrency>1</tes:concurrency>
     <tes:repeatinterval>0</tes:repeatinterval>
     <tes:childrencount>0</tes:childrencount>
     <tes:estimatedmethod>1</tes:estimatedmethod>
     <tes:parentname>xxx Group</tes:parentname>
     <tes:ownerid>1</tes:ownerid>
     <tes:isnotes>false</tes:isnotes>
     <tes:timewindowoption>0</tes:timewindowoption>
     <tes:dependencylogic>1</tes:dependencylogic>
     <tes:active>Y</tes:active>
     <tes:inheritagent>Y</tes:inheritagent>
     <tes:normalexittorange>0</tes:normalexittorange>
     <tes:isrunbook>false</tes:isrunbook>
     <tes:repeat />
     <tes:trackingmethod>1</tes:trackingmethod>
     <tes:waitOperator>N</tes:waitOperator>
     <tes:disablecarryover>0</tes:disablecarryover>
     <tes:name>xxx job</tes:name>
     <tes:command>sleep</tes:command>
     <tes:dirty />
     <tes:durationestimated>60</tes:durationestimated>
     <tes:durationminimum>60</tes:durationminimum>
     <tes:inheritcalendar>Y</tes:inheritcalendar>
     <tes:jobdetailid>22</tes:jobdetailid>
     <tes:alias>22</tes:alias>
     <tes:historyretention>30</tes:historyretention>
     <tes:calendaroffset>0</tes:calendaroffset>
```

```
      <tes:excludeabnormalduration>4</tes:excludeabnormalduration>
      <tes:allowrerun>Y</tes:allowrerun>
      <tes:rerundependency>N</tes:rerundependency>
      <tes:savelogonly>N</tes:savelogonly>
      <tes:saveoutputoption>N</tes:saveoutputoption>
      <tes:durationmaximum>60</tes:durationmaximum>
      <tes:inherittimewindow>N</tes:inherittimewindow>
      <tes:inheritrepeat>N</tes:inheritrepeat>
    </tes:job>
  </tes:Job.create>
</entry>
```

Example JOB–4: Insert a Job into the Schedule

The **Job** object provides the insert method. One can specify the attributes of the Job (see the following sample XML).

```
<?xml version="1.0" encoding="UTF-8"?>
<entry xmlns="http://purl.org/atom/ns#">
  <id>1</id>
  <tes:Job.insert
  xmlns:tes="http://www.tidalsoftware.com/client/teservlet">
    <tes:id>xxx</tes:id>
    <tes:startdate>xxx</tes:startdate>
    <tes:fromtime>xxx</tes:fromtime>
    <tes:params>xxx</tes:params>
    <tes:vars>xxx</tes:vars>
    <tes:deps>xxx</tes:deps>
  </tes:Job.insert>
</entry>
```

JobRun

Overview

Associated with a Job is JobRun. Job Run is used to monitor and assign status, and take actions (e.g. cancel, rerun) based on the current status.

JobRun Type

Currently supported job run types include:

- FTPJobRun
- JobGroupRun
- MPEJobRun
- OS400JobRun
- OSJobRun
- OVMSBatchRun
- OVMSCommandRun
- OVMSJobRun
- ServiceJobRun

JobRun is the base of TES 6.2 job run types. The FTPJobRun, JobRunGroup, MPEJobRun, OS400JobRun, OSJobRun, OVMSCommandJobRun, OVMSJobRun and ServiceJobRun objects are derived from JobRunand and share a common base structure.

Typical Operations

- Assign status
- Cancel
- Cancel all
- Hold
- Override a JobRun instance
- Override a JobDependency instance

- Release
- Remove
- Rerun

REST API Examples

From the browser, issue and then click **Manual Commands (Post)** to take you to the API post screen of the Client Manager. You can test the REST API examples from that screen. For valid parameters and successful execution of a request, you will get a response/result. For invalid parameters and unsuccessful execution of a request, you will get error messages and/or exceptions.

Example JOBRUN–1: Override JobRun instance

The **JobRun** object provides a method called override. The parameter called **id** (e.g. 120) specifies the JobRun instance to be overridden. The following REST API call will override JobRun 120.

```
<?xml version="1.0" encoding="UTF-8"?>
<entry xmlns="http://purl.org/atom/ns#">
  <id>xxx</id>
  <tes:JobRun.override xmlns:tes="http://www.tidalsoftware.com/client/teservlet">
  <tes:id>120</tes:id>
  </tes:JobRun.override>
</entry>
```

Example JOBRUN–2: Override JobDependency instance of a JobRun

The **JobRun** object provides a method called **overrideDependency**. The method has two parameters. The first parameter called **id** (e.g. 124) specifies the JobRun instance. The second parameter called **jobrundepid** (e.g. 40)specifies the JobDep (Job dependency) instance. The following REST API call will override JobDependency 40 of JobRun 124.

```
<?xml version="1.0" encoding="UTF-8"?>
<entry xmlns="http://purl.org/atom/ns#">
  <id>1</id>
  <tes:JobRun.overrideDependency
xmlns:tes="http://www.tidalsoftware.com/client/teservlet">
    <tes:id>124</tes:id>
    <tes:jobrundepid>40</tes:jobrundepid>
  </tes:JobRun.overrideDependency>
</entry>
```

Example JOBRUN–3: Cancel a job run

The **JobRun** object provides a method called **cancel**.

```
<?xml version="1.0" encoding="UTF-8"?>
<entry xmlns="http://purl.org/atom/ns#">
  <id>1</id>
  <tes:JobRun.cancel xmlns:tes="http://www.tidalsoftware.com/client/teservlet">
    <tes:id>xxx</tes:id>
  </tes:JobRun.cancel>
</entry>
```

Example JOBRUN–4: Set status of a job run

The **JobRun** object provides a method called **assignStatus**.

```
<?xml version="1.0" encoding="UTF-8"?>
<entry xmlns="http://purl.org/atom/ns#">
  <id>1</id>
  <tes:JobRun.assignStatus xmlns:tes="http://www.tidalsoftware.com/client/teservlet">
    <tes:id>xxx</tes:id>
    <tes:status>status_value</tes:status>
  </tes:JobRun.assignStatus>
</entry>
```

Node

Overview

All TES 6.2 configurations have a Master node (also known as the Primary Master node). TES systems with licenses for Fault Tolerance will also have a Backup Master node and a Fault Monitor mode. Some TES systems may also have a Remote Master.

TES systems also allow licensing of TES Agent nodes for selected OS and platforms.

Agent types (such as MVS, OVMS, Service, Unix, and Windows) support Agent List and an associated ordering (e.g. sequential, random, load balancing) of the agents in the Agent List.

Node Type

Node is the base of TES 6.2 node types.

Currently supported node types include:

- AgentNode (base class of agents)
- BackupMasterNode
- FaultMonitorNode
- MasterNode
- MPENode
- MVSNode
- OS400Node
- OVMSNode
- RemoteMasterNode
- ServiceNode
- ZOSGatewayNode

Typical Operations

The Master Node supports the following:

- Check master connection status

- Update
- Stop scheduler
- Resume scheduler
- Create schedule
- Start today's schedule
- Pause schedule
- Pause queues
- Resume queues
- Reconnect

Common operations for an Agent node include:

- Add an agent – The operation is similar to adding a node but it associates with an agent such as Windows, UNIX, etc.
- Enable an agent – The operation is similar to enabling a connection
- Disable an agent – The operation is similar to disabling a connection

Common operations for an Agent List include:

- Get all agents by agent list id

REST API Examples

From the browser, issue and then click **Manual Commands (Post)** to take you to the API post screen of the Client Manager. You can test the REST API examples from that screen. For valid parameters and successful execution of a request, you will get a response/result. For invalid parameters and unsuccessful execution of a request, you will get error messages and/or exceptions.

Example NODE–1: Get a list of available nodes

The **Node** object provides a method called getList. It has no parameters. The following REST API call will return a list of nodes.

```
<?xml version="1.0" encoding="UTF-8"?>
<entry xmlns="http://purl.org/atom/ns#">
  <id>xxx</id>
  <tes:Node.getList  xmlns:tes="http://www.tidalsoftware.com/client/teservlet">
  </tes:Node.getList>
</entry>
```

Example NODE–2: Create schedule in master

The **MasterNode** object provides a method called createSchedule. The parameter called **id** (e.g. 1) specifies the master node. Additional parameters specify whether we need forecast, and the duration (start and end dates). The following REST API call will create a schedule for the specified node and period if the values are valid and an exception with appropriate error message for invalid data.

```
<?xml version="1.0" encoding="UTF-8"?>
<entry xmlns="http://purl.org/atom/ns#">
  <id>xxx</id>
```

```
  <tes:MasterNode.createSchedule
xmlns:tes="http://www.tidalsoftware.com/client/teservlet">
    <tes:id>1</tes:id>
    <tes:forecast></tes:forecast>
    <tes:fromDate>20100828</tes:fromDate>
    <tes:toDate>20100831</tes:toDate>
  </tes: MasterNode.createSchedule>
</entry>
```

Example NODE–3: Add an agent node and enable it

The **Node** object provides a method called **create**. The parameter type (e.g. 6 = Agent) specifies the node type to be created. The parameter ostype (e.g. 1 = Windows) specifies the node type to be created. The parameter active (e.g. Y or N) indicates whether the node should be enabled or disabled. The following REST API call will add a Windows agent node and enable it.

```
<?xml version="1.0" encoding="UTF-8"?>
<entry xmlns="http://purl.org/atom/ns#">
  <id>xxx</id>
  <tes:Node.create  xmlns:tes="http://www.tidalsoftware.com/client/teservlet">
    <tes:id>1</tes:id>
    <tes:name>name</tes:name>
            …
    <tes:type>6</tes:type>
    <tes:ostype>1</tes:ostype>
    <tes:active>Y</tes:active>
  </tes:Node.create>
</entry>
```

Action

Overview

Action is the base of TES 6.2 action types. An action is associated with a job and/or event.

Action Type

Currently supported action types include:

- AlarmptAction
- AlertAction
- EmailAction
- ITOAction
- JobAction
- LogAction
- ServiceAction
- SNMPAction
- VariableAction
- Typical Operations
- Get a list of available actions
- Create a Variable action with a variable and the update type (e.g. set, increment, decrement)

REST API Examples

From the browser, issue and then click **Manual Commands (Post)** to take you to the API post screen of the Client Manager. You can test the REST API examples from that screen. For valid parameters and successful execution of a request, you will get a response/result. For invalid parameters and unsuccessful execution of a request, you will get error messages and/or exceptions.

Example ACTION–1: Get a list of available actions

The **Action** object provides a method called **getList**. It has no parameters. The following REST API call will return a list of nodes.

```
<?xml version="1.0" encoding="UTF-8"?>
<entry xmlns="http://purl.org/atom/ns#">
  <id>xxx</id>
  <tes:Action.getList  xmlns:tes="http://www.tidalsoftware.com/client/teservlet">
  </tes:Action.getList>
</entry>
```

Example ACTION–2: Set the value of a variable

The **VariableAction** object provides a method called create. The parameter **updatetype** (e.g. 1) specifies the type of update (1= set, 2 = increment, 3 = decrement). The parameter **varid** (e.g. count) specifies the id of the variable to be updated. The parameter value (e.g. 10) specifies the value of the variable to be updated (set, incremented, or decremented). The following REST API call will set the value of the specified variable.

