



## **Opswise Controller 5.2.0**

### **Help and Support**

© 2014 by Stonebranch, Inc. All Rights Reserved.

1. Troubleshooting	3
1.1 Problem Resolution	4
1.1.1 Error in your SQL syntax	5
1.1.2 Maximum open cursors have been exceeded	6
1.1.3 Out-of-Range Value during Database Initialization	7
1.1.4 JDBC Connections Time Out	8
1.1.5 Processes Will Not Start Automatically (Debian Linux)	9
1.1.6 Windows install fails with 'Service marked for deletion'	10
1.1.7 Error when Starting Controller or Message Hub	11
1.1.8 Cannot launch a task	12
1.1.9 VBScript stuck in "Running" state	13
1.1.10 My Opswise Controller License has Expired	14
1.1.11 Packet for query is too large	15
1.1.12 Invalid Login Credentials for Refreshing Target Agents	16
1.2 Error Messages	17
2. Instructional Videos	19
3. Tutorials	20
3.1 Tutorial - Creating and Manually Launching a Simple Task	22
3.2 Tutorial - Running a Windows Task	24
3.3 Tutorial - Launching a Task Automatically Using a Simple Time Trigger	28
3.4 Tutorial - Launching a Task Every Monday Except Holidays	30
3.5 Tutorial - Launching a Task Every Two Hours During Workday	37
3.6 Tutorial - Launching an Email Task Based on a File Monitor	39
3.7 Tutorial - Creating a Simple Workflow	44
3.8 Tutorial - Running a Workflow with a Conditional Path	49
3.9 Tutorial - Running a Workflow with Skipped Criteria	62
3.10 Tutorial - Using Variables in a Simple Task	65
3.11 Tutorial - Using Variables in a Workflow	68
3.12 Tutorial - Creating Custom Days and Periods	69
3.13 Tutorial - Generating Forecast Data	76
3.14 Tutorial - Finding and Inserting Tasks in an Active Workflow	82
3.15 Tutorial - Skipping, Unskipping, and Showing-Hiding Skipped Task Instances	91
3.16 Tutorial - Launching Tasks at a Future Date and Time	96
3.17 Tutorial - Launching Tasks Using a Cron Trigger	97
3.18 Tutorial - Launching an Email Task Based on a Task Monitor	98
3.19 Tutorial - Force Finishing, Force Finish-Cancelling, and Cancelling a Task	108
3.20 Tutorial - Selecting Widgets for the Home Page	111
3.21 Tutorial - Creating a Gauge	113
3.22 Tutorial - Adding Options to the Navigation Pane	115
3.23 Tutorial - Creating Business Services	118
3.24 Tutorial - Assigning Records to Business Services	119
3.25 Tutorial - Taking Advantage of Business Services	120
3.26 Tutorial - Creating a Report Based on Business Services	121
3.27 Tutorial - Viewing Activity by Business Service	122
3.28 Tutorial - Creating a Report	123
3.29 Tutorial - Scheduling and Distributing Reports	125
3.30 Tutorial - Setting Up a Virtual Resource	126
3.31 Tutorial - Accessing Task Instance Details	128
3.32 Tutorial - Aborting a Process Launched by a Task	134
3.33 Tutorial - Creating Users and Assigning Permissions	137
3.34 Tutorial - Creating Security Groups and Assigning Permission	141

# Troubleshooting

## Troubleshooting

Troubleshooting information is categorized into two areas:

- [Problem Resolution](#)
- [Error Messages](#)

## Problem Resolution

- Problem Resolution
  - Database
  - Installation
  - Operations

## Problem Resolution

This page provides links to problems, and their solutions, that you might encounter with Opswise Controller.

### Database

- Error in your SQL syntax
- Maximum open cursors have been exceeded
- Out-of-Range Value during Database Initialization
- JDBC Connections Time Out

### Installation

- Processes Will Not Start Automatically (Debian Linux)
- Windows install fails with 'Service marked for deletion'
- Error when Starting Controller or Message Hub

### Operations

- Cannot launch a task
- VBScript stuck in "Running" state
- My Opswise Controller License has Expired
- Packet for query is too large
- Invalid Login Credentials for Refreshing Target Agents

## Error in your SQL syntax

### Problem

When you execute an SQL task that includes multiple SQL commands, the following error message (for example) may display:

```
INSERT INTO opwise_demo (name, value) values ('A', 'F');  
INSERT INTO opwise_demo (name, value) values ('B', 'S');  
INSERT INTO opwise_demo (name, value) values ('C', 'F');
```

### Solution

Multiple queries, by default, are disabled for MySQL. To enable multiple queries, append the following string to the Connection URL field in the [Database Connection](#) resource definition:

```
?allowMultiQueries=true
```

The following example is a URL connection string for a MySQL Database Connection resource definition:

```
jdbc:mysql://localhost:3306/opwise?allowMultiQueries=true
```

## Maximum open cursors have been exceeded

### Problem

During large imports on Oracle, you could receive following error message:

```
ORA-01000: maximum open cursors exceeded
```

(The cursors are used only during the import; they then are closed.)

Issue the following **sql\*plus** utility command to check the current value for maximum open cursors:

```
show parameter open_cursors
```

A listing similar to the following will display:

```
SQL> show parameter open_cursors;

NAME                                TYPE        VALUE
-----                                -
open_cursors                        integer     1000
```

### Solution

An **open\_cursors** value of 1000 should be sufficient for all large imports.

You can temporarily set the **open\_cursors** value with the following SQL:

```
alter system set open_cursors=1000
```

To make a permanent change, you must set the **open\_cursors** value in the initialization parameters file.

## Out-of-Range Value during Database Initialization

### Problem

During the database initialization performed on initial start-up, you could receive the following message:

```
The conversion of a varchar data type to a datetime data type of the value is out of range.
```

The problem likely is that the database was created in SQL SERVER Management Studio with a user that has other than English as the default language.

### Solution

Verify the installed default language and set the language to U.S. English.

To check what default language a server has installed, use the following SQL command:

```
sp_configure 'default language'
```

If the resulting value is not 0, the default language is not U.S. English. Run the following SQL command to find the installed default language setting and date format used:

```
select name ,alias, dateformat
from syslanguages
where langid =
(select value from master..sysconfigures
where comment = 'default language')
```

To set the default language to U.S. English, use the following SQL statements:

```
sp_configure 'default language', 0
reconfigure with override
```

For further details, refer to this [Microsoft Support](#) page.

## JDBC Connections Time Out

### Problem

JDBC connections from Linux to MS SQL Server 2008 R2/Windows 2008 R2 time out after 40 seconds causing SQL/Stored Procedure Tasks that take longer than 40 seconds to fail with the following exception:

```

2014-09-22-14:51:37:034 -0400 ERROR
[Ops.General.15.EP.SqlHandler.ecd8ab62183f4b9dbf32d3ea4ad0a126.74b824ad1ca84142a40d3ec1f84d4d2b.0]
SQLServerException - Connection reset
com.microsoft.sqlserver.jdbc.SQLServerException: Connection reset
    at com.microsoft.sqlserver.jdbc.SQLServerConnection.terminate(SQLServerConnection.java:1667)
    at com.microsoft.sqlserver.jdbc.SQLServerConnection.terminate(SQLServerConnection.java:1654)
    at com.microsoft.sqlserver.jdbc.TDSChannel.read(IOBuffer.java:1789)
    at com.microsoft.sqlserver.jdbc.TDSReader.readPacket(IOBuffer.java:4838)
    at com.microsoft.sqlserver.jdbc.TDSCommand.startResponse(IOBuffer.java:6150)
    at
com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.doExecutePreparedStatement(SQLServerPreparedStatement
at
com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement$PrepStmtExecCmd.doExecute(SQLServerPreparedStatement
at com.microsoft.sqlserver.jdbc.TDSCommand.execute(IOBuffer.java:5696)
    at
com.microsoft.sqlserver.jdbc.SQLServerConnection.executeCommand(SQLServerConnection.java:1715)
    at
com.microsoft.sqlserver.jdbc.SQLServerStatement.executeCommand(SQLServerStatement.java:180)
    at
com.microsoft.sqlserver.jdbc.SQLServerStatement.executeStatement(SQLServerStatement.java:155)
    at
com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.execute(SQLServerPreparedStatement.java:332)
    at
com.stonebranch.opwise.server.events.SqlEventHandler.storedProc(SqlEventHandler.java:266)

```

### Resolution

To disable TCP Chimney Offload, follow these steps:

<b>Step 1</b>	Use administrative credentials to open a command prompt.
<b>Step 2</b>	At the command prompt, enter the following command: netsh int tcp set global chimney=disabled
<b>Step 3</b>	Press ENTER:.

For additional information, see:

<http://support.microsoft.com/kb/951037>



## Processes Will Not Start Automatically (Debian Linux)

### Problem

For Debian Linux environment: Outboard 5.1.0 processes will not start automatically at boot time.

Debian Linux does not provide the **chkconfig** command and therefore cannot work with the runlevels specified in the opsagent, opstransport, opsmgshub scripts provided in `/etc/init.d`.

***This is a known problem; we are working on a solution.***

## Windows install fails with 'Service marked for deletion'

### Problem

In a Windows environment, for an upgrade installation of the Outboard 5.1.0 components, the install fails with the following message:

```
'This Service is marked for deletion'
```

### Solution

Before running the install or upgrade process, make sure you first exit the Windows Services Console. If you do not, you likely will get this error.

If this occurs, exit the Windows Services application and re-run the install.

## Error when Starting Controller or Message Hub

### Problem

Upon starting the Opswise Controller or the 5.1.0 Message Hub server, the `opswise.log` shows ERR:

```
SQLSTATE: HY000, SQLERR: 1040, ERRMSG: [unixODBC][MySQL][ODBC 5.1 Driver]Too many connections
```

### Solution

You must set additional connections in your database server:

```
MySQL - /etc/my.conf - max_connections=500 (default is either 100 or 150)
```

## Cannot launch a task

### Problem

You cannot launch a task.

The problem may be with your credentials. Check the stderr for the following message:

```
ops_suexec: Not enough privileges. Check SUID bit and binary owner.
```

### Solution

If the error message is present, issue the following commands as **root** in the `$WMS_HOME/bin` directory:

```
chown root ops_suexec  
chmod 4755 ops_suexec
```

## VBScript stuck in "Running" state

### Problem

By default, Windows uses a GUI-based VBScript interpreter (`wscript.exe`). With this interpreter, if your script tries to display an error message that requires a user-response (for example, **Click OK**), you will never see the dialog box. The script therefore gets stuck in the "Running" state.

### Solution


To avoid this, we recommend you use the console version of the VBScript interpreter (`cscript.exe`). To do so, specify `cscript.exe` before the script name in a task definition, as shown in the following example:

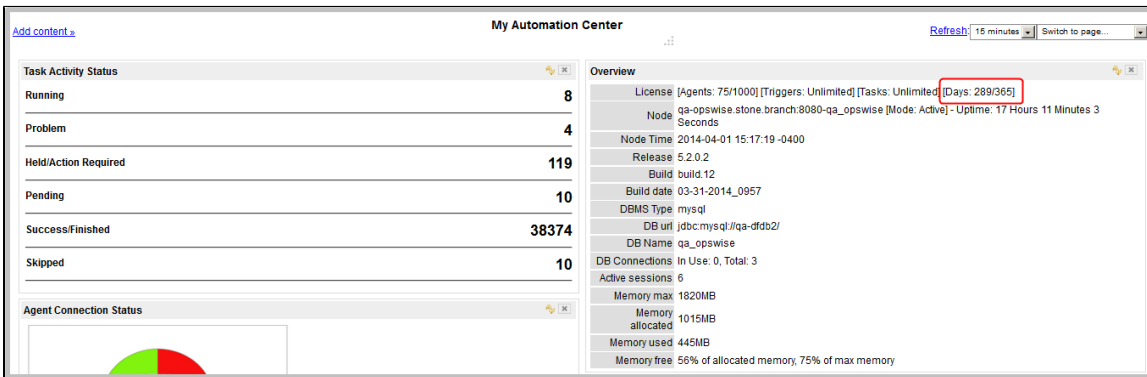
```
"cscript.exe C:\Work\script.vbs".
```

## My Opswise Controller License has Expired

If your Opswise Controller license has expired, you will not be able to run any tasks. If you attempt to run a task under an expired license, the Controller will place the task in the Defined state.

To see if your license has expired, check the Overview gauge on your Opswise Controller user interface Home Page to see how many days are left on your license:

1. Click the Home  icon.
2. In the Overview gauge, check the number of days left on your license:



The screenshot shows the 'My Automation Center' interface. On the left, there is a 'Task Activity Status' table with the following data:

Task Activity Status	Count
Running	8
Problem	4
Held/Action Required	119
Pending	10
Success/Finished	38374
Skipped	10

Below the table is an 'Agent Connection Status' gauge showing a green and red semi-circle. On the right, the 'Overview' section displays system metrics. The 'License' field is highlighted with a red box and shows '[Days: 289/365]'. Other metrics include Node Time, Release, Build date, DBMS Type, DB Name, DB Connections, and Memory usage.

The **Days: ##** field indicates the current day of your current license and the total number of days in the license. If the numbers are identical, your license has expired.

To otherwise verify that your license has expired, check the log file.