```
<?xml version="1.0" encoding="UTF-8"?>
<entry xmlns="http://purl.org/atom/ns#">
  <id>xxx</id>
  <tes:VariableAction.create
xmlns:tes="http://www.tidalsoftware.com/client/teservlet">
    <tes:updatetype>1</tes:updatetype>
    <tes:varid>count</tes:varid>
    <tes:value>10</tes:value>
  </tes:VariableAction.create>
</entry>
```

Queues

Overview

TES 6.2 has predefined queues (e.g. system queue). The queues may be further classified as:

- Queue for long running scheduled jobs
- Queue for short scheduled jobs
- Queue for long running unscheduled (ad-hoc) jobs
- Queue for short running unscheduled (ad-hoc) jobs

Typical Operations

- Enable a specified queue (and make it active)
- Set system queue limit – This operation specifies the number of jobs from the queue that can run at the same time

REST API Examples

From the browser, issue and then click **Manual Commands (Post)** to take you to the API post screen of the Client Manager. You can test the REST API examples from that screen. For valid parameters and successful execution of a request, you will get a response/result. For invalid parameters and unsuccessful execution of a request, you will get error messages and/or exceptions.

Example QUEUE–1: Enable a queue

The **Queue** object provides a method called **enable**. The first parameter **id** (e.g. 1) specifies the queue. The second parameter **active** (e.g. Y) specifies the status. The following REST API call will enable the specified queue and make it active.

```
<?xml version="1.0" encoding="UTF-8"?>
<entry xmlns="http://purl.org/atom/ns#">
  <id>xxx</id>
  <tes:Queue.enable xmlns:tes="http://www.tidalsoftware.com/client/teservlet">
    <tes:id>1</tes:id>
    <tes:active>Y</tes:joblimit>
  </tes:Queue.enable>
```

```
</entry>
```

Appendix

Object Descriptions

Table A-1 Action

Name	Atom	Type	Description
id	Element	Int	Id (primary key)
name	Element	String	Object Name (length 64)
description	Element	String	Description (length 4000)
ownerid	Element	Integer	Owner Id (either user or workgroup)
publicflag	Element	String	Public flag? (Y/N)
lastusermodifiedtime	Element	Date	Last user modified time
id	Id	--	Id (primary key)
name	Title	--	Object name (length 64)

Table A-2 AgentLicense

Name	Atom	Type	Description
id	Element	Integer	Id (primary key)
name	Element	String	Object name of AgentLicense
ostype	Element	String	Operating system type of the agent
serial	Element	String	Serial of AgentLicense
demo	Element	String	Demo license for the agent
expiration	Element	String	Expiration data of AgentLicense
limit	Element	String	Limit of AgentLicense
floating	Element	String	Floating license for the Agent
cpucount	Element	String	Number of CPUs licensed for the agent
servicexml	Element	String	Service xml fo agent portion of a service connection
license	Element	String	Agent License

Table A-2 AgentLicense (Continued)

Name	Atom	Type	Description
agentlicense	ENTRY		GET, GETLIST
id	Id	--	Id (primary key)
name	Title	--	Object name of AgentLicense

Table A-3 AgentList

Name	Atom	Type	Description
description	Element	String	Description (length 4000)
name	Element	String	Object name of (length 64)
parentid	Element	Integer	Parent ID
fullpath	Element	String	FullPath (length 4000)
listtype	Element	Short	List type (short)
ostype	Element	Short	Operating system type (short)
id	Element	int	Id (primary key)
lastusermodifiedtime	Element	Date	Last user modified time
agentlist	ENTRY		
id	Id	--	Id (primary key)
name	Title	--	Object name (length 64)

Table A-4 AgentListJoin

Name	Atom	Type	Description
id	Element	int	Id (primary key)
agentlistid	Element	Integer	Agent list id of the parent record
agentseq	Element	Short	The order (short) of this agent in the list
connectionid	Element	Integer	AgentID of this associated agent
agentlistjoin	ENTRY		CREATE, DELETE, GET, GETLIST
id	Id	--	Id (primary key)

Table A-5 AgentNode

Name	Atom	Type	Description
machine	Element	String	Machine name (1024 chars)
joblimit	Element	Short	Job limit (applies to all except Email, TA and z/OS)
port	Element	Short	Agent port
runuser	Element	Integer	Runtime user

Table A-6 AlarmptAction

Name	Atom	Type	Description
options	Element	String	Alarmpoint action options (length 30). Alarmpoint menu options to be sent to AP (e.g., YYNNNNNNNN)
contact	Element	String	Contact User ID or TES Variable resolving to a User ID to be notified of the AP event.
escalation	Element	String	Escalation User ID or TES Variable resolving to a User ID. It is only required for Option to escalate. **getEscalation** and **setEscalation**.
recoveryjobid	Element	Integer	Recovery Job ID for Option to add recover job. **getRecoveryJobID** and **setRecoveryJobID**.
alarmptaction	ENTRY		CREATE, DELETE, GET, GETLIST

Table A-7 Alert

Name	Atom	Type	Description
id	Element	int	Id (primary key)
jobrunid	Element	Integer	Jobrun Id
jobid	Element	Integer	Job Id
responseuser	Element	Integer	Response user
timeasstring	Element	String	Time (Date) of Alert. **getTimeAsString** and **setTimeAsString**
time	Element	Date	Time (Date)
responsetimeasstring	Element	String	Response time (String)
responsetime	Element	Date	Response time (Date)
description	Element	String	Description (clob 512M)
response	Element	String	Response
type	Element	Short	Type (Short)
level	Element	Short	Level (Short)
state	Element	Short	State (Short)
Lastusermodifiedtime	Element	Date	Last user modified time (Date)
alert	ENTRY		CREATE, DELETE, GET, GETLIST
id	Id	--	Id (primary key)

Table A-8 AlertAction

Name	Atom	Type	Description
level	Element	Short	Level (Short)
message	Element	String	Message
alertaction	ENTRY		CREATE, DELETE, GET, GETLIST

Table A-9 Api

Name	Atom	Type	Description
lastchangetime	Element	Date	Last change time (Date)

Table A-10 BackupMasterNode

Name	Atom	Type	Description
machine	Element	Short	Machine name (length 1024) of BackupMasterNode
port	Element	Short	Backup to master port
alertaction	ENTRY		CREATE, DELETE, GET, GETLIST

Table A-11 BusinessView

Name	Atom	Type	Description
id	Element	int	Id (primary key)
name	Element	String	Name (length 256)
ownerid	Element	Integer	Owner Id
publicflag	Element	String	Public flag? (Y/N)
layout	Element	String	Layout (clob)
groupid	Element	Integer	Group Id
viewjobs	Element	Integer	View jobs
lastusermodifiedtime	Element	Date	Last user modified time
businessview	ENTRY		CREATE, DELETE, GET, GETLIST
id	Id	--	Id (primary key)
			(Continued)
name	Title	--	Name (length 256)

Table A-12 Calendar

Name	Atom	Type	Description
id	Element	int	Id (primary key)
name	Element	String	Calendar name (length 64)
calendar	Element	Integer	Calendar
createtime	Element	Date	Time of calendar creation
description	Element	String	Description of Calendar
fiscal	Element	Short	Fiscal calendar
forecastdateasstring	Element	String	Calendar forecast date (String)
forecastdate	Element	Date	Calendar forecast date (Date)
frequency	Element	Short	Frequency
fromdateasstring	Element	String	From date (String)
fromdate	Element	Date	From date (Date)
lastusermodifiedtime	Element	Date	Last user modified time
lastcompiletime	Element	Date	Last compile time
monthdays	Element	String	Days in month (lenght 32)
months	Element	String	Calendar months (length 12)
occur	Element	Short	Occur
offset	Element	Short	Offset
ownerid	Element	Integer	Owner Id
pub	Element	String	Pub
subtype	Element	Short	Calendar subtype
timeframe	Element	Short	Time frame
totalcount	Element	Short	Total count
type	Element	Short	Calendar type
untildateasstring	Element	String	Until date (String)
untildate	Element	Date	Until date (Date)
weekdays	Element	String	Week days (length 7)
weeks	Element	String	Weeks (length 5)
intsect	Element	String	Intsect (length 7)
typestring	Element	String	Type string (e.g. Group)
calendar	ENTRY		CREATE, DELETE, GET, GETLIST
id	Id	--	Id (primary key)
name	Title	--	Calendar name (length 64)

Table A-13 CalendarChild

Name	Atom	Type	Description
id	Element	int	Id (primary key)
sequencenumber	Element	Short	Sequence number
calendarid	Element	Integer	Calendar Id
logic	Element	Short	Logic
offset	Element	Short	Offset
parentid	Element	Integer	Parent Id
calendarchild	ENTRY	Integer	CREATE, DELETE, GET, GETLIST
id	Id	--	Id (primary key)

Table A-14 CalendarCondition

Name	Atom	Type	Description
id	Element	int	Id (primary key)
action	Element	Short	Action
actionid	Element	Integer	Action Id
calendarid	Element	Integer	Calendar Id
conflictcalendarid	Element	Integer	Conflict calendar Id
offset	Element	Short	Offset
sequencenumber	Element	Short	Sequence number
calendarcondition	ENTRY		CREATE, DELETE, GET, GETLIST
id	Id	--	Id (primary key)

Table A-15 CalendarYear

Name	Atom	Type	Description
id	Element	int	Id (primary key)
calendarid	Element	Integer	Calendar Id
dates1	Element	String	Dates1 (length 183)
dates2	Element	String	Dates2 (length 183)
calendaryear	Element	Integer	Calendar year
calendaryear	ENTRY		CREATE, DELETE, GET, GETLIST
id	Id	--	Id (primary key)

Table A-16 EmailAction

Name	Atom	Type	Description
fromaddr	Element	String	From email address (length 4000)
toaddresses	Element	String	Delimited list of user id's containing the email address of token for the workgroup to send to.
otheraddress	Element	String	External email addresses to send to (e.g. Other email addresses)
subject	Element	String	Subject of the email (length 1024)
message	Element	String	Message (email content)
attachment	Element	String	Attachment (path to any attachments to include in the email)
emailaction	ENTRY		CREATE, DELETE, GET, GETLIST

Table A-17 Event

Name	Atom	Type	Description
id	Element	Int	Id (primary key)
eventinfo	Element	String	Event info
pub	Element	String	Pub
alljobs	Element	String	Whether to apply to all jobs? Y or N (default to N)
description	Element	String	Description of event (length 4000)
active	Element	String	Enabled event? Y or N (default to N)
			(Continued)
eventtype	Element	Short	Event type (Short)
jobcontrol	Element	Short	Job control (Short)
exitcodefromval	Element	Integer	From value of exit code
exitcodetoval	Element	Integer	To value of exit code
name	Element	String	Event name (lenth 64)
nodeid	Element	Integer	Node id for monitor event types (e.g. file, variable). Only required for monitoring events.
ownerid	Elementt	Integer	Event owner Id (either user or workgroup)
serviceid	Element	Integer	Service id for service events. Only required for service events.
triggerid	Element	Short	Trigger id
lastusermodifiedtime	Element	Date	Last user modified time
maxreruns	Element	Integer	Maximum number of reruns
monitor	Element	Short	Event monitor (Short)

Table A-17 Event (Continued)

Name	Atom	Type	Description
params	Element	String	Event parameters (length 30)
eventactioncount	Element	Integer	Event action count
eventactionrunid	Element	Integer	Event action run id
actionjoinlist	Element	String	Action Join list
jobclassjoinlist	Element	String	Job Class Join list
events	ENTRY		CREATE, DELETE, GET, GETLIST
id	Id	--	Id (primary key)
name	Title	--	Event name (length 64)

Table A-18 EventActionJoin

Name	Atom	Type	Description
id	Element	int	Id (primary key)
actionid	Element	Integer	Action id (link into the actions)
jobcontrol	Element	Short	Type of job contrl in the case of Job Control action. This value is saved back to the linked event record.
actionorder	Element	Integer	The order of the action within the event
rerunmax	Element	Integer	(Continued) Maximum number of reruns in the case of Job Control action
triggerid	Element	Integer	Trigger ID (link into the events)
type	Element	Integer	Action type
eventactionjoin	ENTRY		CREATE, DELETE, GET, GETLIST
id	Id	--	Id (primary key)

Table A-19 EventActionRun

Name	Atom	Type	Description
id	Element	int	Id (primary key)
payload	Element	String	Payload (clob 64M)
status	Element	Integer	Status
createdateasstring	Element	String	Create date (String) **getCreateDatAsString** **setCreateDateAsString**
createdate	Element	Date	Create date

Table A-19 EventActionRun

Name	Atom	Type	Description
rundateasstring	Element	String	Run date (String) **getRunDateAsString** **setRunDateAsString**
rundate	Element	String	Run date
parentid	Element	String	Parent id
transactionvalue	Element	Integer	Transaction value
expire	Element	Date	Expiration date
data	Element	String	Data of EventActionRun
eventid	Element	Integer	Event id
actionid	Element	Integer	Action id
jobid	Element	Integer	Job id
jobrunid	Element	Integer	Job run id
nodeid	Element	Integer	Node id
nodetarget	Element	Integer	Node target
serviceid	Element	Integer	Service id
userid	Element	Integer	User id
eventactionrun	ENTRY		CREATE, DELETE, GET, GETLIST
			(Continued)
id	Id	--	Id (primary key)

Table A-20 EventCalendarJoin

Name	Atom	Type	Description
id	Element	int	Id (primary key)
triggerid	Element	Integer	Trigger id (Link to the trigger)
calendarid	Element	Integer	Calendar id
offset	Element	Integer	Offset
fromdateasstring	Element	String	From date (String)
fromdate	Element	Date	From date (Date)
untildateasstring	Element	String	Until date (String)
untildate	Element	Date	Until date (Date)
maxruns	Element	Integer	Maximum number of runs
maxinterval	Element	Integer	Maximum interval
timewindow	Element	String	Time window (lenth 1024)
ignoreoccurrence	Element	String	Ignore occurrence (length 1)
ignoreconstraints	Element	String	Ignore contstraints (length 1)
ignoreTimeWindow	Element	String	Ignore time window (length 1)

Table A-20 EventCalendarJoin

Name	Atom	Type	Description
eventcalendarjoin	ENTRY		CREATE, DELETE, GET, GETLIST
id	Id	--	Id (primary key)

Table A-21 EventJobClassJoin

Name	Atom	Type	Description
id	Element	int	Id (primary key)
jobclassid	Element	Integer	Job class id (Link to the job class recorde)
actionorder	Element	Integer	Order of the event within the job
triggerid	Element	Integer	Trigger id (Link to the events)
triggerclasstype	Element	Short	Trigger class type
eventjobclassjoin	ENTRY		CREATE, DELETE, GET, GETLIST
			(Continued)
id	Id	--	Id (primary key)

Table A-22 EventJobJoin

Name	Atom	Type	Description
id	Element	int	Id (primary key)
jobid	Element	Integer	Job id (Link to the job)
level	Element	Integer	Level
actionorder	Element	Integer	Order of the event within the job
triggerid	Element	Integer	Trigger id (Link to the events)
triggertype	Element	Short	Trigger type
eventjobjoin	ENTRY		CREATE, DELETE, GET, GETLIST
id	Id	--	Id (primary key)

Table A-23 FaultMonitorNode

Name	Atom	Type	Description
machine	Element	String	Machine name (1024 chars)
fmclientport	Element	Short	Fault Monitor Client port
fmmasterport	Element	Short	Fault Monitor Master port

Table A-23 FaultMonitorNode

Name	Atom	Type	Description
faultmonitornode	ENTRY		CREATE, DELETE, GET, GETLIST

Table A-24 FileDependency

Name	Atom	Type	Description
connectionid	Element	Integer	Connection id
fileextent	Element	Long	File extent
filename	Element	String	File name (length 4000)
filetype	Element	Short	File type
id	Element	int	Id (primary key)
jobid	Element	Integer	(Continued) Job id
inheritagent	Element	String	Inherit agent
inheritedagentname	Element	String	Inherited agent name (length 64)
filedependency	ENTRY		CREATE, DELETE, GET, GETLIST
id	Id	--	Id (primary key)

Table A-25 FiscalCalendar

Name	Atom	Type	Description
id	Element	int	Id (primary key)
daybegin	Element	Short	Day begin
fixedbegin	Element	String	Fixed begin (length 1)
monthbegin	Element	Short	Month begin
name	Element	String	Name (length 30) of FiscalCalendar. **getName** **setName** **hasValueName**.
period1	Element	Short	First period
period2	Element	Short	Second period
period3	Element	Short	Third period
quarter	Element	Integer	Quarter
quarterbreakdown	Element	String	Quarter breakdown
startyear	Element	Integer	Start year
calyear	Element	Integer	Year
fiscalcalendar	ENTRY		CREATE, DELETE, GET, GETLIST

Table A-25 FiscalCalendar

Name	Atom	Type	Description
id	Id	--	Id (primary key)
name	Title	--	Name (length 30) of FiscalCalendar. **getName** **setName** **hasValueName**

Table A-26 FolderConfiguration

Name	Atom	Type	Description
id	Element	int	Id (primary key)
name	Element	String	Name (length 128)
parentid	Element	Integer	Parent id
objecttype	Element	Integer	Object type
linkedto	Element	Integer	Link to
type	Element	Integer	Type
logicaltype	Element	Integer	Logical type
ownerid	Element	Integer	Owner id
publicflag	Element	String	Public flag? (Y/N)
data	Element	String	Data of FolderConfiguration
lastusermodifiedtime	Element	Date	Last user modified time
folderconfiguration	ENTRY		CREATE, DELETE, GET, GETLIST
id	Id	--	Id (primary key)
name	Title	--	Name (length 128)

Table A-27 FTPJob

Name	Atom	Type	Description
ftpuserid	Element	Integer	Ftp user id
localuserid	Element	Integer	Local user id
localpath	Element	String	Local path
remotepath	Element	String	Remote path (length 255)
quote	Element	String	Quote (length 255)
ftpoperation	Element	Short	Ftp operation
ftpprotocol	Element	String	Ftp protocol
ftphost	Element	String	Ftp host (length 255)
filename	Element	String	File name

Table A-27 FTPJob

Name	Atom	Type	Description
newfilename	Element	String	New file name
asciiformat	Element	String	Ascii format
anonymous	Element	String	Anonymous
replacefiles	Element	String	Replace files
encryptioncypherselection	Element	String	(Continued) Encryption cypher selection.
userpasswordauthenticatio n	Element	String	User password authentication
passphrase	Element	String	Pass phrase
privatekeylocation	Element	String	Private key location
ftpparameters	Element	String	Ftp parameters (length 255)
extendedinfo	Element	String	Extended info
ftpjob	ENTRY		CREATE, DELETE, GET, GETLIST

Table A-28 FTPJobRun

Name	Atom	Type	Description
ftpuserid	Element	Integer	Ftp user id
localuserid	Element	Integer	Local user id
localpath	Element	String	Local path
remotepath	Element	String	Remote path (length 255)
quote	Element	String	Quote (length 255)
ftpoperation	Element	Short	Ftp operation
ftpprotocol	Element	String	Ftp protocol
ftphost	Element	String	Ftp host (length 255)
filename	Element	String	File name
newfilename	Element	String	New file name
asciiformat	Element	String	Ascii format
anonymous	Element	String	Anonymous
replacefiles	Element	String	Replace files
encryptioncypherselection	Element	String	Encryption cypher selection.
userpasswordauthenticatio n	Element	String	User password authentication
passphrase	Element	String	Pass phrase
privatekeylocation	Element	String	Private key location
ftpparameters	Element	String	Ftp parameters (length 255)

Table A-28 FTPJobRun (Continued)

Name	Atom	Type	Description
extendedinfo	Element	String	Extended info
ftpjobrun	ENTRY		CREATE, DELETE, GET, GETLIST

Table A-29 ImageRepository

Name	Atom	Type	Description
id	Element	int	Id (primary key)
info	Element	String	Info/Data (clob 10M)
name	Element	String	Name (length 128)
filename	Element	String	File name (length 128)
lastusermodifiedtime	Element	Date	Last user modified time
size	Element	Integer	Size of ImageRepository. **getSize** and **setSize**
type	Element	String	Type (length 5). **getType** and **setType**
imagerepository	ENTRY		CREATE, DELETE, GET, GETLIST
id	Id	--	Id (primary key)
name	Title	--	Name (length 128)

Table A-30 ImageRepositoryLink

Name	Atom	Type	Description
id	Element	int	Id (primary key)
imageid	Element	Integer	Image id