If your license has expired, the following two messages should appear in the log at the midnight roll-over:

```
2013-01-07-00:00:00:006    WARN [Ops.Timer.Forecast_Refresh.0] License Violation: Number of Days has
exceeded # suspending system
2013-01-07-00:00:00:006    INFO [Ops.Timer.Forecast_Refresh.0] Pausing the server.
```

Additionally, this message should appear in the log if you try to run a task under an expired license:

```
2013-01-07-09:32:27:728    INFO [Ops.Available.2367.0] System paused, waiting for resume
```

## Packet for query is too large

### Problem

During operations, the following message may appear in the Opswise Controller log:

```
Packet for query is too large (1084852 > 1048576).
```

### Solution

Change this value on the database server by setting the MySQL **max\_allowed\_packet** configuration variable.

For detailed information about this variable, refer to the [MySQL reference manual](#).

## Invalid Login Credentials for Refreshing Target Agents

### Problem

An error occurs when you click **Refresh Target Agents** on a Promotion Target record and you are using invalid login credentials for the target Opswise Controller instance.

The user interface on source machine will show the following error:

```
GET http://NN.NNN.NN.N:8080/opswise/resources/agents/list returned a response status of 401
Unauthorized
```

The the Controller log on the source machine will show the following error:

```
ERROR [http-8080-10] com.sun.jersey.api.client.UniformInterfaceException:
GET http://opswise/resources/agents/list returned a response status of 401 Unauthorized"
```

The target machine will return the following error:

```
ERROR [http--#] *** ERROR *** Login using Basic Authentication failed for:
```

### Solution

Update the Promotion Target record with valid login credentials and try the promotion again.



## Error Messages

### Error Messages

This page identifies error messages (in alphabetical order) that you may receive for Opswise Controller.

For each error, there is a link to the cause problem, and its solution, in [Problem Resolution](#).

```
GET http://NN.NNN.NN.N:8080/opswise/resources/agents/list returned a response status of 401
Unauthorized
```

(in user interface on source machine)

```
ERROR [http-8080-10] com.sun.jersey.api.client.UniformInterfaceException:
GET http://opswise/resources/agents/list returned a response status of 401 Unauthorized"
```

(in Opswise log on source machine)

```
ERROR [http--#] *** ERROR *** Login using Basic Authentication failed for:
```

(on Target machine)

See [Invalid Login Credentials for Refreshing Target Agents](#).

```
INSERT INTO opswise_demo (name, value) values ('A', 'F');
```

See [Error in your SQL syntax](#).

```
ops_suexec___Not enough privileges. Check SUID bit and binary owner
```

See [Cannot launch a task](#).

```
ORA-01000: maximum open cursors exceeded
```

See [Maximum open cursors have been exceeded](#).

```
Packet for query is too large (1084852 > 1048576
```

See [Packet for query is too large](#).

```
SQLSTATE: HY000, SQLERR. 1040, ERRMSG. unixODBC MySQL ODBC 5.1 Driver Too many connections
```

See [Error when Starting Controller or Message Hub](#).

```
The conversion of a varchar data type to a datetime data type resulted in an out-of-range value.
```

See [Out-of-Range Value during Database Initialization](#).

```
'This Service is marked for deletion'
```

See [Windows install fails with 'Service marked for deletion'](#).

# Instructional Videos

## Introduction

Stonebranch provides a set of videos that instruct you on how to perform various tasks in Opswise Automation Center. You can access the videos two ways.

<b>Accessing Videos from this Website</b>	To access the videos from this documentation website, click <a href="#">All Opswise Automation Center 5.2.0 Videos</a> in the home page navigation panel.
<b>Accessing Videos from Opswise Automation Center</b>	To access videos from the Opswise Automation Center user interface, click <b>Automation Center &gt; Support Links &gt; Video Classroom</b> in the navigation pane.

# Tutorials

These tutorials guide you through features of Opwise Controller. They also provide links to more detailed information about each aspect of the Controller.

Once you have completed the tutorials, you should have a basic understanding of how the features work together to automate your environment.

Before you begin, we recommend you watch a short (two minute) [video](#) about how to navigate around the Opwise Controller user interface and/or read the [Getting Started](#) page.



## Note

During the course of performing these tutorials, you will build up a small database of demonstration records that you will re-use in subsequent exercises. To avoid having to reenter data, do not delete the records.

The navigation pane on the left lists the tutorials in the sequence they should be read; many of them refer to information created in one or more previous tutorials.

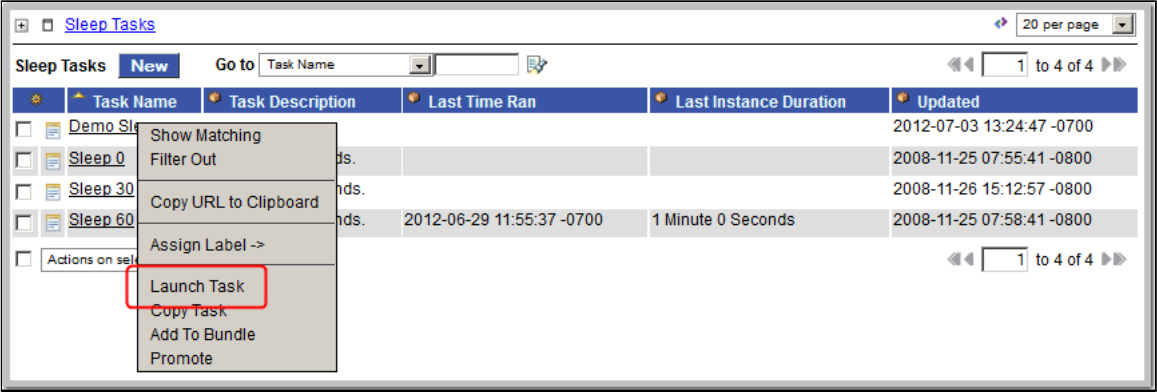
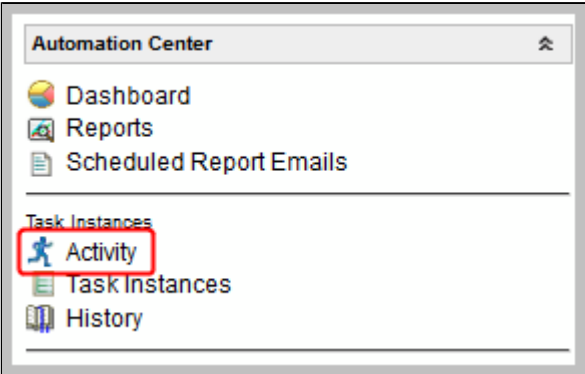
The following table lists the tutorials by topic, rather than sequence.

<b>Tasks</b>
<a href="#">Creating and Manually Launching a Simple Task</a>
<a href="#">Running a Windows Task</a>
<a href="#">Launching a Task Automatically Using a Simple Time Trigger</a>
<a href="#">Launching a Task Every Monday Except Holidays</a>
<a href="#">Launching a Task Every Two Hours During Workday</a>
<a href="#">Launching Tasks at a Future Date and Time</a>
<a href="#">Launching an Email Task Based on a File Monitor</a>
<a href="#">Launching an Email Task Based on a Task Monitor</a>
<a href="#">Launching Tasks Using a Cron Trigger</a>
<a href="#">Aborting a Process Launched by a Task</a>
<a href="#">Force Finishing, Force Finish-Cancelling, and Cancelling a Task</a>
<a href="#">Accessing Task Instance Details</a>
<b>Workflows</b>
<a href="#">Creating a Simple Workflow</a>
<a href="#">Running a Workflow with a Conditional Path</a>
<a href="#">Running a Workflow with Skipped Criteria</a>
<a href="#">Finding and Inserting Tasks in an Active Workflow</a>
<a href="#">Skipping, Unskipping, and Showing-Hiding Skipped Task Instances</a>
<b>Variables</b>

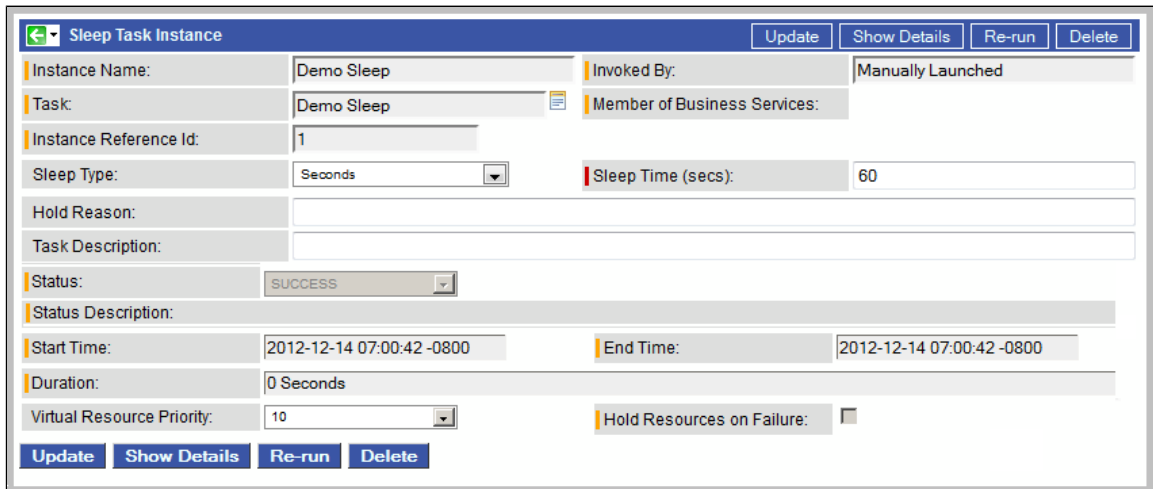
Using Variables in a Simple Task
Using Variables in a Workflow
<b>Custom Days</b>
Creating Custom Days and Periods
<b>Forecasting</b>
Generating Forecast Data
<b>User Interface</b>
Selecting Widgets for the Home Page
Creating a Gauge
Adding Options to the Navigation Pane
<b>Business Services</b>
Business Services
Assigning Records to Business Services
Taking Advantage of Business Services
<b>Reports</b>
Creating a Report
Creating a Report Based on Business Services
Viewing Activity by Business Service
Scheduling and Distributing Reports
Setting Up a Virtual Resource
<b>Security</b>
Creating Users and Assigning Permissions
Creating Security Groups and Assigning Permission

## Tutorial - Creating and Manually Launching a Simple Task

In this exercise, we will create a simple Sleep task and manually launch it.

<b>Step 1</b>	From the navigation pane, select <b>Automation Center &gt; Tasks &gt; Sleep Task</b> . The Sleep Tasks List screen displays.
<b>Step 2</b>	Click <b>New</b> . The Sleep Task Definition screen displays.
<b>Step 3</b>	In the Task Name field, enter <b>Demo Sleep</b> .
<b>Step 4</b>	In the Sleep Time (secs) field, enter <b>60</b> .
<b>Step 5</b>	Click <b>Submit</b> .
<b>Step 6</b>	On the Sleep Task Lists screen, right-click the new Demo Sleep record to display an Action menu.
	 <p>The screenshot shows the 'Sleep Tasks' list with columns: Task Name, Task Description, Last Time Ran, Last Instance Duration, and Updated. The 'Demo Sleep' record is selected, and an action menu is open over it. The menu items are: Show Matching, Filter Out, Copy URL to Clipboard, Assign Label -&gt;, Launch Task (highlighted with a red box), Copy Task, Add To Bundle, and Promote.</p>
<b>Step 7</b>	Select <b>Launch Task</b> .
<b>Step 8</b>	To view the running task, click <b>Activity</b> in the navigation pane.
	 <p>The screenshot shows the 'Automation Center' navigation pane with options: Dashboard, Reports, Scheduled Report Emails, Task Instances, Activity (highlighted with a red box), Task Instances, and History.</p>
<b>Step 9</b>	Locate the task instance Demo Sleep on the Activity screen. When the task completes, the status changes from <b>Running</b> to <b>Success</b> .

**Step 10** To view more details about the Demo Sleep task instance, click its **Instance Name** on the Activity screen. The task instance screen for Demo Sleep displays.



The screenshot displays the 'Sleep Task Instance' details screen. At the top, there is a navigation bar with a back arrow and the title 'Sleep Task Instance', followed by buttons for 'Update', 'Show Details', 'Re-run', and 'Delete'. The main form contains the following fields:

- Instance Name:** Demo Sleep
- Task:** Demo Sleep
- Instance Reference Id:** 1
- Sleep Type:** Seconds (dropdown menu)
- Sleep Time (secs):** 60
- Hold Reason:** (empty text field)
- Task Description:** (empty text field)
- Status:** SUCCESS (dropdown menu)
- Status Description:** (empty text field)
- Start Time:** 2012-12-14 07:00:42 -0800
- End Time:** 2012-12-14 07:00:42 -0800
- Duration:** 0 Seconds
- Virtual Resource Priority:** 10 (dropdown menu)
- Hold Resources on Failure:**

At the bottom of the form, there are buttons for 'Update', 'Show Details', 'Re-run', and 'Delete'.