jobclass	Element	Integer	Job class
jobmst	Element	Integer	Job master
imagerepository	ENTRY		CREATE, DELETE, GET, GETLIST
id	Id	--	Id (primary key)

Table A-31 InfoNodeJoin

Name	Atom	Type	Description
id	Element	int	Id (primary key)
infonodeid	Element	Integer	Info node id
runnodeid	Element	Integer	Run node id

Table A-31 InfoNodeJoin

Name	Atom	Type	Description
infonodejoin	ENTRY		CREATE, DELETE, GET, GETLIST
id	Id	--	Id (primary key)

Table A-32 InterMasterVariable

Name	Atom	Type	Description
id	Element	int	Id (primary key)
name	Element	String	Name (length 64)
oldvalue	Element	String	Old value
currentvalue	Element	String	Current value
variableid	Element	Integer	Variable id
nodeid	Element	Integer	Node id
varpublic	Element	String	Public variable (length 1)
varreadonly	Element	String	Read only variable (length 1)
variabletype	Element	Short	Variable type
lastusermodifiedtime	Element	Date	Last user modified time
intermastervariable	ENTRY		CREATE, DELETE, GET, GETLIST
id	Id	--	Id (primary key)
name	Title	--	Name (length 64)

Table A-33 ITOAction

Name	Atom	Type	Description
severity	Element	Short	Severity
community	Element	String	Community
message	Element	String	Message
itotype	Element	Short	ITO type
itoaction	ENTRY		CREATE, DELETE, GET, GETLIST

Table A-34 JESDependency

Name	Atom	Type	Description
condition	Element	Short	Condition
connectionid	Element	Integer	Connection id

Table A-34 JESDependency

Name	Atom	Type	Description
id	Element	int	Id (primary key)
ignore	Element	String	Ignore (length 1)
			(Continued)
includeabend	Element	String	Include abnormal end
instanceoffset	Element	Short	Instance offset (applies to specific and offset only, else 0)
jobid	Element	Integer	Job id
jobname	Element	String	Job name (length 255)
occurrence	Element	Short	Occurrence
procstep	Element	String	Proc step (length 255)
ccrangehi	Element	Integer	Cc range high
ccrangelo	Element	Integer	Cc range low
jesdependency	ENTRY		CREATE, DELETE, GET, GETLIST
id	Id	--	Id (primary key)

Table A-35 Job

Name	Atom	Type	Description
id	Element	int	Id (primary key)
active	Element	String	Active (length 1)
agentid	Element	Integer	Agent id
agentlistid	Element	Integer	Agent list id
alias	Element	String	Alias (length 10)
allowrerun	Element	String	Allow rerun (length 1)S
allowunscheduled	Element	String	Allow unscheduled (length 1)
calendarfromdateasstring	Element	String	Calendar from date (String)
calendarfromdate	Element	Date	Calendar from date (Date)
calendarid	Element	Integer	Calendar id
calendaroffset	Element	Short	Calendar offset
calendaruntildateasstring	Element	String	Calendar until date (String)
calendaruntildate	Element	Date	Calendar until date (Date)
type	Element	SHort	Type of job
concurrency	Element	Short	Concurrency
createtime	Element	Date	Create time
dependencylogic	Element	Short	Dependency logic
			(Continued)

Table A-35 Job (Continued)

Name	Atom	Type	Description
dirty	Element	String	Dirty (length1)
disablecarryover	Element	Short	Disable carryover
durationestimated	Element	Integer	Estimated duration
durationmaximum	Element	Integer	Maximum duration
durationminimum	Element	Integer	Minimum duration
estimatedmethod	Element	Short	Estimated method
excludeabnormalduration	Element	Integer	Exclude abnormal duration
inheritagent	Element	String	Inherit agent (length 1)
inheritcalendar	Element	String	Inherit calendar (length 1)
inheritoptions	Element	String	Inherit options (length 1)
inheritrepeat	Element	String	Inherit repeat (length 1)
inherittimewindow	Element	String	Inherit time window (length 1)
jobclassid	Element	Integer	Job class id
lastusermodifiedtime	Element	Date	Last user modified time
name	Element	String	Job name (length 256)
nearoutage	Element	Short	Near outage
isnotes	Element	Boolean	Is notes
notes	Element	String	Notes
jobdetailid	Element	Integer	Job detail id
ownerid	Element	Integer	Owner id
parentid	Element	Integer	Parent id
parentname	Element	String	Parent name (length 4000)
fullpath	Element	String	Full path (length 4000)
repeat	Element	String	Repeat
repeatcount	Element	Short	Repeat count
repeatinterval	Element	Short	Repeat interval of job
rerundependency	Element	String	Rerun dependency (length1)
isrunbook	Element	Boolear	Is run book
runbook	Element	String	Run book
runuserid	Element	Integer	Run user id
timewindowfromtimeasstring	Element	String	Time window from time (String)
timewindowfromtime	Element	Date	Timewindow from time (Date)
			(Continued)
timewindowoption	Element	Short	Time window option
timewindowuntilmeasstring	Element	String	Time window until time (String)

Table A-35 Job (Continued)

Name	Atom	Type	Description
timewindowuntiltime	Element	Date	Time window until time (String)
waitoperator	Element	String	Wait operator
exitcodenormaloperator	Element	Short	Exit code normal operator
historyretention	Element	Short	Retention history
normalexitfromrange	Element	Short	From range of normal exit
normalexittorange	Element	Short	To rang of normal exit
priority	Element	Short	Priority
saveoutputoption	Element	String	Option to save output (length 1)
trackingcommand	Element	String	Tracking command (length 4000)
unixprofile	Element	Short	Unix profile
alternateoutputfile	Element	String	Alternate output file (length 4000)
command	Element	String	Command (length 4000)
environmentfile	Element	String	Environment file (length 4000)
parameters	Element	String	Parameters
workingdirectory	Element	String	Working directory (length 4000)
serviceid	Element	Integer	Service id
variables	Element	String	Variables
jobdeplist	Element	String	Job dependency list
hasparent	Element	String	Job has parent
successorsvalue	Element	String	Has successor (length 1)
predecessorsvalue	Element	String	Has predecessor (length 1)
job	ENTRY		CREATE, DELETE, GET, GETLIST
id	Id	--	Id (primary key)
name	Title	--	Job name (length 256)

Table A-36 JobAction

Name	Atom	Type	Description
jobid	Element	Integer	Job id of the job to run
agentid	Element	Integer	Agent id of agent that runs the job
userid	Element	Integer	Runtime user
parameters	Element	String	Any parameter overrides
jobaction	ENTRY	--	CREATE, DELETE, GET, GETLIST

Table A-37 JobClass

Name	Atom	Type	Description
id	Element	int	Id (primary key)
name	Element	String	Name (length 64)
description	Element	String	Description (length 4000)
lastusermodifiedtime	Element	Date	Last user modified time
jobclass	ENTRY	--	CREATE, DELETE, GET, GETLIST
id	Id	--	Id (primary key)
name	Title	--	Name (length 64)

Table A-38 JobDependency

Name	Atom	Type	Description
operator	Element	Short	Operator
status	Element	Short	Status
logic	Element	Short	Dependency logic
datoffset	Element	Short	Date offset
depjobid	Element	Integer	Dependency job id
id	Element	int	Id (primary key)
ignoredep	Element	String	Ignore dependency (length 1)
ingroup	Element	String	In group (length 1)
instanceoffset	Element	Short	Instance offset
jobid	Element	Integer	Job id
			(Continued)
lastusermodifiedtime	Element	Date	Last user modified time
jobdependency	ENTRY		CREATE, DELETE, GET, GETLIST

Table A-38 JobDependency (Continued)

Name	Atom	Type	Description
id	Id	--	Id (primary key)

Table A-39 JobGroup

Name	Atom	Type	Description
jobgroup	ENTRY		CREATE, DELETE, GET, GETLIST

Table A-40 JobOutput

Name	Atom	Type	Description
id	Element	int	Id (primary key)
jobrunid	Element	Integer	Job run id
outputlength	Element	int	Output length
outputtext	Element	String	Output text (clob 512M)
joboutput	ENTRY		CREATE, DELETE, GET, GETLIST
id	Id	--	Id (primary key)

Table A-41 JobReport

Name	Atom	Type	Description
id	Element	int	Id (primary key)
type	Element	Integer	Type
date	Element	Date	Date
status	Element	Integer	Status
jobid	Element	Integer	Job id
jobrunid	Element	Integer	Job run id
infonodeid	Element	Integer	Info node id
runnodeid	Element	Integer	Run node id
			(Continued)
data	Element	String	Data
jobreport	ENTRY		CREATE, DELETE, GET, GETLIST
id	Id	--	Id (primary key)

Table A-42 JobRun

Name	Atom	Type	Description
id	Element	int	Id (primary key)
parentid	Element	Integer	Parent id
fullpath	Element	String	Full path (length 4000)
calendarid	Element	Integer	Calendar id
jobid	Element	Integer	Job id
serviceid	Element	Integer	Service id
instance	Element	Integer	Instance
grouinstance	Element	Short	Group instance
agentinstance	Element	Short	Agent instance
status	Element	Short	Status
originalstatus	Element	Short	Original status
statuschangetime	Element	Date	Status change time
metflag	Element	String	Met flag (length 1)
type	Element	Short	Job type
productiondateasstring	Element	String	Production date (String) **getProductionDateAsString** **setProductionDataAsString**
productiondate	Element	Date	Production date.
currenteststarttimeasstring	Element	String	Current estimated start time (String) **getCurrentEstimatedStartTimeAsString** **setCurrentEstimatedStartTimeAsString** **hasValueCurrentEstimatedStartTimeAsString**
currenteststarttime	Element	Date	Current estimated start time
duration	Element	Integer	Duration
estimatedduration	Element	Integer	Estimated duration
estimatedstarttimeasstring	Element	String	Estimated start time (String)
estimatedstarttime	Element	Date	Estimated start time (Date)
			(Continued)
actualstarttimeasstring	Element	String	Actual start time (String)
actualstarttime	Element	Date	Actual start time (Date)
nodeid	Element	Integer	Node id
nodelistid	Element	Integer	Node list id
queueid	Element	Integer	Queue id
adhoc	Element	String	Adhoc job (length 1)
waitop	Element	String	Wait op (length 1)
fromtimeasstring	Element	String	From time (String)
fromtime	Element	Date	From time (Date)

Table A-42 JobRun (Continued)

Name	Atom	Type	Description
untiltimeasstring	Element	String	Until time (String)
untiltime	Element	Date	Until time (Date)
owner	Element	Integer	Owner
runuserid	Element	Integer	Run user id
proxy2	Element	Integer	Proxy2
concur	Element	Short	Concur
priority	Element	Short	Priority
command	Element	String	Command (length 4000)
workingdirectory	Element	String	Working directory (length 4000)
envfile	Element	String	Environment file (length 4000)
exitcode	Element	Integer	Exit code
timewindow	Element	Short	Time window
saveoutput	Element	String	Save output (length 1)
trackmethod	Element	Short	Track method
trackcommand	Element	String	Track command (length 4000)
reruns	Element	Short	Reruns
flags1	Element	Integer	Flags1
outputtxt	Element	String	Output text (length 1)
outputname	Element	String	Output name (length 4000)
nontimedependencies	Element	String	No time dependencies (length 1)
allowadhoc	Element	String	Allow adhoc job (length 1)
hasdependents	Element	String	Has dependents (length 1)
rundate	Element	Date	Run date
			(Continued)
batch	Element	Short	Batch
params	Element	String	Parameters
variables	Element	String	Variables
jobparentid	Element	Integer	Job parent id
minruntime	Element	Integer	Minimum run time
maxruntime	Element	Integer	Maximum run time
dependencylogic	Element	Short	Dependency logic
jobclassid	Element	Integer	Job class id
interval	Element	Short	Interval
intervalcount	Element	Short	Interval count
rerun	Element	String	Rerun (length 1)
timebatch	Element	Short	Time batch

Table A-42 JobRun (Continued)

Name	Atom	Type	Description
timedifference	Element	Integer	Time difference
externidstring	Element	String	External id
outputlength	Element	Integer	Output length
normalexit	Element	Short	Normal exit
normalrange	Element	Short	Normal range
normalop	Element	Short	Normal op
dependencyrerun	Element	String	Dependency rerun (length 1)
carryover	Element	Short	Carry over
resetsuccessor	Element	String	Reset successor (length 1)
ignoretimedependency	Element	String	Ignore time dependency (length 1)
launchacknowledged	Element	Integer	Launch acknowledged
fromstep	Element	Short	From step
savelogonly	Element	String	Save log only (length 1)
expire	Element	Date	Expiration date
dependencies	Element	String	Dependencies
deptextlength	Element	Integer	Dependency text length
transactionid	Element	Integer	Transaction id
triggeractionrun_transaction	Element	Integer	Trigger action run transaction
triggeractionrunid	Element	Integer	Trigger aciton run id
hasresourceslockedflag	Element	String	Has resources locked flag (length 1)
			(Continued)
xmldata	Element	String	Xml data
dirty	Element	String	Dirty
estmethod	Element	Short	Estimation method
nearoutage	Element	Short	Near outage (length 1)
queuetime	Element	Date	Queue time
trackcl	Element	String	Track cl (length 1)
statuscl	Element	String	Status cl (length 1)
abrtonclderr	Element	String	Abort on child error (length 1)
launchtimeasstring	Element	String	Launch time (String)
launchtime	Element	Date	Launch time (Date)
name	Element	String	Name (length 255)
childjobloadedflag	Element	String	Child job loaded flag (length 1)
runinfoxml	Element	String	Info xml (clob 32M)
jobrunxml	Element	String	Job run xml (clob 32M)

Table A-42 JobRun (Continued)

Name	Atom	Type	Description
successorlist	Element	String	Successor list (clob 32M)
calcdatevarlist	Element	String	Calc date var list (clob 32M)
successorsvalue	Element	String	Has successor (length 1)
predecessorsvalue	Element	String	Has predecessor (length 1)
jobrun	ENTRY		CREATE, DELETE, GET, GETLIST
id	Id	--	Id (primary key)
name	Title	--	Name (length 255)

Table A-43 JobRunGroup

Name	Atom	Type	Description
jobgrouprun	ENTRY		CREATE, DELETE, GET, GETLIST

Table A-44 LogAction

Name	Atom	Type	Description
loglevel	Element	Short	Log level
message	Element	String	Message (length 4000)
winevent	Element	String	Windows event (length 1)
eventnum	Element	String	Event num (length 30)
logaction	ENTRY		CREATE, DELETE, GET, GETLIST

Table A-45 MasterLicense

Name	Atom	Type	Description
id	Element	int	Id (primary key)
customer	Element	String	Customer
name	Element	String	Name
expiration	Element	String	Expiration date
demomode	Element	boolean	Demonstration mode
enterprisemode	Element	boolean	Enterprise mode
annualmode	Element	boolean	Annual mode
emergencymode	Element	boolean	Emergency mode
productmode	Element	boolean	Product mode

Table A-45 MasterLicense (Continued)

Name	Atom	Type	Description
licensemode	Element	String	License mode
agents	Element	String	Agents
floating	Element	String	Floating
ostype	Element	String	Operating system type
database	Element	String	Database
serial	Element	String	Serial
clients	Element	String	Clients
zekefloat	Element	String	Zeke float
zekeagents	Element	String	Zeke agents
zosfloat	Element	String	ZOS float
zosagents	Element	String	ZOS agents
os400agents	Element	String	OS400 agents
			(Continued)
servicexml	Element	String	Service xml
options	Element	String	Options
license	Element	String	License
masterlicense	ENTRY		CREATE, DELETE, GET, GETLIST
id	Id	--	Id (primary key)
customer	Title	--	Customer

Table A-46 MasterNode

Name	Atom	Type	Description
machine	Element	String	Machine (length 1024)
port	Element	Short	Agent to Master port
clientport	Element	Short	Client to Master port
hostloglevel	Element	Short	Host log level
masternode	ENTRY		CREATE, DELETE, GET, GETLIST

Table A-47 MessageCatalog

Name	Atom	Type	Description
id	Element	int	Id (primary key)
type	Element	Short	Message catalog type
exclude	Element	String	Exclude (length 1)

Table A-47 MessageCatalog

Name	Atom	Type	Description
eventlog	Element	String	Windows event log (length 1)
messagetxt	Element	String	Message text (length 255)
languageid	Element	Integer	Language id
messagecatalog	ENTRY		CREATE, DELETE, GET, GETLIST
id	Id	--	Id (primary key)

Table A-48 MessageLog

Name	Atom	Type	Description
id	Element	int	Id (primary key)
createdateasstring	Element	String	Create date (String)
createdate	Element	Date	Create date (Date)
masgseveritytype	Element	Short	Message severity type
msgsourcetype	Element	Short	Message source type
msglogtxt	Element	String	Message log text (length 4000)
computer	Element	String	Name of computer for log record (length 16)
origuserid	Element	Integer	Log of original ID of user prior to impersonating another user.
userid	Element	Integer	User id
nodeid	Element	Integer	Node id
jobid	Element	Integer	Job id
jobrunid	Element	Integer	Job run id
msgid	Element	Integer	Unique id for hte log message
messagelog	ENTRY		CREATE, DELETE, GET, GETLIST
id	Id	--	Id (primary key)

Table A-49 MPE

Name	Atom	Type	Description
mpejob	ENTRY		CREATE, DELETE, GET, GETLIST

Table A-50 MPEJobRun

Name	Atom	Type	Description
mpejobrun	ENTRY		CREATE, DELETE, GET, GETLIST

Table A-51 MPENode

Name	Atom	Type	Description
mpenode	ENTRY		CREATE, DELETE, GET, GETLIST

Table A-52 MVSAgentList

Name	Atom	Type	Description
mvsagentlist	ENTRY		CREATE, DELETE, GET, GETLIST

Table A-53 MVSNode

Name	Atom	Type	Description
mvsnode	ENTRY		CREATE, DELETE, GET, GETLIST

Table A-54 Node

Name	Atom	Type	Description
id	Element	int	Id (primary key) of Node **getId** **setId**
active	Element	String	Active? (Y/N) **getActive** **setActive** **hasValueActive**
connectionactive	Element	String	Connectin active? (Y/N) **getConnectionActive** **setConnectionActive** **hasValueConnectionActive**
name	Element	String	Name (length 64) **getName** **setName** **hasValueName**

Table A-54 Node (Continued)

Name	Atom	Type	Description
description	Element	String	Description (length 4000) **getDescription** **setDescription** **hasValueDescription**
type	Element	Short	Node type **getType** **setType** **hasValueType**
			(Continued)
ostype	Element	Short	Os type **getOstype** **setOstype** **hasValueOstype**
timediff	Element	Integer	Time difference between master and agent **getActivejobs** **setActivejobs** **hasValueTimediff**