For additional information, see:

- [Task Overview and Navigation \(8 minute movie\)](#)
- [Defining a Task \(2 minute movie\)](#)
- [Launching a Task \(1 minute movie\)](#)
- [Creating Tasks](#)

# Tutorial - Running a Windows Task

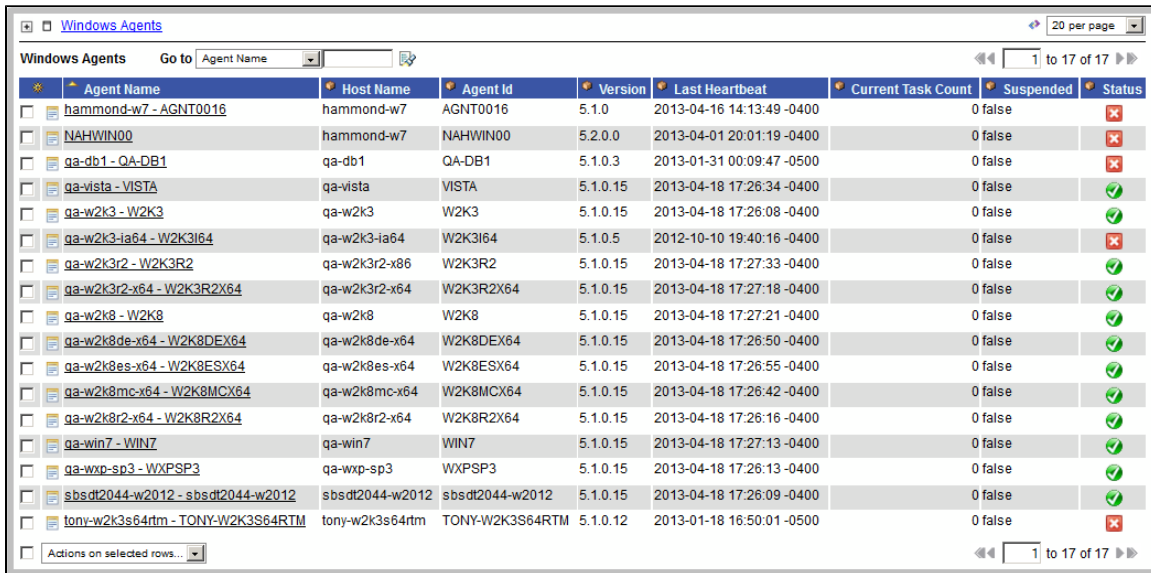
- Viewing Information about the Agent
- Creating a Simple Windows Task
- Manually Retrieving Output from a Windows Task
- Attaching Output to an Email Notification

To do this exercise, you first need a running Opwise Universal Agent for Windows.

## Viewing Information about the Agent

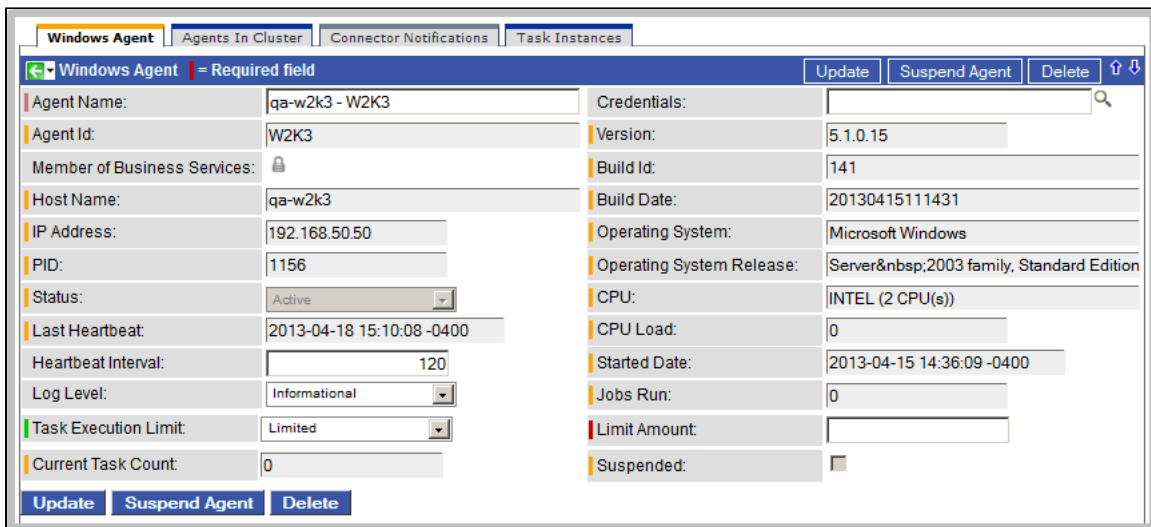
When you install and start an Agent, information about the Agent is automatically registered and added to the database. To view the information:

**Step 1** From the navigation pane, select **Automation Center Resources > Windows Agents**. Depending on what Agents are running, you will see a list similar to the following:



Agent Name	Host Name	Agent ID	Version	Last Heartbeat	Current Task Count	Suspended	Status
hammond-w7 - AGNT0016	hammond-w7	AGNT0016	5.1.0	2013-04-16 14:13:49 -0400	0	false	✗
NAHWIN00	hammond-w7	NAHWIN00	5.2.0.0	2013-04-01 20:01:19 -0400	0	false	✗
qa-db1 - QA-DB1	qa-db1	QA-DB1	5.1.0.3	2013-01-31 00:09:47 -0500	0	false	✗
qa-vista - VISTA	qa-vista	VISTA	5.1.0.15	2013-04-18 17:28:34 -0400	0	false	✓
qa-w2k3 - W2K3	qa-w2k3	W2K3	5.1.0.15	2013-04-18 17:26:08 -0400	0	false	✓
qa-w2k3-ia64 - W2K3I64	qa-w2k3-ia64	W2K3I64	5.1.0.5	2012-10-10 19:40:16 -0400	0	false	✗
qa-w2k3r2 - W2K3R2	qa-w2k3r2-x86	W2K3R2	5.1.0.15	2013-04-18 17:27:33 -0400	0	false	✓
qa-w2k3r2-x64 - W2K3R2X64	qa-w2k3r2-x64	W2K3R2X64	5.1.0.15	2013-04-18 17:27:18 -0400	0	false	✓
qa-w2k8 - W2K8	qa-w2k8	W2K8	5.1.0.15	2013-04-18 17:27:21 -0400	0	false	✓
qa-w2k8de-x64 - W2K8DEX64	qa-w2k8de-x64	W2K8DEX64	5.1.0.15	2013-04-18 17:26:50 -0400	0	false	✓
qa-w2k8es-x64 - W2K8ESX64	qa-w2k8es-x64	W2K8ESX64	5.1.0.15	2013-04-18 17:26:55 -0400	0	false	✓
qa-w2k8mc-x64 - W2K8MCX64	qa-w2k8mc-x64	W2K8MCX64	5.1.0.15	2013-04-18 17:26:42 -0400	0	false	✓
qa-w2k8r2-x64 - W2K8R2X64	qa-w2k8r2-x64	W2K8R2X64	5.1.0.15	2013-04-18 17:26:16 -0400	0	false	✓
qa-win7 - WIN7	qa-win7	WIN7	5.1.0.15	2013-04-18 17:27:13 -0400	0	false	✓
qa-wxp-sp3 - WXPSP3	qa-wxp-sp3	WXPSP3	5.1.0.15	2013-04-18 17:26:13 -0400	0	false	✓
sbsdt2044-w2012 - sbsdt2044-w2012	sbsdt2044-w2012	sbsdt2044-w2012	5.1.0.15	2013-04-18 17:26:09 -0400	0	false	✓
tony-w2k3s64rtm - TONY-W2K3S64RTM	tony-w2k3s64rtm	TONY-W2K3S64RTM	5.1.0.12	2013-01-18 16:50:01 -0500	0	false	✗

**Step 2** Note that the Status column indicates whether this Agent is running and accessible. Click on the Agent Name for your Windows Agent. Details for the Agent display, as shown in this example.



Agent Name:	qa-w2k3 - W2K3	Credentials:	
Agent ID:	W2K3	Version:	5.1.0.15
Member of Business Services:	🔒	Build ID:	141
Host Name:	qa-w2k3	Build Date:	20130415111431
IP Address:	192.168.50.50	Operating System:	Microsoft Windows
PID:	1156	Operating System Release:	Server&nbsp;2003 family, Standard Edition
Status:	Active	CPU:	INTEL (2 CPU(s))
Last Heartbeat:	2013-04-18 15:10:08 -0400	CPU Load:	0
Heartbeat Interval:	120	Started Date:	2013-04-15 14:36:09 -0400
Log Level:	Informational	Jobs Run:	0
Task Execution Limit:	Limited	Limit Amount:	
Current Task Count:	0	Suspended:	<input type="checkbox"/>



**Step 3** Most of the information is protected. See [Displaying Details about Agents](#) for information about which fields you can change.

## Creating a Simple Windows Task

<b>Step 1</b>	Select <b>Tasks &gt; Windows Tasks</b> and click <b>New</b> .
<b>Step 2</b>	In Task Name, type <b>Making Dir "Tutorial"</b> .
<b>Step 3</b>	In the Agent field, select your Windows agent.
<b>Step 4</b>	In the Command field, type: <div style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;"> <pre>md c:\tutorial</pre> </div>
<b>Step 5</b>	Click <b>Submit</b> .
<b>Step 6</b>	From the Windows Tasks list, right-click on <b>Making Dir "Tutorial"</b> and select <b>Launch Task</b> .
<b>Step 7</b>	Check the Activity screen for the task instance.

## Manually Retrieving Output from a Windows Task

In this Windows task, we will run a DIR command. Normally, you would use the [Automatic Output Retrieval](#) field to specify that any output generated by the command be attached to the task instance after the task completes. However, if you did not specify that output be attached, you can manually retrieve it after the task instance has run. In this exercise, we will manually retrieve and display the output.

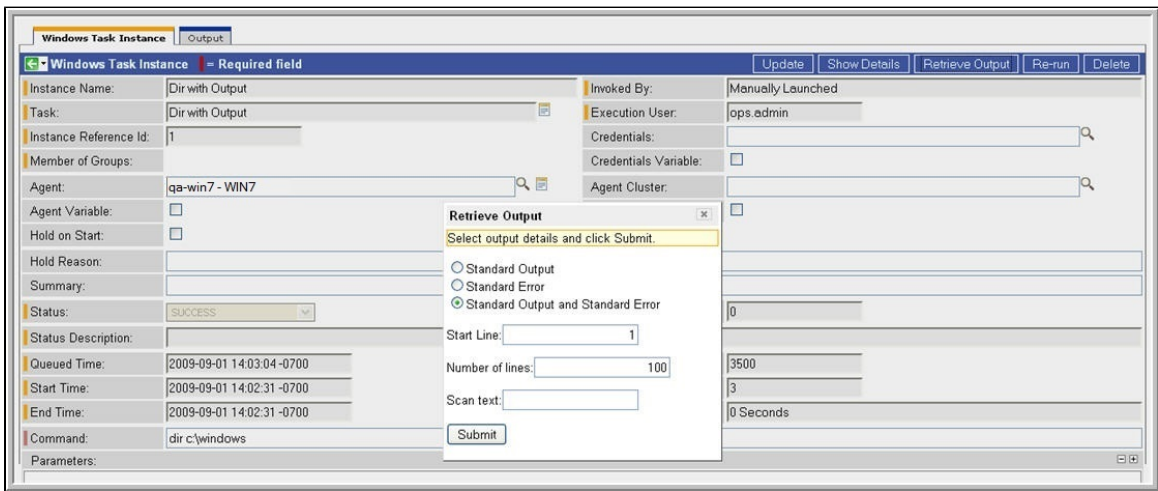
**Step 1** Use the same steps to create another Windows task called **Dir with Output**. Do not specify Automatic Output Retrieval, and use the following command:

```
dir c:\windows
```

**Step 2** Save and launch the task.

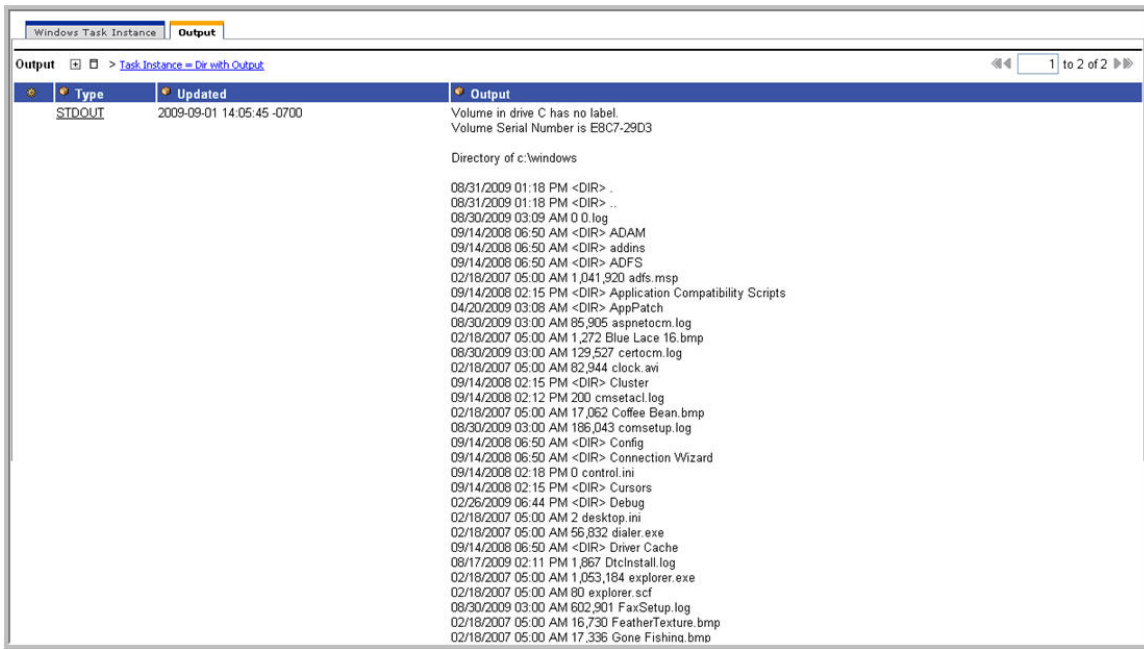
**Step 3** View the task on the Activity screen, and click on the name.

**Step 4** Note that the Output tab is empty. Click **Retrieve Output**. The Retrieve Output window appears:



**Step 5** Change the number of lines to 300 and click **Submit**.

**Step 6** Click the Output tab. The STDOUT (standard output) displays on the output list, as shown in the following example:



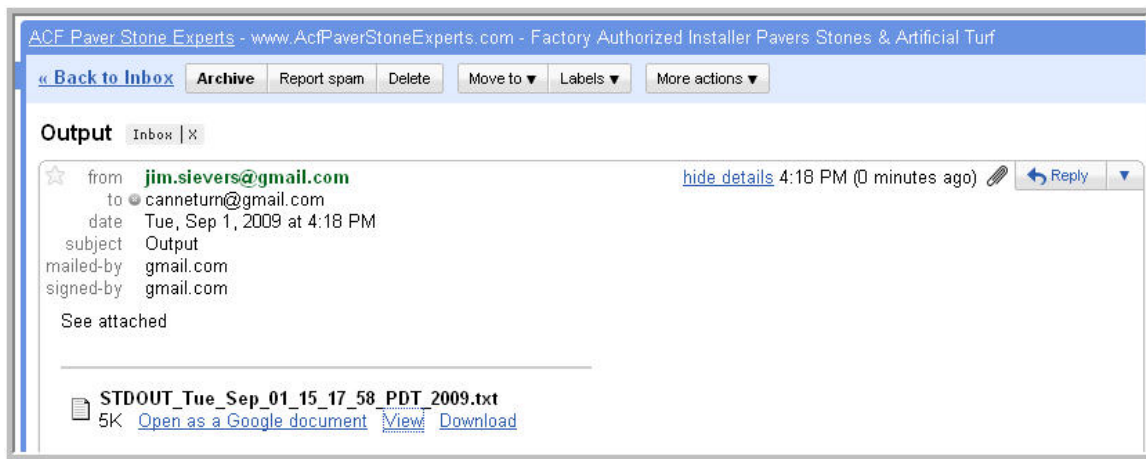
You can also click the STDOUT to view the record.

## Attaching Output to an Email Notification

In this exercise, we will modify our DIR Windows task with an Email notification that includes the output from our DIR command.



<b>Step 1</b>	Open the <b>Dir with Output</b> > task you just created.
<b>Step 2</b>	Click the Actions tab and click <b>New</b> .
<b>Step 3</b>	Select <b>Email Notification</b> .
<b>Step 4</b>	Specify the following: <ul style="list-style-type: none"> <li>• Status=Success</li> <li>• Email Connection=Your email connection</li> <li>• To=Your email address</li> <li>• Subject=Output</li> <li>• Body=See attached.</li> <li>• Attach Standard Output=enabled</li> <li>• Start Line=1</li> <li>• Number of lines=300</li> </ul>
<b>Step 5</b>	Click <b>Submit</b> .
<b>Step 6</b>	Launch the task.

**Step 7** Once the Windows task goes to success, check your inbox for the email. The output is attached, as shown in the following example.



## Tutorial - Launching a Task Automatically Using a Simple Time Trigger

In this exercise, we will define a time trigger that launches our Sleep task every one minute.

<b>Step 1</b>	From the navigation pane, select <b>Automation Center &gt; Triggers &gt; Time Triggers</b> .
<b>Step 2</b>	Click <b>New</b> .
<b>Step 3</b>	In the Trigger Name field, type <b>Launch Sleep Every Minute</b> .
<b>Step 4</b>	In the Task(s) field, click the lock icon.
<b>Step 5</b>	Click the magnifying glass to <b>browse</b> for the Demo Sleep task (created in the <a href="#">Creating and Manually Launching a Simple Task tutorial</a> ).
	<div style="background-color: #e0ffe0; padding: 10px; border: 1px solid #c0ffc0;"> <p> <b>Tip</b> Type <b>D</b> and click the Search icon  to skip to tasks beginning with "D".</p> </div>
	 <p>The screenshot shows the 'Automation Center - Mozilla Firefox' window. The URL is <a href="http://www.opswissoftware.com:8080/opswise/ops_task_list.do?sysparm_target=ops_trigger_time.task&amp;sysparm_target_value=">http://www.opswissoftware.com:8080/opswise/ops_task_list.do?sysparm_target=ops_trigger_time.task&amp;sysparm_target_value=</a>. The page displays a list of tasks under the heading 'Tasks'. The 'Demo Sleep' task is circled in red, and a callout box with the text 'Click to select' points to it. Other tasks in the list include: \${STATE}, 3-Job Dependencies, Broadcast in Middle, Check out Get OpsWise, Everything Workflow, Get Opwise, gs1, gs2, gs3, gs4, gs5, My First Windows Task, Opwise, and Opwise - A=\${A}.</p>
<b>Step 6</b>	Locate the Demo Sleep task and click on the name. Demo Sleep is added to the list of tasks that will be launched by this trigger. You can select as many tasks as you want for each trigger.
<b>Step 7</b>	In the Time Style field, select <b>Time Interval</b> .
<b>Step 8</b>	In the Time Interval field, type 1.

**Step 9** In the Time Interval Units field, select **Minutes**.

The screenshot shows the 'Time Trigger' configuration form. The 'Time Interval Units' field is highlighted with a red box and set to 'Minutes'. The 'Time Interval' field is also highlighted and set to '1'. The 'Trigger Name' is 'Launch Sleep Every Minute'. Other fields include 'Calendar' (System Default), 'Skip Count' (0), 'Time Zone' (US/Eastern), 'Day Style' (Simple), and 'Daily' (checked). A 'Submit' button is visible at the bottom left.

**Step 10** Click **Submit**.

**Step 11** By default, triggers are disabled. To enable this trigger:

1. On the trigger list, locate the Launch Sleep Every Minute trigger you just submitted.
2. Right-click the name of the trigger and, on the **Action** menu, select **Enable Trigger**. Opwise Controller enables the trigger. Note that the Enabled column on the trigger list now displays a green check-mark for this trigger.

**Step 12** From the navigation pane, select **Activity**. Note that a new instance of Demo Sleep appears every one minute.

**Step 13** Click the most recent Sleep Task to view its details, and note that the Invoked by field contains the name of the trigger that launched this task.

For additional information, see:

- [Creating Triggers](#)
- [Time Trigger](#)
- [Enabling and Disabling Triggers](#)
- [Time Triggers \(2 minute movie\)](#)

## Tutorial - Launching a Task Every Monday Except Holidays

- Introduction
- Create Calendar and Custom Days
- Create Time Trigger
- List Qualifying Times

### Introduction

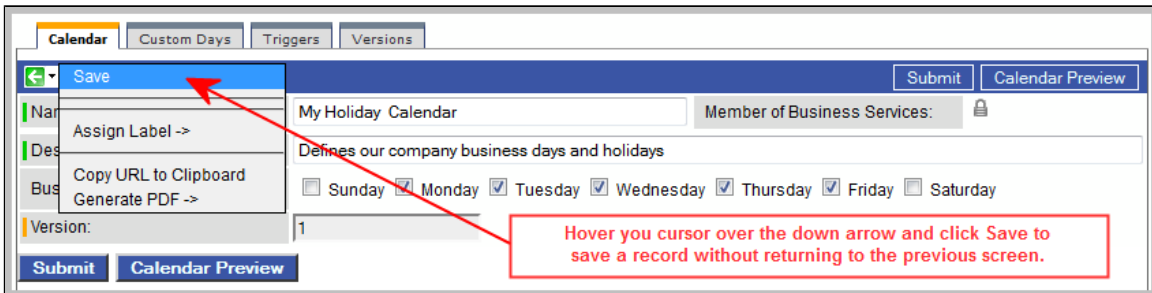
In this exercise, we will define a trigger for the Demo Sleep task (created in the [Creating and Manually Launching a Simple Task](#) tutorial) that runs the task automatically every Monday at one o'clock in the afternoon, except holidays.

For cases where a Monday falls on a holiday, we will define a special restriction that instructs Opwise Controller to run the task on the next business day. We will define the business days and holidays in a calendar and define the special restriction in the trigger.

### Create Calendar and Custom Days

Before building the trigger, we will create the calendar we will be using:

- |               |   |
|---------------|---|
| <b>Step 1</b> | From the navigation pane, select <b>Automation Center &gt; Calendars</b> .  |
| <b>Step 2</b> | Click <b>New</b> .  |
| <b>Step 3</b> | In the Name field, type <b>My Holiday Calendar</b> .  |
| <b>Step 4</b> | In the Comments field, type <b>Defines our company business days and holidays</b> .   |
| <b>Step 5</b> | Leave the default selections for business days (Monday through Friday).   |
| <b>Step 6</b> | We will define each of the holidays using the Calendar Custom Days tab. Before we can access the tab, we must first save the Calendar record to the database by accessing the Action menu and selecting <b>Save</b> . |



The screenshot shows the 'Calendar' form in the Opwise Controller interface. The form has tabs for 'Calendar', 'Custom Days', 'Triggers', and 'Versions'. The 'Calendar' tab is active. The 'Name' field contains 'My Holiday Calendar' and the 'Description' field contains 'Defines our company business days and holidays'. The 'Business Days' section has checkboxes for Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, and Saturday. The 'Save' button is highlighted in the Action menu, and a red arrow points to it. A red box contains the text: 'Hover your cursor over the down arrow and click Save to save a record without returning to the previous screen.'

- |               |                                   |
|---------------|-----------------------------------|
| <b>Step 7</b> | Click the <b>Custom Days</b> tab. |
|---------------|-----------------------------------|

**Step 8** A number of Custom Days definitions are included in the default database. First we will select several of those for our calendar:

1. Click **Edit**. The Edit Members screen is available in many areas of the Controller. It allows you to select records and associate them with the current record. In this case, we will select a number of holidays and associate them with our holiday calendar.
2. Press the CTRL key and click several of the holidays listed in the Collection list.
3. Click the **Add** button (right arrow). The days you selected are added to the Custom Days list.

The screenshot shows the 'Edit Members' interface. At the top, there are 'Add Filter' and 'Run Filter' buttons. Below them is a dropdown menu labeled '-- choose field --'. The main area is divided into two panes: 'Collection' on the left and 'Has Custom Days List' on the right. The 'Collection' pane contains a search box and a list of holidays: Ops - Christmas, Ops - Columbus Day, Ops - Flag Day, Ops - Independence Day, Ops - Labor Day, Ops - Martin Luther King Jr. Day, Ops - Memorial Day, Ops - New Year's Day, Ops - President's Day, Ops - Thanksgiving Day USA, and Ops - Veterans Day. A red arrow points from a text box 'Select days and click Add.' to the 'Ops - Christmas' entry. The 'Has Custom Days List' pane is titled 'My Holiday Calendar' and currently shows '--None--'. Between the two panes are 'Add' and 'Remove' buttons, with the 'Add' button circled in red and a red arrow pointing to it. At the bottom of the interface are 'Save' and 'Cancel' buttons. Below the buttons, there is a summary table:

Name	Ops - Christmas
Comments	Christmas Day - Dec 25th
Holiday	true

Use "Add Filter" and "Run Filter" to isolate the records to pick from

4. Click **Save**. The holidays you selected are now shown in the Custom Days tab.

**Step 9** Now we will add two new custom days to our holiday calendar. For our first custom holiday, let's say our company holds an inventory day on the third Monday of each January, during which the company closes. To create this Custom Day:

1. From the Custom Days list, click **New**.
2. In the Name field, type **Inventory Day**.
3. Click to enable the Holiday field, meaning this day should be treated like a holiday. We need to do this because we are going to include instructions in the trigger to roll the date to the following Tuesday if a Monday falls on a holiday.
4. In the Type field, select **Relative Repeating Date**. Three additional fields appear: When, Day of Week, and Month.
5. Make the following selections:
  - When = 3rd
  - Day of Week = Monday
  - Month = January

6. Click **Submit** to add this custom day to the calendar.
7. Now we will add another holiday. Let's say our company celebrates the birthday of its founder, which is April 1. Click the **New** button and fill in the fields as follows:
  - Name = Founder's birthday
  - Holiday = enabled
  - Comments = The founder's birthday
  - Type = Absolute Repeating Date
  - Month = April
  - Day = 1
8. Click **Submit** to add this day to the calendar.

**Step 10** You have already saved the calendar record and each custom day record, so you do not need to return to the main screen and save the calendar again. Note that you can also use the new custom days records in other calendars.

Now we are ready to build our trigger.

## Create Time Trigger

<b>Step 1</b>	From the navigation pane, select <b>Automation Center &gt; Triggers &gt; Time Trigger</b> .
<b>Step 2</b>	Select <b>New</b> .
<b>Step 3</b>	In the Trigger Name field, type <b>Every Monday at One, Tuesday on Holidays</b> .



**Step 4**

In the tasks field, select the Demo Sleep task. Hint: To skip to tasks beginning with "D" without opening the browse window, type "D" into field and select from the list that displays.