activejobs	Element	Short	Number of jobs currently running on agent **getActivejobs** **setActivejobs** **hasValueActivejobs**
currentoutageend	Element	Date	IThe DateTime at which the current outage will end, or null if not currently in an outage period **getCurrentoutageend** **setCurrentoutageend** **hasValueCurrentoutageend**
nextoutagebegin	Element	Date	The DateTime at which the current outage will end, or null if not currently in an outage period. **getNextoutagebegin** **setNextoutagebegin** **hasValueNextoutagebegin**
timezone	Element	String	Time zone of Node **getTimezone** **setTimezone** **hasValueTImezone**
agentversion	Element	String	The version of the agent **getAgentversion** **setAgentversion** **hasValueAgentversion**
serviceid	Element	Integer	Service id **getServiceID** **setServiceID** **hasValueServiceID**

Table A-54 Node (Continued)

Name	Atom	Type	Description
agentload	Element	Integer	Current load of the agent getAgetnLoad setAgentLoad hasValueAgentLoad
lastusermodifiedtime	Element	Date	Last user modified time **getLastUserModifiedTime** **setLastUserModifiedTime** **hasValueLastUserModifiedTime**
id	Id	--	Id (primary key) of Node **getId** **setId**
name	Title	--	Name (length 64) **getName** **setName** **hasValueName**

Table A-55 NodeOutage

Name	Atom	Type	Description
calendarid	Element	Integer	Calendar id
connectionid	Element	Integer	Connection id
fromtimeasstring	Element	String	From time (String)
fromtime	Element	Date	From time (Date)
id	Element	int	Id (primary key)
state	Element	Short	State
untiltimeasstring	Element	String	Until time (String)
untiltime	Element	Date	Until time (Date)
lastusermodifiedtime	Element	Date	Last user modified time **getLastUserModifiedTime** **setLastUserModifiedTime**
nodeoutage	ENTRY		CREATE, DELETE, GET, GETLIST
id	Id	--	Id (primary key)

Table A-56 OS400Job

Name	Atom	Type	Description
pge1command	Element	String	Page 1 command **getPge1Command** **setPge1Command**
pge1jobname	Element	String	Page 1 job name
pge1jobdescription	Element	String	Page 1 job description
pge1jobdescriptionlib	Element	String	Page 1 job description library
pge1jobqueue	Element	String	Page 1 job queue
pge1jobqueuelib	Element	String	Page 1 job queue library
pge1jobpriority	Element	String	Page 1 job priority
pge1outputpriority	Element	String	Page 1 output priority
pge1printdevice	Element	String	Page 1 print device
pge1outputqueue	Element	String	Page 1 output queue
pge1outputqueuelib	Element	String	Page 1 output queue library
pge2user	Element	String	Page 2 user
pge2printtext	Element	String	Page 2 print text
pge2routedata	Element	String	Page 2 route data
			(Continued)
pge2reqdatacmd	Element	String	Page 2 request data command
pge3systemlib	Element	String	Page 3 system library
pge3currentlib	Element	String	Page 3 current library
pge3initiallib	Element	String	Page 3 initial library
pge3mloglevel	Element	String	Page 3 mlog level
pge3mlogseverity	Element	String	Page 3 mlog severity
pge3mlogtext	Element	String	Page 3 mlog text (String)
pge3progcommand	Element	String	Page 3 program command
pge3inqmsgreply	Element	String	Page 3 message reply
pge3holdjobqueue	Element	String	Page 3 hold job queue
pge3jobswitches	Element	String	Page 3 job switches
pge3displaysbmjob	Element	String	Page 3 display sbm job
pge4msgqueue	Element	String	Page 4 message queue
pge4msgqueuelib	Element	String	Page 4 message queue library
pge4sortsequence	Element	String	Page 4 short sequence
pge4sortsequencelib	Element	String	Page 4 short sequence library
pge4languageid	Element	String	Page 4 language id
pge4countryid	Element	String	Page 4 country id
pge4charsetid	Element	String	Page 4 character set id

Table A-56 OS400Job (Continued)

Name	Atom	Type	Description
pge4submittedfor	Element	String	Page 4 submitted for
pge4user	Element	String	Page 4 user
pge4number	Element	String	Page 4 number
pge4jobmqueuesize	Element	String	Page 4 job mqueue size
pgejobmqueuefullaction	Element	String	Page 4 job mqueue full action
pgecopyenvvariables	Element	String	Page 4 copy environment variables
pge4allowmultithreads	Element	String	Page 4allow multiple threads
pge4initialaspgroup	Element	String	Page 4 initial asp group
pge4spoolfileaction	Element	String	Page 4 spool file action
extendedinfo	Element	Date	Last user modified time **getLastUserModifiedTime** **setLastUserModifiedTime**
os400job	ENTRY		CREATE, DELETE, GET, GETLIST

Table A-57 OS400JobRun

Name	Atom	Type	Description
os400jobrun	ENTRY		CREATE, DELETE, GET, GETLIST

Table A-58 OS400Node

Name	Atom	Type	Description
machine	Element	String	Machine name (1024 chars)
joblimit	Element	Short	Job limit
runuser	Element	Integer	Runtime user
os400node	ENTRY		CREATE, DELETE, GET, GETLIST

Table A-59 OSJob

Name	Atom	Type	Description
os400job	ENTRY		CREATE, DELETE, GET, GETLIST

Table A-60 OSJobRun

Name	Atom	Type	Description
osjobrun	ENTRY		CREATE, DELETE, GET, GETLIST

Table A-61 OVMSAgentList

Name	Atom	Type	Description
ovmsagentlist	ENTRY		CREATE, DELETE, GET, GETLIST

Table A-62 OVMSBatch

Name	Atom	Type	Description
cmdjobname	Element	String	Cmd:job name (length 1024)
cmdjobqueue	Element	String	Cmd:job queue (length 1024)
			(Continued)
cmdWSDefault	Element	String	Cmd:WS default (length 1024)
cmdlogfile	Element	String	Cmd:log file
cmdlogfile	Element	String	Cmd:WS extent (length 1024)
cmdcli	Element	String	Cmd:Cli (length 1024)
cmdwsquota	Element	String	Cmd:WS quota (length 1024)
cmdjobargs	Element	String	Cmd:job arguments (length 1024)
ovmsbatch	ENTRY		CREATE, DELETE, GET, GETLIST

Table A-63 OVMSBatchJobRun

Name	Atom	Type	Description
cmdjobname	Element	String	Cmd:job name (length 1024)
cmdjobqueue	Element	String	Cmd:Job queue (length 1024)
cmdwsdefault	Element	String	Cmd:WS default (length 1024)
cmdlogfile	Element	String	Cmd:log file (length 1024)
cmdWSExtent	Element	String	Cmd:WS extent (length 1024)
cmdcli	Element	String	Cmd:Cli (length 1024)
cmdwsquota	Element	String	Cmd:WS quota (length 1024)
cmdjobargs	Element	String	Cmd:job arguments (length 1024)
ovmsbatchjobrun	ENTRY		CREATE, DELETE, GET, GETLIST

Table A-64 OVMSCommand

Name	Atom	Type	Description
ovmscommand	ENTRY		CREATE, DELETE, GET, GETLIST

Table A-65 OVMSCommandJobRun

Name	Atom	Type	Description
ovmscommandjobrun	ENTRY		CREATE, DELETE, GET, GETLIST

Table A-66 OVMSJob

Name	Atom	Type	Description
ovmsjob	ENTRY		CREATE, DELETE, GET, GETLIST

Table A-67 OVMSJobRun

Name	Atom	Type	Description
ovmsjobrun	ENTRY		CREATE, DELETE, GET, GETLIST

Table A-68 OVMSNode

Name	Atom	Type	Description
ovmsnode	ENTRY		CREATE, DELETE, GET, GETLIST

Table A-69 OwnerAgent

Name	Atom	Type	Description
id	Element	int	Id (primary key) of OwnerAgent **getId** **setId**
nodeid	Element	Integer	Node id
ownerid	Element	Integer	Owner id
owneragent	ENTRY		CREATE, DELETE, GET, GETLIST

Table A-69 OwnerAgent (Continued)

Name	Atom	Type	Description
id	Id	--	Id (primary key) of OwnerAgent **getId** **setId**

Table A-70 Queue

Name	Atom	Type	Description
id	Element	int	Id (primary key) of OwnerAgent **getId** **setId**
name	Element	String	Name (length 64) **getName** **setName** **hasValueName**
active	Element	String	Active? (Y/N) **getActive** **hasValueActive**
bump	Element	String	Bump? (Y/N) **getBump** **setBump** **hasValueBump**
bumpfrequency	Element	Short	Bump frequency **getBumpFrequency** **setBumpFrequency** **hasValueBumpFrequency**
bumppriority	Element	Short	Bump priority **getBumpPriority** **setBumpPriority** **hasValueBumpFrequency**
bumpqueue	Element	Integer	Bump queue **getBumpQueue** **setVBumpQueue** **hasValueBumpQueue**
description	Element	String	Description (length 4000) **getDescription** **setDescription** **hasValueDescription**
fromtimeasstring	Element	String	From time (String) **getFromTimeAsString** **setFromTimeAsString**
fromtime	Element	Date	From time (Date) **getFromTime** **setFromTime**
highpriority	Element	String	High priority? (Y/N) **getHighPriority** **setHighPriority** **hasValueHighPriority**
info	Element	String	Info **getInfo** **setInfo** **hasValueInfo**

Table A-70 Queue (Continued)

Name	Atom	Type	Description
numwaitingjobs	Element	Integer	Number of waiting jobs **getNumWaitingJobs** **setNumWaitingJobs** **hasValueNumWaitingJobs**
			(Continued)
numrunningjobs	Element	Integer	Number of running jobs **getNumRunningJobs** **setNumRunningJobs** **hasValueNumRunningJobs**
limit	Element	Short	Limit **getLimit** **setLimit** **hasValueLimit**
nicevalue	Element	Short	Nice value **getNiceValue** **setNiceValue** **hasValueNiceValue**
parentid	Element	Integer	Parent id **getParentId** **setParentId** **hasValueParentId**
fullpath		String	Full path (length 4000) **getFullPath** **setFullPath**
priority		Short	Priority **getPriority** **setPriority** **hasValuePriority**
untiltimeasstring	Element	String	Until time (String) **getUntilTime** **setUntilTime**
untiltime	Element	Date	Until time (Date) **getUntilTime** **setUntilTime**
lastusermodifiedtime	Element	Date	Last user modified time **getLastUserModifiedTime** **setLastUserModifiedTime** **hasValueLastUserModifiedTime**
availabel	Element	String	Available? (Y/N) **getAvailable** **setAvailable** **hasValueAvailable**
queue	ENTRY		CREATE, DELETE, GET, GETLIST

Table A-70 Queue (Continued)

Name	Atom	Type	Description
id	Id	--	Id (primary key) of Queue **getId** **setId**
name	Title	--	Name (length 64) **getName** **setName** **hasValueName**

Table A-71 QueuedJobRun

Name	Atom	Type	Description
id	Element	Integer	Id (primary key) of QueuedJobRun **getId** **setId**
queuetype	Element	String	Queue type **getQueueType** **setQueueType**
nodeid	Element	Integer	Node id **getNodID** **setNodeID**
nodename	Element	String	Node name **getNodeName** **setNodeName**
externalid	Element	String	External id **getExternalIDAsString** **setExternalIDAsString**
status	Element	Short	Status **getStatus** **setStatus**
priority	Element	Short	Priority **getPriority** **setPriority**
duration	Element	Integer	Duration **getDuration** **setDuration**
isexternaljob	Element	Boolean	Is external job? (True/False) **getIsExternalJob** **setIsExternalJob**
jobname	Element	String	Job name **getJobName** **setJobName**

Table A-71 QueuedJobRun (Continued)

Name	Atom	Type	Description
username	Element	String	User name **getUserName** **setUserName**
attributestring	Element	String	Attribute string **getAttributeString** **setAttributeString**
queuedjobrun	ENTRY		CREATE, DELETE, GET, GETLIST
id	Id	--	Id (primary key) of QueuedJobRun **getId** **setId**

Table A-72 QueueFilter

Name	Atom	Type	Description
id	Element	int	Id (primary key) of QueueFilter **getId** **setId**
exclude	Element	String	Exclude? (Y/N) **getExclude** **setExclude**
integer1	Element	Integer	Integer1 **getInteger1** **setInteger1**
integer2	Element	Integer	Integer2 **getInteger2** **setInteger2**
queuefilterinfo	Element	String	Que filter info **getQueueFilterInfo** **setQueueFilterInfo**
queueid	Element	Integer	Queue id **getQueueId** **setQueueId**
serviceid	Element	Integer	Service id **getServiceId** **setServiceId**
connectionid	Element	Integer	Connection id **getConnectionId** **setConnectionId**
string	Element	String	String (length 4000) **getString** **setString**

Table A-72 QueueFilter (Continued)

Name	Atom	Type	Description
type	Element	Short	Type **getType** **setType**
lastusermodifiedtime	Element	Date	Last user modified time **getLastUserModifiedTime** **setLastUSerModifiedTime**
queuefilter	ENTRY		CREATE, DELETE, GET, GETLIST
id	Id	--	Id (primary key) of QueueFilter **getId** **setId**

Table A-73 RemoteMasterNode

Name	Atom	Type	Description
Backupmaster	Element	String	Backup master (30 chars) **getBackupMaster** **setBackupMaster**
commport	Element	Short	Communication port **getCommPort** **setCommPort** **hasValueCommPort**
primarymaster	Element	String	Primary master (1024 chars) **getPrimaryMaster** **setPrimaryMaster**
machine	Element	String	Machine (length 1024) **getMachine** **setMachine** **hasValueMachine**
remotemasternode	ENTRY		CREATE, DELETE, GET, GETLIST

Table A-74 Resource

Name	Atom	Type	Description
id	Element	int	Id (primary key) of Resource **getId** **setId**
name	Element	String	Name **getName** **setName** **hasValueName**

Table A-74 Resource (Continued)

Name	Atom	Type	Description
description	Element	String	Description (length 4000) **getDescripiton** **setDescription**
limit	Element	Integer	Limit **getLimit** **setLimit**
inuse	Element	Integer	In use **getInuse** **setInuse**
active	Element	String	Active? (Y/N) **getActive** **setActive**
locked	Element	Integer	Locked **getLocked** **setLocked**
offline	Element	Integer	Offline **getOffline** **setOffline**
			(Continued)
ownerid	Element	Integer	Owner **getOwner** **setOwner**
publicflag	Element	String	Public flag? (Y/N) **getActive** **setActive**
lastusermodifiedtime	Element	Date	Last user modified time **getLastUserModifiedTime** **setLastUserModifiedTime**
resource	ENTRY		CREATE, DELETE, GET, GETLIST
id	Id	--	Id (primary key) of Resource **getId** **setId**
name	Title	--	Name **getName** **setName** **hasValueName**

Table A-75 ResourceJobJoin

Name	Atom	Type	Description
id	Element	int	Id (primary key) of ResourceJobJoin **getId** **setId**
jobid	Element	Integer	Job id **getJobId** **setJobId**
resourceid	Element	Integer	Resource id **getResourceId** **setResourceId**
used	Element	Integer	Used **getUsed** **setUsed**
resourcejobjoin	ENTRY		CREATE, DELETE, GET, GETLIST
id	Id	--	Id (primary key) of ResourceJobJoin **getId** **setId**

Table A-76 ResourceLock

Name	Atom	Type	Description
id	Element	int	Id (primary key) of ResourceJobJoin **getId** **setId**
jobrunid	Element	Integer	Job run id **getJobRunId** **setJobRunId**
resourceid	Element	Integer	Resource id **getResourceId** **setResourceId**
numlocked	Element	Integer	Num locked **getNumLocked** **setNumLocked**
resourcelock	ENTRY		CREATE, DELETE, GET, GETLIST
id	Id	--	Id (primary key) of ResourceLock **getId** **setId**

Table A-77 Schedule

Name	Atom	Type	Description
id	Element	int	Id (primary key) of Schedule **getId** **setId**
adhocgroups	Element	Integer	Ad hoc groups **getAdHocGropus** **setAdHocGroups**
batchcount	Element	Integer	Batch count **getBatchCount** **setBatchCount**
adhocjobs2	Element	Integer	Ad hoc jobs2 **getAdHocJobs2** **setAdHocJobs2**
carryover2	Element	Integer	Carry over2 **getCarryOver2** **setCarryOver2**
compilefinished	Element	Short	Compile finished **getCompileFinished** **setCompileFinished**
compiletime	Element	Short	Compile time **getCompileTime** **setCompileTime**
nextbatch	Element	Integer	Next batch **getNextBatch** **setNextBatch**
reruns2	Element	Integer	(Continued) Reruns2 **getReRuns2** **setReRuns2**
scheduledgroups	Element	Integer	Scheduled groups **getScheduledGroups** **setScheduledGroups**
scheduledjobs2	Element	Integer	Scheduled jobs2 **getScheduledJobs2** **setScheduledJobs2**
lastcompiletime	Element	Date	Last cimpile time **getLastCompileTime** **setLastCompileTime**
lastcompiletimeasstring	Element	String	Last compile time (String) **getLastCompileTimeAsString** **setLastCompileTimeAsString**
productiondateasstring	Element	String	Production date (String) **getProductionDateAsString** **setProductionDateAsString**

Table A-77 Schedule (Continued)

Name	Atom	Type	Description
productiondate	Element	Date	Production date (Date) **getProductionDate** **setProductionDate**
lastusermodifiedtime	Element	Date	Last user modified time **getLastUserModifiedTime** **setLastUserModifiedTime**
issecondarycachesynccompleted	Element	Integer	Is secondary cache sync completed **getIsSecondaryCacheSyncCompleted** **setIsSecondaryCacheSyncCompleted**
schedules	ENTRY		CREATE, DELETE, GET, GETLIST
id	Id	--	Id (primary key) of Schedule **getId** **setId**

Table A-78 SecurityPolicy

Name	Atom	Type	Description
id	Element	int	Id (primary key) of SecurityPolicy **getId** **setId**
name	Element	String	Name (length 64) **getName** **setName** **hasValueName**
description	Element	String	Description **getDescription** **setDescription**
catgeneral	Element	Integer	(Continued) Category:General **getCatGeneral** **setCatGeneral**
catsecurity	Element	Integer	Category:Security **getCatSecurity** **setCatSecurity**
catusers	Element	Integer	Category:Users **getCatUsers** **setCatUsers**
catclasses	Element	Integer	Category:Classes **getCatClasses** **setCatClasses**

Table A-78 SecurityPolicy (Continued)

Name	Atom	Type	Description
catconfig	Element	Integer	Category:Configuration **getCatConfig** **setCatConfig**
catconsole	Element	Integer	Category:Console **getCatConsole** **setCatConsole**
catworkgroups	Element	Integer	Category:Workgroups **getCatWorkGroups** **setCatWorkGroups**
catvariables	Element	Integer	Category:Variables **getCatVariables** **setCatVariables**
catjobs	Element	Integer	Category:Jobs **getCatJobs** **setCatJobs**
catjobcontrol	Element	Integer	Category:Job control **getCatJobControl** **setCatJobControl**
catcalendars	Element	Integer	Category:Calendars **getCatCalendars** **setCatCalendars**
catjobevents	Element	Integer	Category:Job events **getCatJobEvents** **setCatJobEvents**
catactions	Element	Integer	Category:Actions **getCatActions** **setCatActions**
catqueues	Element	Integer	Category:Queues **getCatQueues** **setCatQueues**
catlicenses	Element	Integer	Category:Licenses **getCatLicenses** **setCatLicenses**