The screenshot shows the 'Time Trigger' configuration form. The 'Task(s)' field is highlighted with a red box, showing a search for 'D' and the selection of 'Demo Sleep'. The form includes the following fields and values:

- Trigger Name: Every Monday at One, Tuesday on Holi
- Enabled:
- Enabled By: [Empty]
- Forecast: [Empty]
- Member of Business Services: [Empty]
- Version: 1
- Calendar: System Default
- Skip Count: 0
- Skip Trigger if Active:
- Description: Run every Monday at one, roll to Tuesday if Monday is a holiday
- Time Zone: US/Eastern
- Time Style: Time
- Day Style: Simple
- Daily:
- Business Days:
- Specific Day(s):
- Time (hh:mm): 13:00
- Sun:
- Mon:
- Tue:
- Wed:
- Thu:
- Fri:
- Sat:
- Special Restriction:
- Simple Restriction:
- Complex Restriction:
- Situation: On Holiday
- Restriction Mode: Or
- Restriction Adjective: Last
- Restriction Noun: Monday
- Restriction Qualifier: Year
- Action: Next Business Day
- Next Scheduled Time: [Empty]

**Step 5** In the Calendar field, select **My Holiday Calendar**.

**Step 6** In the Description field, type **Run every Monday at one, roll to Tuesday if Monday is a holiday**.

**Step 7** In the Time Style field, leave the default, **Time**, and in the Time field, enter **13:00**, which is 1 o'clock p.m. in 24-hour time.

**Step 8** First we will define a simple restriction. In the Day Style field, leave the default, **Simple**.

**Step 9** Enable the Specific Day field and select **Monday**.

**Step 10** Enable the Special Restriction field. Several additional fields appear, which allow you to specify special processing. We will use these fields to instruct the Controller to roll the job to Tuesday if a Monday falls on any of the holidays defined in the calendar – in our case, My Holiday Calendar. Fill in the fields as follows:

- Situation = On Holiday (holidays are defined in the calendar)
- Action = Next Business Day (business days are defined in the calendar)

<b>Step 11</b>	<p>For the purposes of the next exercise (below), we are going to change our simple restriction to a more complex restriction. We not only want to roll to Tuesdays on holidays, but our company has a extra processing load on the last Monday of the year so we want to roll this task to the next business day on that day as well. To provide such instructions in the trigger, select <b>Complex Restriction</b> and make the following selections:</p> <ul style="list-style-type: none"><li>• Situation = On Holiday</li><li>• Restriction Mode = Or</li><li>• Restriction Adjective = Last</li><li>• Restriction Noun = Monday</li><li>• Restriction Qualifier = Year</li></ul>
<b>Step 12</b>	Click <b>Submit</b> .
<b>Step 13</b>	To enable this trigger, right-click on the trigger name in the list and select <b>Enable Trigger</b> .

## List Qualifying Times

Now we will test our Time trigger and its date selection parameters using the List Qualifying Times function.

<b>Step 1</b>	Display the Time Trigger you just created called <b>Every Monday at One, Tuesday on Holidays</b> .
---------------	--

**Step 2** Click **List Qualifying Times**. The Controller calculates the next 20 dates that this trigger will be satisfied and lists them in a new browser tab. In the following example, three Tuesdays were selected in place of the following Mondays:

- Labor Day, 1st Monday of September, specified in the calendar
- Columbus Day, 2nd Monday of October, specified in the calendar
- The last Monday of the year, specified in the trigger

**Trigger Name: Every Monday at One, Tuesday on Holidays**

**Run every Monday at one, roll to Tuesday if Monday is a holiday**

User/Trigger Timezone: America/Los_Angeles
Monday, August 31, 2009 00:00:00 PDT -0700
Tuesday, September 08, 2009 00:00:00 PDT -0700
Monday, September 14, 2009 00:00:00 PDT -0700
Monday, September 21, 2009 00:00:00 PDT -0700
Monday, September 28, 2009 00:00:00 PDT -0700
Monday, October 05, 2009 00:00:00 PDT -0700
Tuesday, October 13, 2009 00:00:00 PDT -0700
Monday, October 19, 2009 00:00:00 PDT -0700
Monday, October 26, 2009 00:00:00 PDT -0700
Monday, November 02, 2009 00:00:00 PST -0800
Monday, November 09, 2009 00:00:00 PST -0800
Monday, November 16, 2009 00:00:00 PST -0800
Monday, November 23, 2009 00:00:00 PST -0800
Monday, November 30, 2009 00:00:00 PST -0800
Monday, December 07, 2009 00:00:00 PST -0800
Monday, December 14, 2009 00:00:00 PST -0800
Monday, December 21, 2009 00:00:00 PST -0800
Tuesday, December 29, 2009 00:00:00 PST -0800
Monday, January 04, 2010 00:00:00 PST -0800
Monday, January 11, 2010 00:00:00 PST -0800

For additional information, see:

- [Creating Triggers](#)
- [Time Trigger](#)

- [Calendars](#)
- [Displaying Trigger Forecast Information](#)

## Tutorial - Launching a Task Every Two Hours During Workday

In this exercise, we will define a recurring task that runs every two hours, limited to business hours – Monday through Friday from 9 a.m. to 5 p.m. We will also instruct Opwise Controller not to run the task on holidays.

Also, if the task is still running two hours later when it is time to run the next task instance, the Controller will be instructed not to run the next instance.

<b>Step 1</b>	From the navigation pane, select Time Triggers.
<b>Step 2</b>	Click <b>New</b> .
<b>Step 3</b>	In the Trigger Name field, type <b>Every Two Hours During Business Hours</b> .
<b>Step 4</b>	Select the Demo Sleep task (created in the <a href="#">Creating and Manually Launching a Simple Task</a> tutorial) in the Task(s) field and My Holiday Calendar (created in the <a href="#">Launching a Task Every Monday Except Holidays</a> tutorial) in the Calendar field.
<b>Step 5</b>	Enable the <b>Skip Trigger if Active</b> field. This tells the Controller not to trigger the task if the previous instance of the task is still active.
<b>Step 6</b>	In the Description field, type <b>Run every two hours, Monday thru Friday, 9 to 5; Do not run if holiday</b> .
<b>Step 7</b>	Now we will specify a recurring time trigger of every two hours. In the Time Style field, select <b>Time Interval</b> and specify the following: <ul style="list-style-type: none"> <li>• Time Interval = 2</li> <li>• Time Interval Units = Hours</li> </ul>
<b>Step 8</b>	Next, we will restrict the times that this task will run to business hours. (Note that all times are specified based on the TimeZone specified in the TimeZone field). To restrict times, select the following: <ul style="list-style-type: none"> <li>• Restrict Times = Enabled</li> <li>• Enabled Start = 09:00</li> <li>• Enabled End = 17:00</li> </ul>

**Step 9** To specify that this trigger should run on weekdays, you could select either of the following:

- Day Style = Complex
- Date Adjective = Every
- Date Noun = Business Day
- Date Qualifier = Year

OR

- Day Style = Simple
- Business Days = Enabled

Both methods use the business days specified in the calendar.

The screenshot displays the 'Time Trigger' configuration window. At the top, there are tabs for 'Time Trigger', 'Variables', and 'Versions'. The main area is titled 'Time Trigger' and includes a 'Required field' indicator. The configuration is organized into several sections:

- Trigger Name:** 'Every Two Hours During Business Hours'. To the right, there are fields for 'Enabled' (checkbox), 'Enabled By', 'Forecast' (checkbox), 'Member of Business Services' (lock icon), and 'Version' (set to '1').
- Task(s):** A dropdown menu showing 'Demo Sleep'.
- Calendar:** 'My Holiday Calendar'.
- Skip Count:** '0'.
- Skip Trigger if Active:** (checkbox).
- Description:** 'Run every two hours, Monday thru Friday, 9 to 5; Do not run if holiday'.
- Time Zone:** 'System (US:Eastern)'.
- Time Style:** 'Time Interval'.
- Time Interval:** '2'.
- Time Interval Units:** 'Hours'.
- Enabled Start (hh:mm):** '09:00'.
- Enabled End (hh:mm):** '17:00'.
- Restrict Times:** (checkbox checked).
- Day Style:** 'Complex'.
- Date Adjective:** 'Every'.
- Date Noun:** 'Business Day'.
- Date Qualifier:** 'Year'.
- Special Restriction:** (checkbox checked).
- Simple Restriction:** (checkbox checked).
- Complex Restriction:** (checkbox unchecked).
- Action:** 'Do Not Trigger'.
- Situation:** 'On Holiday'.
- Next Scheduled Time:** (empty field).

**Step 10** To tell the Controller not to run the task on holidays, specify:

- Special Restriction = Enabled
- Situation = On Holiday
- Action = Do Not Trigger

**Step 11** Access the **Action** menu and click **Save** to save the new trigger, then click **Enable Trigger** to activate it.

**Step 12** Click **List Qualifying Times** to display the next 20 scheduled instances.

For additional information, see:

- [Creating Triggers](#)
- [Time Trigger](#)
- [Calendars](#)

## Tutorial - Launching an Email Task Based on a File Monitor

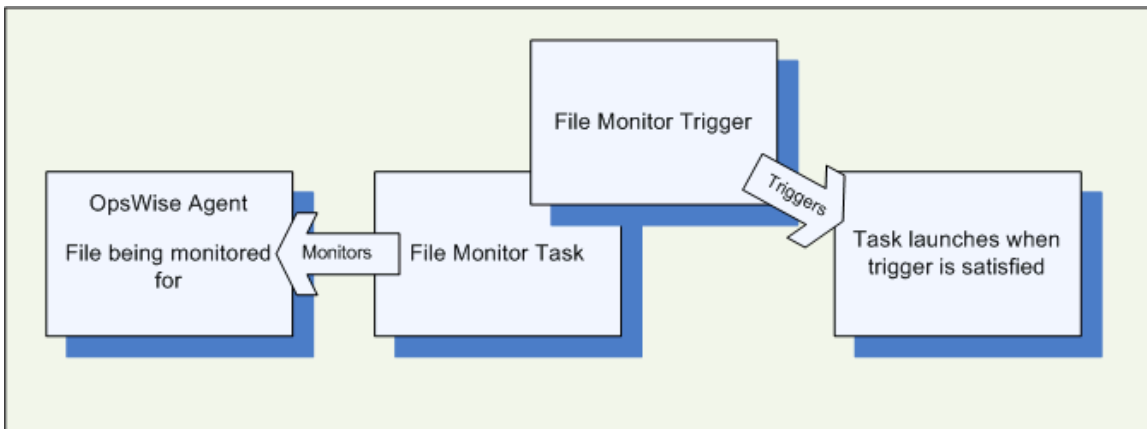
- Introduction
- Create File Monitor Task
- Create an Email Task
- Create File Monitor Trigger
- Test Your File Monitor Setup

### Introduction

In this exercise, we will monitor a machine for a specific file. When the file appears, we will send an email that uses variables to supply information about when and how the email was sent.

In order to set this up, we need the following:

- Opswise Universal Agent
- File Monitor task
- File Monitor trigger
- Email task being launched by the trigger.



#### Prerequisites:

- In order to perform this exercise, you need a Windows, Linux/Unix, or z/OS [Agent](#) running on the machine that is being monitored for the file. Create a directory on the machine called OPS TUTORIAL. Later on, you will copy a text file to this directory to satisfy the File Monitor trigger.



#### Note

If you do not have a running Agent, you can bypass this step by manually satisfying the trigger, as per instructions in the tutorial. However, you do need an Agent defined in the database.)

- Since we are going to generate an email when the trigger is satisfied, you will need an [Email Connection](#) defined.

### Create File Monitor Task

The File Monitor task monitors the agent machine for the specified file.

<b>Step 1</b>	From the navigation pane, select <b>Tasks &gt; File Monitors</b> .
<b>Step 2</b>	Click <b>New</b> .
<b>Step 3</b>	In the Task Name field, type <b>Demo File Monitor</b> .
<b>Step 4</b>	In the Agent field, select the agent you are using for the exercise.

<b>Step 5</b>	In the Monitor Type field, leave the default <b>Create</b> . See <a href="#">File Monitor field descriptions</a> for details about the other file monitor options.
<b>Step 6</b>	In the Monitor File field, type <b>XYZ.TXT</b> . Since we have not specified any directory, the Controller will search the root directory.
<b>Step 7</b>	Enable the Recursive field. Since we are going to write our file to the OPS TUTORIAL directory, we want the Controller to search all sub-directories for the file.
<b>Step 8</b>	In the Stable field, type <b>5</b> . This tells the Controller to satisfy the trigger only when the file has not changed in 5 seconds.
<b>Step 9</b>	Click <b>Submit</b> .

## Create an Email Task

Next, we need to define the task that will run when the File Monitor is satisfied. In this case, we will **generate an email, using the Email task**:

<b>Step 1</b>	From the navigation pane, select <b>Tasks &gt; Email Tasks</b> .
<b>Step 2</b>	Click <b>New</b> .
<b>Step 3</b>	In the Task Name field, type <b>Send Email When File Appears</b> .
<b>Step 4</b>	In the Email Connection field, select the <a href="#">Email Connection</a> you set up for the exercise.
<b>Step 5</b>	In the To field, type in your email address. This is where the email will be sent.
<b>Step 6</b>	In the Subject field, type <b>File XYZ arrived</b> .
<b>Step 7</b>	In the body field, we are going to use an <a href="#">Opswise Controller variable</a> and an <a href="#">Opswise Controller function</a> . Type the following: <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> <pre>Triggered by: \${ops_trigger_name} Date: \${_date}</pre> </div>



**Step 8** When the email is generated, the variables will be substituted with the name of the trigger that launched the Email task, along with the date and time the task was launched.

The screenshot shows the 'Email Task' configuration window. The 'Task Name' is 'Send Email When File Appears'. The 'Email Connection' is 'Opwise - Gmail Account'. The 'To' field is 'can@gmail.com'. The 'Subject' is 'File XYZ arrived'. The 'Body' field contains the text 'Triggered by: \${ops\_trigger\_name}' and 'Date: \$\_date'. There are also checkboxes for 'Late Start', 'Late Finish', 'Early Finish', and 'Hold Resources on Failure', and a dropdown for 'Virtual Resource Priority'.

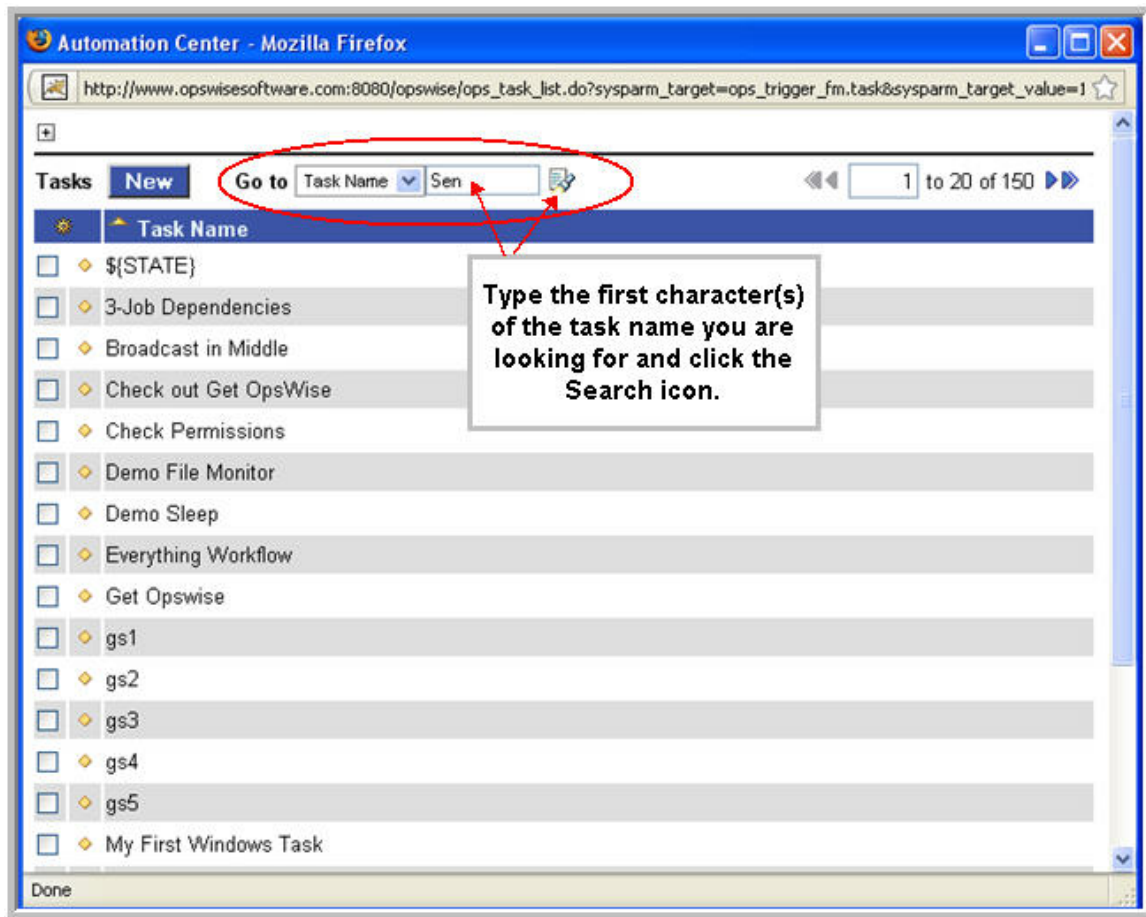
**Step 9** Click **Submit**.

## Create File Monitor Trigger

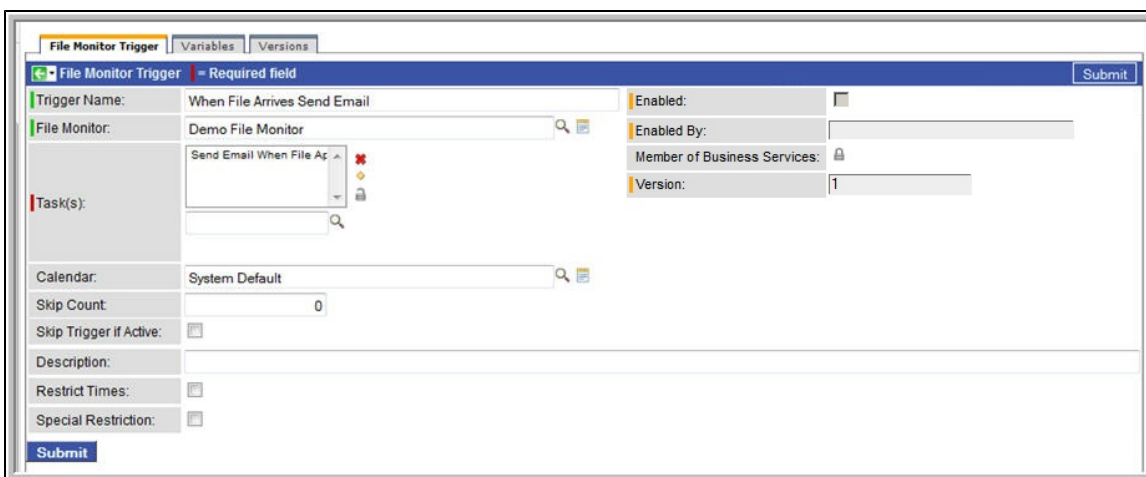
Once the File Monitor task and the Email task being triggered have been defined, we are ready to **create the File Monitor trigger**:

<b>Step 1</b>	From the navigation pane, select <b>Automation Center &gt; Triggers &gt; File Triggers</b> .
<b>Step 2</b>	Click <b>New</b> .
<b>Step 3</b>	In the Trigger Name field, type <b>When File Arrives Send Email</b> .
<b>Step 4</b>	In the File Monitor field, select the File Monitor task that you created, <b>Demo File Monitor</b> .

- Step 5** In the Task(s) field, select the Email Task that you created, **Send Email When File Appears**. (If you have a lot of task definitions, you can jump to the task you want by typing the first few characters into the **Go to** field in the pop-up window and clicking the search icon.)



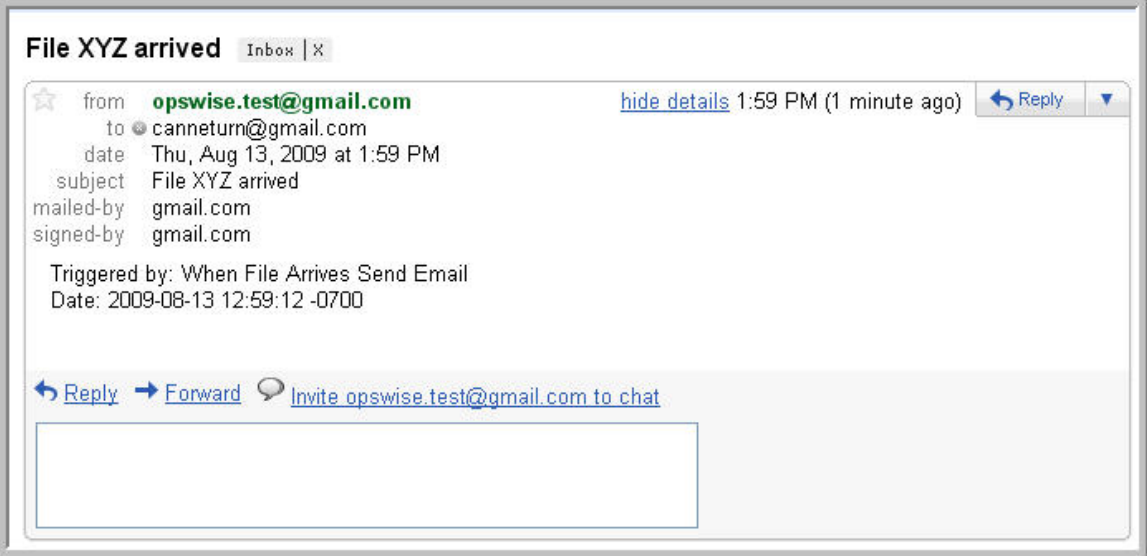
- Step 6** Click **Submit**.



## Test Your File Monitor Setup

The final step is to test the entire setup.

- Step 1** Enable the File Monitor trigger. This launches the File Monitor Task. It will appear in the *Activity display* with a status of Running.

<b>Step 2</b>	<p>Do one of the following:</p> <ul style="list-style-type: none"> <li>• If you have a running agent, place a text file called XYZ.TXT in the OPS TUTORIAL directory on the machine that is being monitored by the File Monitor task. Once the file appears, the File Monitor task waits five seconds as indicated, then satisfies the trigger.</li> <li>• If you do not have a running agent but do have an agent connected to your instance, you can manually satisfy the trigger as follows:             <ol style="list-style-type: none"> <li>1. Select <b>Triggers &gt; File Triggers</b>.</li> <li>2. Locate the When File Arrives Send Email trigger and right-click the name.</li> <li>3. Select <b>Trigger Now</b>.</li> </ol> </li> </ul>
<b>Step 3</b>	<p>When the trigger is satisfied, the Email is sent. Go to the Activity screen and note that the Email task, Send Email When File Appears has been launched.</p>
<b>Step 4</b>	<p>Go to your email account where the email was sent and open the email. Note that the variables were resolved, as shown in the following example:</p>  <p>The screenshot shows an email interface with the following details:</p> <ul style="list-style-type: none"> <li><b>Title:</b> File XYZ arrived</li> <li><b>From:</b> opwise.test@gmail.com</li> <li><b>To:</b> canneturn@gmail.com</li> <li><b>Date:</b> Thu, Aug 13, 2009 at 1:59 PM</li> <li><b>Subject:</b> File XYZ arrived</li> <li><b>Mailed-by:</b> gmail.com</li> <li><b>Signed-by:</b> gmail.com</li> <li><b>Body:</b> Triggered by: When File Arrives Send Email Date: 2009-08-13 12:59:12 -0700</li> <li><b>Actions:</b> Reply, Forward, Invite opwise.test@gmail.com to chat</li> </ul>

For additional information, see:

- [File Trigger \(2 minute movie\)](#)
- [File Monitors \(3 minute movie\)](#)
- [Email Task](#)
- [File Monitor Task](#)
- [File Trigger](#)
- [Variables Overview \(5 minute movie\)](#)
- [Variables](#)

## Tutorial - Creating a Simple Workflow

- Introduction
- Create Tasks
- Creating a Simple Workflow
- Using the Workflow Editor Tools
- Running the Workflow

### Introduction

In this exercise, we will learn how to copy tasks, create a simple workflow of Sleep tasks, and use the tools available in the workflow editor.

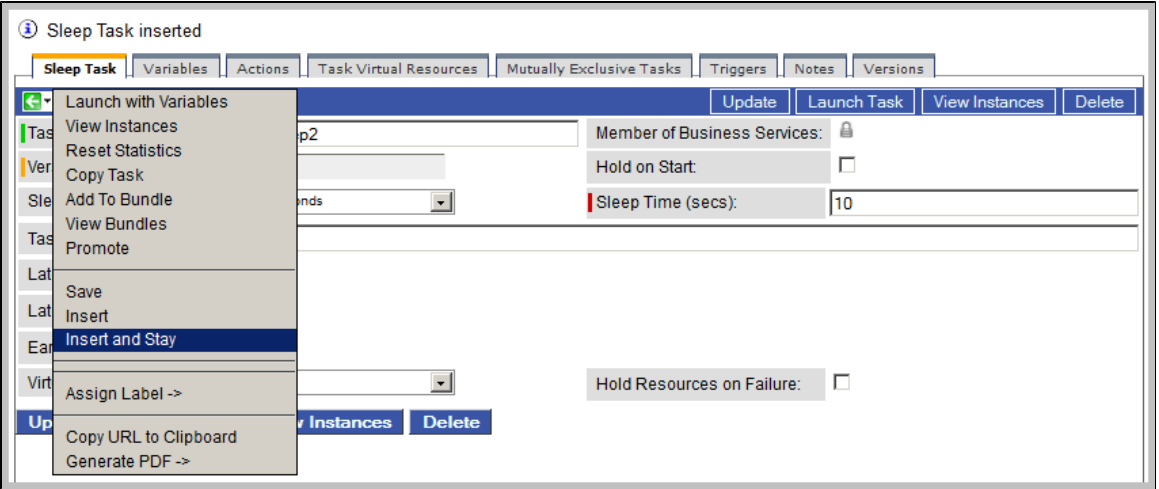
### Create Tasks

Create six [Sleep tasks](#) for use in the workflow. Assign the names Sleep1, Sleep2, and so on, and give each task a sleep time of 10 (seconds).



#### Note

You can [copy tasks](#) using different methods. One method is provided below.

<b>Step 1</b>	Create the first task, Sleep1, access the <a href="#">Action menu</a> , and click <b>Save</b> .
<b>Step 2</b>	In the Task Name field, enter Sleep2.
<b>Step 3</b>	Access the <a href="#">Action menu</a> and click <b>Insert and Stay</b> . 
<b>Step 4</b>	Repeat Steps 2 and 3 for the remaining Sleep tasks.