catagentlists	Element	Integer	Category:AgentLists **getCatAgentLists** **setCatAgentLists**
catschedule	Element	Integer	(Continued) Category:Schedule **getCatSchedule** **setCatSchedule**
catsysevents	Element	Integer	Category:System Events **getCatSysEvents** **setCatSysEvents**

Table A-78 SecurityPolicy (Continued)

Name	Atom	Type	Description
catalerts	Element	Integer	Category:Alertss **getCatAlerts** **setCatAlerts**
catresources	Element	Integer	Category:Resources **getCatResources** **setCatResources**
catfaultmonitor	Element	Integer	Category:Fault monitor **getCatFaultMonitor** **setCatFaultMonitor**
catvarevents	Element	Integer	Category:Events **getCatEvents** **setCatEvents**
catfileevents	Element	Integer	Category:File Events **getCatFileEvents** **setCatFileEvents**
cat26	Element	Integer	Category:Cat26 **getCat26** **setCat26**
cat27	Element	Integer	Category:Cat27 **getCat27** **setCat27**
cat28	Element	Integer	Category:Cat28 **getCat28** **setCat28**
lastusermodifiedtime	Element	Date	Last user modified time **getLastUserModifiedTime** **setLastUserModifiedTime**
securitypolicy	ENTRY		CREATE, DELETE, GET, GETLIST
id	Id	--	Id (primary key) of SecurityPolicy **getId** **setId**
name	Title	--	Name (length 64) **getName** **setName** **hasValueName**

Table A-79 SecurityServiceJoin

Name	Atom	Type	Description
id	Element	int	Id (primary key) of SecuritySecurityJoin **getId** **setId**
securityid	Element	Integer	Secirity id **getSecurityId** **setSecurityId**
category	Element	Integer	Category **getCategory** **setCategory**
custombits	Element	Integer	CustomBits **getCustomBits** **setCustomBits**
serviceid	Element	Integer	Service id **getServiceID** **setServiceID**
securityservicejoin	ENTRY		CREATE, DELETE, GET, GETLIST
id	Id	--	Id (primary key) of SecuritySecurityJoin **getId** **setId**

Table A-80 Service

Name	Atom	Type	Description
id	Element	int	Id (primary key) of Service **getId** **setId**
name	Element	String	Name (length 20) **getName** **setName**
config	Element	String	Configuraiton **getConfig** **setConfig**
fullname	Element	String	Full name **getFullName** **setFullName**
guid	Element	String	GUID **getGUID** **setGUID**
prepjobonwaitop	Element	String	Prep job on wait op **getPropJobOnWaitOp** **setPrepJobOnWaitOp**

Table A-80 Service (Continued)

Name	Atom	Type	Description
type	Element	String	Type **getType** **setType**
version	Element	String	(Continued) Version getVersion setVersion
defaultconnectionid	Element	Integer	Default connection id **getDefaultConnectionID** **setDefaultConnectionID**
service	ENTRY		CREATE, DELETE, GET, GETLIST
id	Id	--	Id (primary key) of Service **getId** **setId**
name	Title	--	Name (length 20) **getName** **setName**

Table A-81 ServiceAction

Name	Atom	Type	Description
serviceid	Element	Integer	Id (primary key) of ServiceAction **getId** **setId**
connectionid	Element	Integer	Secirity id **getConnectionId** **setConnectionId**
serviceinfo	Element	String	Service info **getServiceInfo** **setServiceInfo**
serviceaction	ENTRY		CREATE, DELETE, GET, GETLIST

Table A-82 ServiceAgentList

Name	Atom	Type	Description
serviceid	Element	Integer	Service id (primary key) **getServiceId** **setServiceId** **hasValueServiceID**

Table A-82 ServiceAgentList (Continued)

Name	Atom	Type	Description
serviceagentlist	ENTRY		CREATE, DELETE, GET, GETLIST

Table A-83 ServiceDist

Name	Atom	Type	Description
id	Element	int	Id (primary key) of ServiceDist **getId** **setId**
serviceid	Element	Integer	Service id **getServiceID** **setServiceID**
servicepkg	Element	String	Service package **getServicepkg** **setServicepkg**
servicedist	ENTRY		CREATE, DELETE, GET, GETLIST
id	Id	--	Id (primary key) of ServiceDist **getId** **setId**

Table A-84 ServiceJob

Name	Atom	Type	Description
summaryonly	Element	String	Summary only **getSummaryOnlyAsString** **setSummaryOnlyAsString**
extendedinfo	Element	String	Extended info **getExtendedInfo** **setExtendedInfo**
servicejob	ENTRY		CREATE, DELETE, GET, GETLIST

Table A-85 ServiceJobRun

Name	Atom	Type	Description
servicejobrun	ENTRY		CREATE, DELETE, GET, GETLIST

Table A-86 ServiceNode

Name	Atom	Type	Description
joblimit	Element	Short	Job limit (applies to all except Email, TA and z/OS **getJobLimit** **setJobLimit**
machine	Element	String	Machine name (length 1024) **getMachine** **setMachine** **hasValueMachine**
runuser	Element	Integer	Runtime user **getRunUser** **setRunUser** **hasValueRunUser**
servicexml	Element	String	XML field describing the service **getServiceXml** **setServiceXml**
servicenode	ENTRY		CREATE, DELETE, GET, GETLIST

Table A-87 SNMPAction

Name	Atom	Type	Description
severity	Element	Short	Severity **getSeverityAsShort** **setSeverityAsShort**
community	Element	String	Community **getCommunity** **setCommunity**
message	Element	String	Message **getMessage** **setMessage**
itotype	Element	Short	ITO type **getITOType** **setITOType**
snmpaction	ENTRY		CREATE, DELETE, GET, GETLIST

Table A-88 Statistics

Name	Atom	Type	Description
id	Element	int	Id (primary key)
numscheduledjobs	Element	int	Number of scheduled jobs
numadhocjobs	Element	int	Number of adhoc jobs

Table A-88 Statistics (Continued)

Name	Atom	Type	Description
			(Continued)
numcarryoverjobs	Element	int	Number of carry over jobs
numjobsdone	Element	int	Number of jobs done
numjobscancelled	Element	int	Number of jobs cancelled
numrerunjobs	Element	int	Number of rerun jobs
numcarryovertocojobs	Element	int	Number of carry over to go jobs
jobstotal	Element	int	Jobs total
jobswait	Element	int	Jobs waiting
jobstogo	Element	int	Jobs to go
statistics	ENTRY		CREATE, DELETE, GET, GETLIST
id	Id	--	Id (primary key)

Table A-89 SystemActivityMessage

Name	Atom	Type	Description
id	Element	String	Id (primary key) of SystemActivityMessage **getId** **setId**
timestamp	Element	Date	Time stamp
polltimestamp	Element	Date	Polling time stamp
message	Element	String	System activity message
lastchangetime	Element	Date	Time when the API object was last changed
systemactivitymessag e	ENTRY		CREATE, DELETE, GET, GETLIST
id	Id	--	Id (primary key) of SystemActivityMessage **getId** **setId**

Table A-90 SystemValue

Name	Atom	Type	Description
id	Element	int	Id (primary key)
intvalue	Element	Integer	Integer value
strvalue	Element	String	String value
timevalue	Element	Date	Time value
			(Continued)

Table A-90 SystemValue (Continued)

Name	Atom	Type	Description
systemvalue	ENTRY		CREATE, DELETE, GET, GETLIST
id	Id	--	Id (primary key)

Table A-91 UnixAgentList

Name	Atom	Type	Description
unixagentlist	ENTRY		CREATE, DELETE, GET, GETLIST

Table A-92 UnixNode

Name	Atom	Type	Description
unixnode	ENTRY		CREATE, DELETE, GET, GETLIST

Table A-93 UserJoin

Name	Atom	Type	Description
id	Element	int	Id (primary key)
userid	Element	Integer	User id
nodeid	Element	Integer	Node id
equivalentuserid	Element	Integer	Equivalent user id
userjoin	ENTRY		CREATE, DELETE, GET, GETLIST
id	Id	--	Id (primary key)

Table A-94 UserMessage

Name	Atom	Type	Description
id	Element	int	Id (primary key)
timestamp	Element	String	Time stamp
msgtimestamp	Element	Date	Message time stamp
type	Element	Short	Message type
			(Continued)
fromusersession	Element	Integer	Message sender's session id
tousersession	Element	Integer	Message receiver's session id

Table A-94 UserMessage (Continued)

Name	Atom	Type	Description
readbyuser	Element	String	Flag if the message has been read by the user (Y or N)
message	Element	String	Message text
usermessage	ENTRY		CREATE, DELETE, GET, GETLIST
id	Id	--	Id (primary key)

Table A-95 Users

Name	Atom	Type	Description
id	Element	int	Id (primary key)
name	Element	String	Name (length 256)
fullname	Element	String	Full name (length 1024)
domain	Element	String	Domain (length 1024)
password	Element	String	Password (length 1024)
description	Element	String	Description (length 4000)
externid	Element	String	External id (length 20)
secirity policyid	Element	Integer	Security policy id
languageid	Element	Integer	Language id
superuser	Element	String	Super user (length 1)
phonenumber	Element	String	Phone number (length 40)
pagernumber	Element	String	Pager number (length 40)
emailaddress	Element	String	Email address (length 1024)
emailtype	Element	Short	Email type
wingroup	Element	String	Windows group (length 1)
oracleappspassword	Element	String	Oracle Applications password (length 144)
lastusermodifiedtime	Element	Date	Last user modified time
tempaccount	Element	String	Temporary account (length 1)
allagents	Element	String	All agents? (length 1)
profile	Element	String	Profile (clob 6M)
users	ENTRY		CREATE, DELETE, GET, GETLIST
			(Continued)
id	Id	--	Id (primary key)
name	Title	--	Name (length 2565)

Table A-96 UserService

Name	Atom	Type	Description
id	Element	int	Id (primary key)
password	Element	String	Password (length 144)
serviceid	Element	Integer	Service id (length 20)
userid	Element	Integer	User id
userservice	ENTRY		CREATE, DELETE, GET, GETLIST
id	Id	--	Id (primary key)
password	Title	--	Password (length 144)

Table A-97 UserSession

Name	Atom	Type	Description
id	Element	int	Id (primary key)
computer	Element	String	Computer (length 256)
sessionidstring	Element	String	Session id string
userid	Element	Integer	User id
impersonateuserid	Element	Integer	Client type
starttime	Element	Date	Start time
endtime	Element	Date	End time
clientmanagerid	Element	String	Client manager id (length 128)
clienttype	Element	Integer	Client type
usersession	ENTRY		CREATE, DELETE, GET, GETLIST
id	Id	--	Id (primary key)
computer	Title	--	Computer (length 256)

Table A-98 Variable

Name	Atom	Type	Description
id	Element	int	Id (primary key)
name	Element	String	Name (length 64)
calc	Element	String	Calc (length 1)
description	Element	String	Description (length 4000)
innervalue	Element	String	Inner value (length 4000)
lastusermodifiedtime	Element	Date	Last user modified time
lastvalue	Element	String	Last value (length 4000)
offsets	Element	String	Offsets
ownerid	Element	Integer	Owner id

Table A-98 Variable (Continued)

Name	Atom	Type	Description
pub	Element	String	Pub (length 1)
publish	Element	String	Publish (length 1)
readonly	Element	String	Read only. (length 1)
startcalendar	Element	Integer	Start calendar
startdateasstring	Element	String	Start date (String)
startdate	Element	Date	Start date (Date)
starttype	Element	String	Start type (length 20)
type	Element	Short	Type
variable	ENTRY		CREATE, DELETE, GET, GETLIST
id	Id	--	Id (primary key)
name	Title	--	Name (length 64)

Table A-99 VariableAction

Name	Atom	Type	Description
updatetype	Element	Short	Update action type
varid	Element	Integer	Variable to update
value	Element	String	Value (length 4000) to set the variable to
masterid	Element	Integer	The Master Nod ID in which ithe variable lives (either local master or remote master)
timing type	Element	Short	Timing type
variableaction	ENTRY		CREATE, DELETE, GET, GETLIST

Table A-100 VariableDependency

Name	Atom	Type	Description
connectionid	Element	Integer	Connection id
id	Element	int	Id (primary key)
jobid	Element	Integer	Job id
status	Element	Short	Status
logic	Element	Short	Dependency logic
operator	Element	Short	Operator
variableid	Element	String	Variable id
varvalue	Element	String	Variable value (length 4000)
variabledependency	ENTRY		CREATE, DELETE, GET, GETLIST

Table A-101 WindowsAgentList

Name	Atom	Type	Description
windowsagentlist	ENTRY		CREATE, DELETE, GET, GETLIST

Table A-102 WindowsNode

Name	Atom	Type	Description
windowsnode	ENTRY		CREATE, DELETE, GET, GETLIST

Table A-103 WorkGroup

Name	Atom	Type	Description
id	Element	int	Id (primary key)
name	Element	String	Name (length 256)
description	Element	String	Description (length 4000)
externid	Element	String	External id (length 20)
parentid	Element	Integer	Parent id
ownerid	Element	Integer	Owner
allagents	Element	String	All agents (length 1)
			(Continued)
lastusermodifiedtime	Element	Date	Last user modified time
workgroup	ENTRY		CREATE, DELETE, GET, GETLIST
id	Id	--	Id (primary key)
name	Title	--	Name (length 256)

Table A-104 WorkGroupUserJoin

Name	Atom	Type	Description
id	Element	int	Id (primary key)
workgroupid	Element	Integer	Work group
ownerid	Element	Integer	Owner
exclude	Element	String	Exclude (length 1)
usersecurityids	Element	String	User security policy ids
groupsecurityids	Element	String	Group security policy ids
workgroupuserjoin	ENTRY		CREATE, DELETE, GET, GETLIST
id	Id	--	Id (primary key)

Table A-105 ZOSGatewayNode

Name	Atom	Type	Description
encrypt	Element	String	Encrypt (length 1) **getEncrypt** **setEncrypt**
gatewayhost	Element	String	Gateway host (length 255)
gatewayport	Element	Short	Gateway port
machine	Element	String	Machine (length 1024)
zosgatewaynode	ENTRY		CREATE, DELETE, GET, GETLIST

Appendix

Overview

The following table lists REST API Objects and the associated REST API Methods. Note that a TES task or functionality may be realized in more than one way depending on the level of abstraction and implementation.

Examples are provided to illustrate key TES concepts.

Table B-1 REST API Objects and Associated REST API Methods

REST API Object	REST API Method
AgentList or AgentListJoin	getAllAgentsByAgentListId
Alert	get getList update
BackupServerNode	create update
BusinessView	isGroupBusinessViewExists nonAsyncCreateGroupView autoArrange
Calendar	getJobsUsedByCalendar compileCalendars compileSingleCalendar copy getWhereUsedByIDS getWhereUsedByCalendars
CalendarYear	getForecast getForecastCondition
CmdLine	tescmd
Events	copy hold release
FiscalCalendar	getStartFiscalDate getFiscalCalendar
GeneralInfo	getAllGeneralInfo

Table B-1 REST API Objects and Associated REST API Methods (Continued)

REST API Object	REST API Method
Image	putImagesLocally replaceImage replaceImage addImage
Job	insert replace enable disable copy nonasynccopy getTopLevelInheritedParent getDependencyParentChain generateInfoReports getJobSuccessorsList
JobOutput	getOutputContent getOutputUrl getJobOutputRaw
JobRun	getList assignStatus cance cancelall cancelpending hold insert override overrideDependency release remove rerun rerunstep getJobRunResources getJobRunDependencies generateInfoReports getVariableMenuData getJobRunPredecessorsList getJobRunSuccessorsList getJobRunStatistics
JobRunStatus	update
MasterLicense	getLicense

Table B-1 REST API Objects and Associated REST API Methods (Continued)

REST API Object	REST API Method
MasterNode	tesmcmd pauseScheduler resumeScheduler startTodaysSchedule resumeQueues pauseQueues stopMaster createSchedule checkMasterConnectionStatus update
Node	reconnect
ObjectRequest	get getList create copy enable update delete modifiedObjects
OwnerAgent	getOwnerAgentList
OwnerList	getOwnersList
QueueFilter	get getList create update
Queue	getList enable disable getQueueJobList cancel hold release getChildrenQueueIds copy
Report	eventHistoryReport scheduleSummaryReport dependenciesReport jobLastStatusReport
ResourceLock	release
Schedules	accept reverttoforecast remove
SecurityPolicy	getUserSecurityPolicyBitString

Table B-1 REST API Objects and Associated REST API Methods (Continued)

REST API Object	REST API Method
ServiceAction	get getList create update
ServiceEvent	get getList create update delete hold release getConnectionInfo
ServiceJob	getList create update delete copy
ServiceJobRun	get getList update
ServiceNode	get getList create update enable testConnection
SystemValue	update
TESWebService	request
UserJoin	getRunTimeUserList addToAllSUsers
UsersRequest	copy getAuthenticatedUser getCurrentUserSession loginTransporter getImpersonatedUser impersonate endImpersonation updateUserSecurityPolicy
WorkGroup	get getList copy
WorkGroupUserJoin	getMembersList

Appendix

Overview

There are many ways to invoke the RESTful services available from Enterprise Scheduler. In the examples below, the services are called by a simple Java client using the `HttpURLConnection` class. In addition to this method, you can also use a wide variety of third party frameworks such as the Spring Framework RestTemplate or the Apache CXF Framework.

REST API From Browser

Before writing code to invoke the REST API, one could first browse the services available via a browser. In a live Enterprise Scheduler environment, the URL where the API can be reached is as follows:

For example:

Figure C-1 REST API browser view

Clicking on the links issues a "GET" request to the API. A "POST" request to the API can also be issued from the browser by using the "Manual Commands (Post)" link.

Figure C-2 REST API Post Screen

REST API Security Notes

The calls to the REST API are subjected to the same security restrictions as the same user accessing Scheduler UI. In Code Example 1 below, a call is issued to get all of the available jobs. The list of available jobs returned is determined by the username used in the API call.

Java Client REST API Examples

The following Java client issues a GET request to the REST API. This is the equivalent of clicking on the ApiObject link as described in the REST API From Browser section. This example retrieves all of the jobs currently defined in the Scheduler environment. The username and password pair is Base64 encoded and passed to the server as the "Authorization" property.

An XML document containing a list of jobs is returned from this call.

Code Example 1 - GET Request