### Creating a Simple Workflow

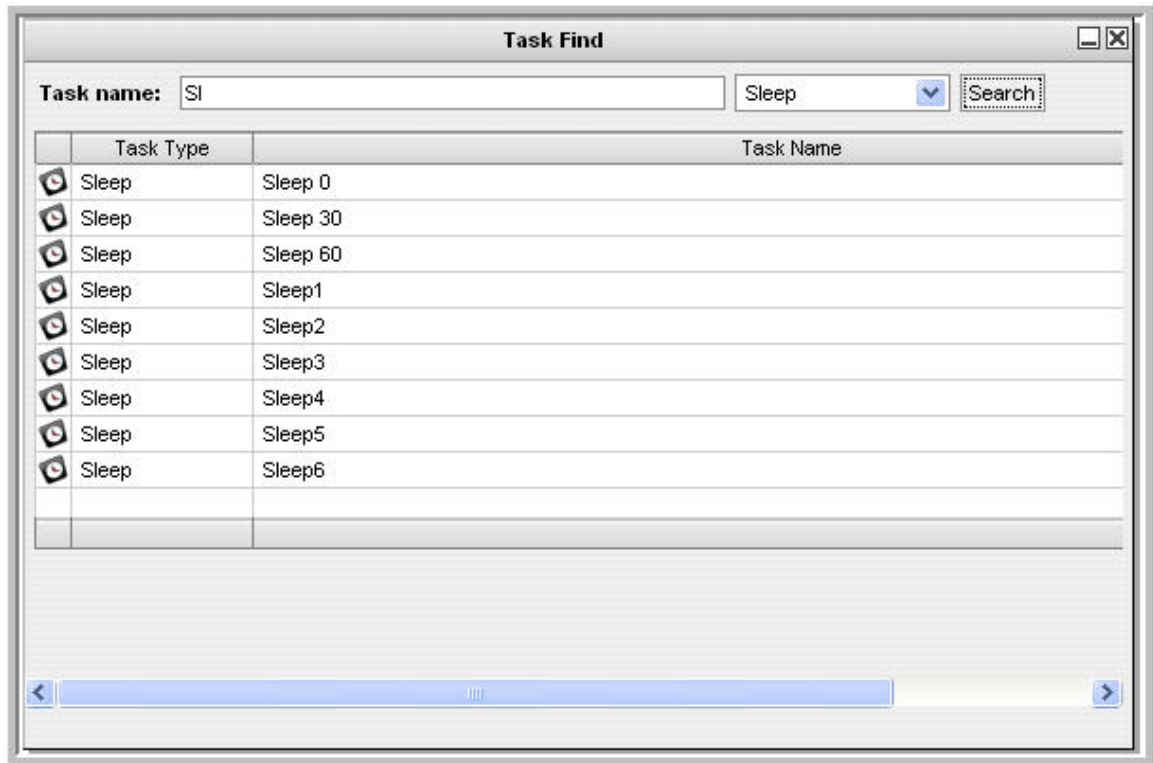
Now that we have six Sleep tasks to work with, we are ready to create a simple workflow:

<b>Step 1</b>	From the navigation pane, select <b>Automation Center &gt; Tasks &gt; Workflow Tasks</b> and click <b>New</b> .
<b>Step 2</b>	In the Task Name field, type <b>Simple Workflow</b> .
<b>Step 3</b>	Use the down-arrow to <a href="#">save the record</a> without exiting the screen. Once the record is saved, note the additional buttons that have appeared: Update, Edit Workflow, Launch Task, and Delete.

**Step 4** Click **Edit Workflow** to access the Workflow Editor. The Workflow Editor operates in several modes, depending on which icon was most recently clicked on the toolbar. See [Workflow Modes](#) for a complete description. Also, see the [icon reference](#) for a description of each icon on the toolbar.

**Step 5** Add a new task:

1. Click the **Add Task**  icon.
2. When the Task Find window pops up, click the \*Search\* button to display all tasks. You can also narrow down the search by typing names or partial names in the Task name field or selecting specific task types, or both.



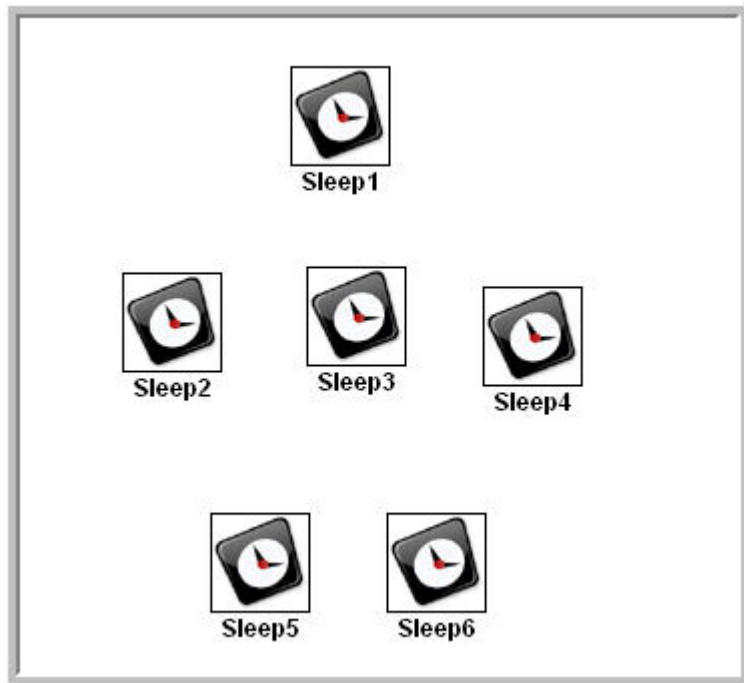
3. Click and drag the icon onto the canvas for the Sleep1 task. Repeat the process for the remaining Sleep tasks. When you are finished, close the Task Find window.

**Step 6** Now we'll arrange our workflow as follows:




- Sleep1 is the top-level task
- Sleep2, 3 and 4 are dependent on Sleep1.
- Sleep5 and 6 are dependent on Sleep3.

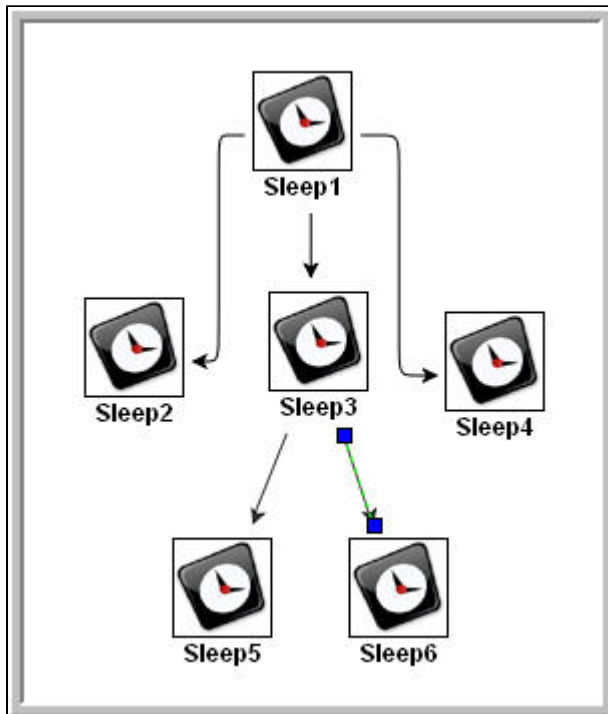
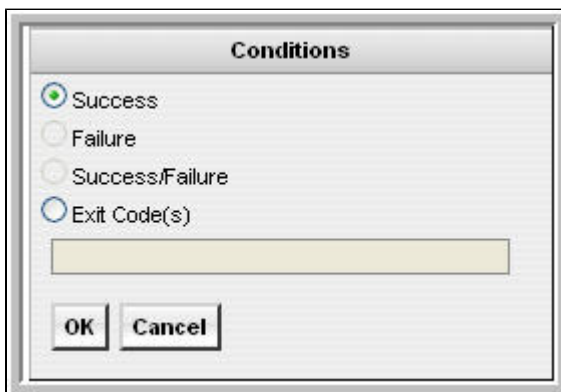
To create these connections:

1. Click and drag Sleep1 to the top of the canvas and place Sleep2, 3, and 4 in a line beneath it.
2. Move Sleep5 and 6 beneath Sleep3.




**Step 7** Now we're ready to create connections.










1. Click the curved connector icon . This puts you in connect mode.
2. Click Sleep1, drag the cursor down to Sleep2, then release the mouse. A curved connection appears.
3. Repeat for Sleep3 and Sleep4.
4. For the next set of connections, we'll use the straight  connector, which creates a diagonal line instead of a curved line. Click the straight  connector, then click Sleep3 and connect it to Sleep5 and Sleep6. Your canvas should now look like this:

**Step 8** The default condition (or dependency) for connectors is Success. That is, the successor task runs if the predecessor task goes to success. To view the conditions, right-click a connector and click **Conditions**. The Conditions window displays.**Note**

Since a Sleep task cannot go to Failure, those options are greyed out.

**Step 9** Cancel out of the Conditions window and click Save  to save the workflow.

## Using the Workflow Editor Tools

- Delete Sleep4 by clicking on it and clicking **Delete** . Note that the connector is also deleted.
- Increase the size of the workflow items by clicking **Zoom In** .
- Decrease the size by clicking **Zoom Out** .
- To change the workflow display to twice its size, click **Zoom**  and specify 200.
- To return it to its original size, click **Actual Size** .
- To reformat your entire workflow into a horizontal display, click **Horizontal Layout** .
- To reformat the workflow into a vertical display, click **Vertical Layout** .
- Two tools are available for viewing very large workflows:
  - To reposition the workflow so you can view a specific section, click **Pan** , click anywhere in the canvas and drag.
  - Click **Outline** . In the Outline window, reposition the blue box around the area you want to display.

See [Icon Reference](#) for a complete description of all workflow editor tools.

## Running the Workflow

Now we will manually launch the workflow and view it from the Activity screen.

<b>Step 1</b>	Click the back arrow to return to the Workflow record.
<b>Step 2</b>	Click <b>Launch Task</b> .
<b>Step 3</b>	From the navigation pane, select <b>Task Instances &gt; Activity</b> . You will see six task instances: the Simple Workflow task, and the five Sleep tasks (assuming you deleted one of the Sleep tasks during the previous exercise).

For additional information, see:

- [Saving, Updating, Deleting, and Copying Records](#)
- [Workflow Overview \(5 minute movie\)](#)
- [Creating Workflows](#)



## Tutorial - Running a Workflow with a Conditional Path

- [Introduction](#)
- [Create a Sleep Task](#)
- [Creating SQL Tasks](#)
- [Creating a Manual Task](#)
- [Creating the Workflow](#)
- [Run the Workflow to Success](#)
- [Run the Workflow Down the Conditional Path](#)

### Introduction

In this exercise, we will create a short workflow of SQL tasks. We will begin with a two-minute Sleep task so that we will have enough time to see what the workflow looks like on the Activity display when we launch it. We will also create a conditional path, as follows:

- The workflow runs seven days a week and creates a new database table. If that is successful, additional SQL tasks run that insert a value, select a count, and delete a value. Each subsequent task runs if the previous is successful.
- If the first (table creation) task fails, the workflow goes to a Manual task instead of the regular flow. This is the conditional path. The Manual task creates a pause in the workflow and sends an email notification. A user is expected to check the database and fix the problem that caused the first task to fail. If the Manual task is set to a Complete status, it goes to Success and the workflow then returns to the remaining SQL tasks. While the Manual task remains in the Action Required status, the successor tasks have a Waiting status.

We will also add an Email notification and a Runbook Note to this workflow.

Prerequisite: Since we are using SQL tasks in this exercise, you will first need to [create a SQL Database Connection](#).

### Create a Sleep Task

We will add a Sleep task at the beginning of our workflow so that we will have a chance to view it when Opwise Controller loads it into the Activity screen.

[Create a Sleep task](#) called Sleep 2 Minutes with a Sleep Time of 120 seconds.