```
public static void postRequest() throws Exception
{
URL url = new URL("http://www.companyscheduler.com:8080/api/tes-6.2/post");
HttpURLConnection conn = (HttpURLConnection) url.openConnection();
conn.setRequestMethod("GET");
conn.setDoInput(true);
conn.setDoOutput(true);

String userNamePassword = "myusername:mypassword";
userNamePassword =
            new
String(org.apache.commons.codec.binary.Base64.encodeBase64(userNamePassword
                .getBytes()));
    conn.setRequestProperty("Authorization", userNamePassword);

    conn.connect();
    BufferedReader reader = new BufferedReader(new InputStreamReader(
                conn.getInputStream()));
    String resp = "";
    String next = null;
    while ((next = reader.readLine()) != null)
```

```
        resp += next;
    System.out.println(resp);
    }
```

Code Example 2 shows a POST request issued to the TES REST API. The URL for issuing a POST request is always the same: http://<hostname>:<port>/api/<DSP Name>/ post.

In the post request, the command to be executed is sent in an URL encoded string. In this particular example, a POST request is sent to insert a job into the schedule. The <id> is the id of the job. Other parameters include <startdate> - the requested runtime for the job; <vars> - local job variable overrides; <params> - local job parameter overrides; and <deps> - the Y/N value for whether or not to override the job's dependencies.

An XML document acknowledging the job insert is returned.

Code Example 2 - POST Request

```
public static void postRequest() throws Exception
{
  URL url = new URL("http://www.companyscheduler.com:8080/api/tes-6.2/post");
  HttpURLConnection conn = (HttpURLConnection) url.openConnection();
  conn.setRequestMethod("POST");
  conn.setDoInput(true);
  conn.setDoOutput(true);

  conn.setRequestProperty("Content-Type","application/x-www-form-urlencoded");

  String userNamePassword = "myusername:mypassword";
  userNamePassword =
            new
String(org.apache.commons.codec.binary.Base64.encodeBase64(userNamePassword
                .getBytes()));
  conn.setRequestProperty("Authorization", userNamePassword);

  String postCommand = "<?xml version=\"1.0\" encoding=\"UTF-8\" standalone=\"yes\" ?>"
                + "<atom:entry xmlns:atom=\"http://purl.org/atom/ns#\">"
+ "<atom:id>1</atom:id><atom:title>api</atom:title>"
+ "<Job.insert>"
+ "<id>2</id>"
+ "<startdate>20110812</startdate>"
+ "<vars></vars>"
+ "<params></params>"
+ "<deps>N</deps>"
+ "</Job.insert>"
+ "</atom:entry>";

  String payload = "data="+ java.net.URLEncoder.encode(postCommand, "ISO-8859-1");

  conn.setRequestProperty("Content-Length", Integer.toString(payload.getBytes().length));
  conn.setFixedLengthStreamingMode(payload.getBytes().length);

  DataOutputStream out = new DataOutputStream(conn.getOutputStream());
  out.writeBytes(payload);
  out.flush();

  BufferedReader reader = new BufferedReader(new InputStreamReader(
                conn.getInputStream()));
  String resp = "";
  String next = null;
  while ((next = reader.readLine()) != null)
```

```
    resp += next;
  System.out.println(resp);
}
```

Session Management

In both previous code examples, the calls establish new sessions on the server. For typical applications that make repeated calls to the REST API, the best practice is to reuse the established sessions so that the server does not create excessive number of active sessions, which eventually could cause it to run out of memory.

Code Example 3 is an extension of Code Example 1 showing the usages of a cookie for session management.

Code Example 3 - Use Session Cookie

```
public static void tesGetRequestWithSession() throws Exception
{
String sessionID = null;

for (int i=0; i<10; i++)
{
    URL url = new URL(
            "http://www.mycompanyscheduler.com:8080/api/tes-6.2/Job.getList");

    HttpURLConnection conn = (HttpURLConnection) url.openConnection();
    conn.setRequestMethod("GET");
    conn.setDoInput(true);
    conn.setDoOutput(true);

    if (sessionID == null)
    {
      String userNamePassword = " myusername:mypassword ";
      userNamePassword =
            new
String(org.apache.commons.codec.binary.Base64.encodeBase64(userNamePassword
                .getBytes()));
      conn.setRequestProperty("Authorization", userNamePassword);
    }
    else
    {
      conn.setRequestProperty("Cookie", sessionID);
    }

    conn.connect();
    BufferedReader reader = new BufferedReader(new InputStreamReader(
                conn.getInputStream()));
    String resp = "";
    String next = null;
    while ((next = reader.readLine()) != null)
       resp += next;
    System.out.println(resp);

    //extract cookie
    String setCookies = conn.getHeaderField("Set-Cookie");
    if (setCookies != null && sessionID == null)
    {
      String cookies[] = setCookies.split(";");
      if (cookies.length > 0)
        sessionID = cookies[0];
```

```
    }
  }
}
```

Execute a Query with Conditions

In Code Example 1, a `GET` request was issued to get a list of all jobs. The `GET` request can accept additional parameters such that the list of jobs returned can be filtered further.

If one needs to get a list of jobs that match a specific name pattern, the `GET` request URL can be constructed as follows:

```
URL url = new URL("http://www.mycompanyscheduler.com:8080/api/tes6.2/Job.getList?query=(Job.name LIKE '%name%')")
```

In this case a where clause (`Job.name LIKE '%name%'`) is sent. The `where` clause must be URL encoded. Similarly, other queries using other field names in the Job object can be constructed.

The same also be achieved using a `POST` request. The `POST` payload is below. In addition to the **queryCondition**, using the `POST` one could also specify columns needed.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<atom:entry xmlns:atom="http://purl.org/atom/ns#">
    <atom:id>1</atom:id>
    <atom:title>api</atom:title>
    <Job.getList>
    <selectColumns>
        id,ownerid,parentid,parentname,runtimeusername
    </selectColumns>
    <queryCondition>
        (Job.name LIKE '%name%')
</queryCondition>
    </Job.getList>
</atom:entry>
```