### Creating SQL Tasks

In this exercise, we will create SQL tasks that execute the following SQL commands:

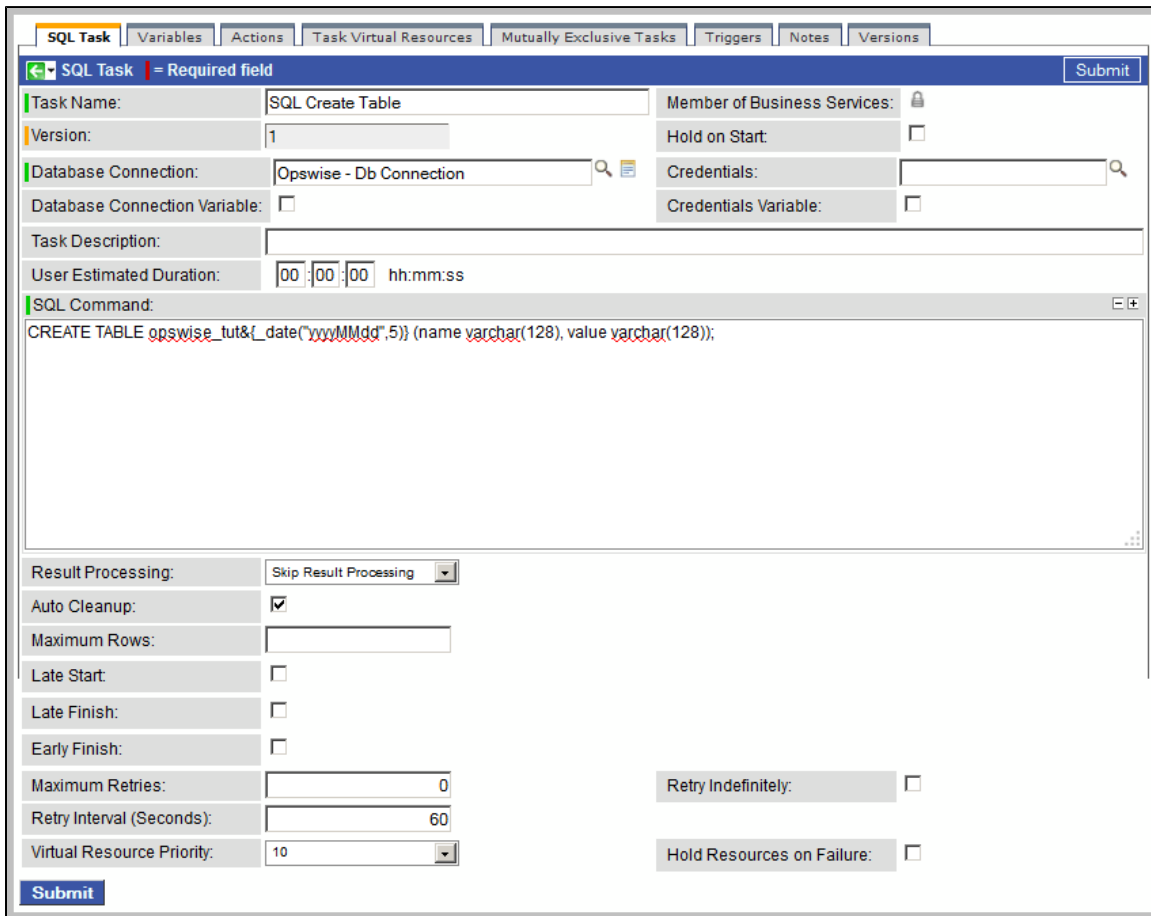
- Create a new table in the database
- Insert a value into the table
- Select a count value from the table
- Delete the value from the table

Follow these steps to create the SQL tasks:

<b>Step 1</b>	From the navigation pane, select <b>Tasks &gt; SQL Tasks</b> and click <b>New</b> .
<b>Step 2</b>	In the Task Name field, type <b>SQL Create Table</b> .
<b>Step 3</b>	In the Database Connection field, select the database connection you created as a prerequisite.

**Step 4** In the SQL Command field, type or cut and paste the following:

```
CREATE TABLE opwise_tut${_date("yyyyMMdd",5)} (name varchar(128), value varchar(128));
```



The screenshot shows the 'SQL Task' configuration window. The 'SQL Command' field is highlighted and contains the following text:

```
CREATE TABLE opwise_tut&{_date('yyyyMMdd',5)} (name varchar(128), value varchar(128));
```

Other visible fields and options include:

- Task Name: SQL Create Table
- Version: 1
- Database Connection: Opwise - Db Connection
- Result Processing: Skip Result Processing
- Auto Cleanup:
- Maximum Rows: (empty)
- Late Start:
- Late Finish:
- Early Finish:
- Maximum Retries: 0
- Retry Interval (Seconds): 60
- Virtual Resource Priority: 10
- Retry Indefinitely:
- Hold Resources on Failure:

**Step 5** Click **Submit**.

**Step 6** Create a SQL task called **SQL Insert Value** with this command:

```
INSERT INTO opwise_tut${_date("yyyyMMdd",5)} (name, value) values ('A', 'F'), ('B', 'S'), ('C', 'F');
```

**Step 7** Create a SQL task called **SQL Select Count** with this command:

```
SELECT count (*) as count FROM opwise_tut${_date("yyyyMMdd",5)} WHERE value = 'F';
```

**Step 8** Create the last SQL task called **SQL Delete** with this command:

```
DELETE FROM opwise_tut${_date("yyyyMMdd",5)};
```

## Creating a Manual Task

A Manual task is used within a workflow to create a pause in processing, during which the user must perform some task. When the user task is complete, the user sets the Manual task to a completed state and processing continues. For our Manual task, we are also going to request a warning if the user takes too long to complete it.

<b>Step 1</b>	Select <b>Tasks &gt; Manual Tasks</b> and click <b>New</b> .
<b>Step 2</b>	In the Task Name field, enter <b>Pause for Manual</b> .
<b>Step 3</b>	In the summary field, we are going to use another variable that indicates the date and time the Manual task launches: A Manual task run at <code>\${_date()}</code> .
<b>Step 4</b>	Enable the Late Finish field and select the following values: <ul style="list-style-type: none"> <li>Late Finish Type = Duration</li> <li>Late Finish Duration = Hours 00 02 00 (2 minutes)</li> </ul>
<b>Step 5</b>	Access the Action menu and click <b>Save</b> . Your task should look like this:

The screenshot shows the configuration form for a Manual Task. The form is titled "Manual Task" and has a "Submit" button. The fields are as follows:

- Task Name:** Pause for Manual
- Version:** 1
- Task Description:** A Manual task run at `${_date()}`
- User Estimated Duration:** 00:00:00 hh:mm:ss
- Late Start:**
- Late Finish:**
- Late Finish Type:** Duration
- Late Finish Duration:** 00:02:00 hh:mm:ss
- Early Finish:**
- Virtual Resource Priority:** 10
- Hold Resources on Failure:**

**Step 6** Add a Runbook Note:

1. Click the **Notes** tab and click **New**. The Notes screen appears.
2. In the Title field, type:

```
Probable database problem
```

3. In the Text field, type:

```
Make sure database is running.
```

4. Click **Submit**.

The screenshot shows a web form titled 'Note'. The 'Title' field contains the text 'Probable database problem'. The 'Text' field contains the text 'Make sure database is running.'. At the bottom of the form, there are two 'Updated:' fields with empty input boxes, and a blue 'Submit' button.

**Step 7** Add an Email Notification:

1. Click the **Actions** tab and click **New**. You are prompted to select Abort Action, Email Notification, Set Variable, or SNMP Notification.
2. Click **Email Notification**.
3. Select the status ACTION REQUIRED.
4. For Email Connection, select the Email Connection you created earlier.
5. In the To field, type your Email address.
6. In the Subject field, type:

```
Issue with Bigger Workflow
```

7. In the Body field, type:

```
${_date} workflow failure; notification triggered by ${ops_task_name}
```

Email Notification = Required field
Submit

<b>Status:</b>	<input type="checkbox"/> DEFINED <input type="checkbox"/> WAITING <input type="checkbox"/> HELD <input type="checkbox"/> RESOURCE REQUESTED <input type="checkbox"/> RESOURCE WAIT <input type="checkbox"/> EXECUTION WAIT <input type="checkbox"/> UNDELIVERABLE <input type="checkbox"/> QUEUED <input type="checkbox"/> SUBMITTED <input checked="" type="checkbox"/> ACTION REQUIRED <input type="checkbox"/> STARTED <input type="checkbox"/> RUNNING <input type="checkbox"/> RUNNING/PROBLEMS <input type="checkbox"/> IN DOUBT <input type="checkbox"/> START FAILURE <input type="checkbox"/> CONFIRMATION REQUIRED <input type="checkbox"/> CANCELLED <input type="checkbox"/> FAILED <input type="checkbox"/> SKIPPED <input type="checkbox"/> FINISHED <input type="checkbox"/> SUCCESS
Exit Codes:	<input style="width: 100%;" type="text"/>
On Late Start:	<input type="checkbox"/>
On Early Finish:	<input type="checkbox"/>
On Late Finish:	<input type="checkbox"/>
Description:	<div style="border: 1px solid gray; height: 30px;"></div>
Email Template:	<input style="width: 90%;" type="text"/>
Email Connection:	<input style="width: 90%;" type="text" value="Opswise - Gmail Account"/>
Reply-To:	<input style="width: 90%;" type="text"/>
To:	<input style="width: 90%;" type="text" value="can@gmail.com"/>
Cc:	<div style="border: 1px solid gray; height: 30px;"></div>
Bcc:	<div style="border: 1px solid gray; height: 30px;"></div>
Subject:	<input style="width: 90%;" type="text" value="Issue with Bigger Workflow"/>
Body:	<div style="border: 1px solid gray; height: 100px;"><p>\$_date} workflow failure; notification triggered by \${op_task_name}</p></div>
Attach Standard Output:	<input type="checkbox"/>
Attach Standard Error:	<input type="checkbox"/>
Attach File:	<input type="checkbox"/>
<b>Submit</b>	

**Note**  
For detailed descriptions, see [Email Notification Actions](#).

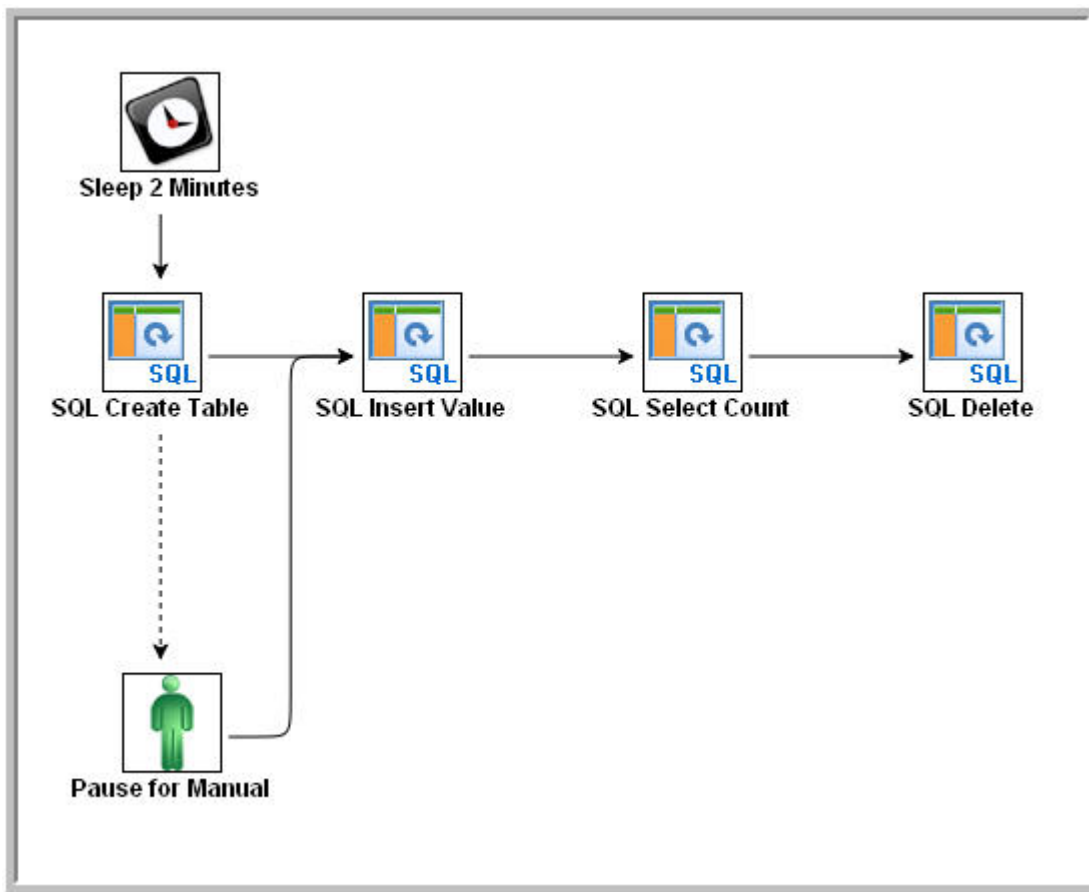
8. Click **Submit**.

## Creating the Workflow

In this exercise, we will use the SQL, Manual, and Sleep tasks we have already created.

<b>Step 1</b>	Select <b>Tasks &gt; Workflow Tasks</b> and click <b>New</b> .
<b>Step 2</b>	In the Task Name field, enter <b>Bigger Workflow</b> and, on the <b>Action</b> menu, click <b>Save</b> .
<b>Step 3</b>	Click <b>Edit Workflow</b> .
<b>Step 4</b>	Use the <b>Add Task</b> tool to drag the tasks you just created onto the canvas: <ul style="list-style-type: none"> <li>• Sleep 2 Minutes</li> <li>• Pause for Manual</li> <li>• SQL Create Table</li> <li>• SQL Delete</li> <li>• SQL Insert Value</li> <li>• SQL Select Count</li> </ul>
<b>Step 5</b>	Organize the tasks as shown in the illustration below.
<b>Step 6</b>	Create the connections shown in the illustration below. If you don't remember how, review the <a href="#">Creating a Simple Workflow</a> exercise.
<b>Step 7</b>	The Success connectors tell the Controller that if SQL Create Table goes to Success, run Insert SQL Value and the rest of the tasks. Now we are going to create a conditional path that says if SQL Create Table fails, the Controller should run the Pause for Manual task: <ol style="list-style-type: none"> <li>1. Right-click the connector between SQL Create Table and Pause for Manual.</li> <li>2. When the menu pops up, select <b>Conditions</b>.</li> <li>3. Enable <b>Failure</b>.</li> <li>4. Click <b>OK</b>. Note that the connector is a dotted line, which indicates a Failure connection.</li> </ol>

**Step 8** On the toolbar, click **Save** .



## Run the Workflow to Success

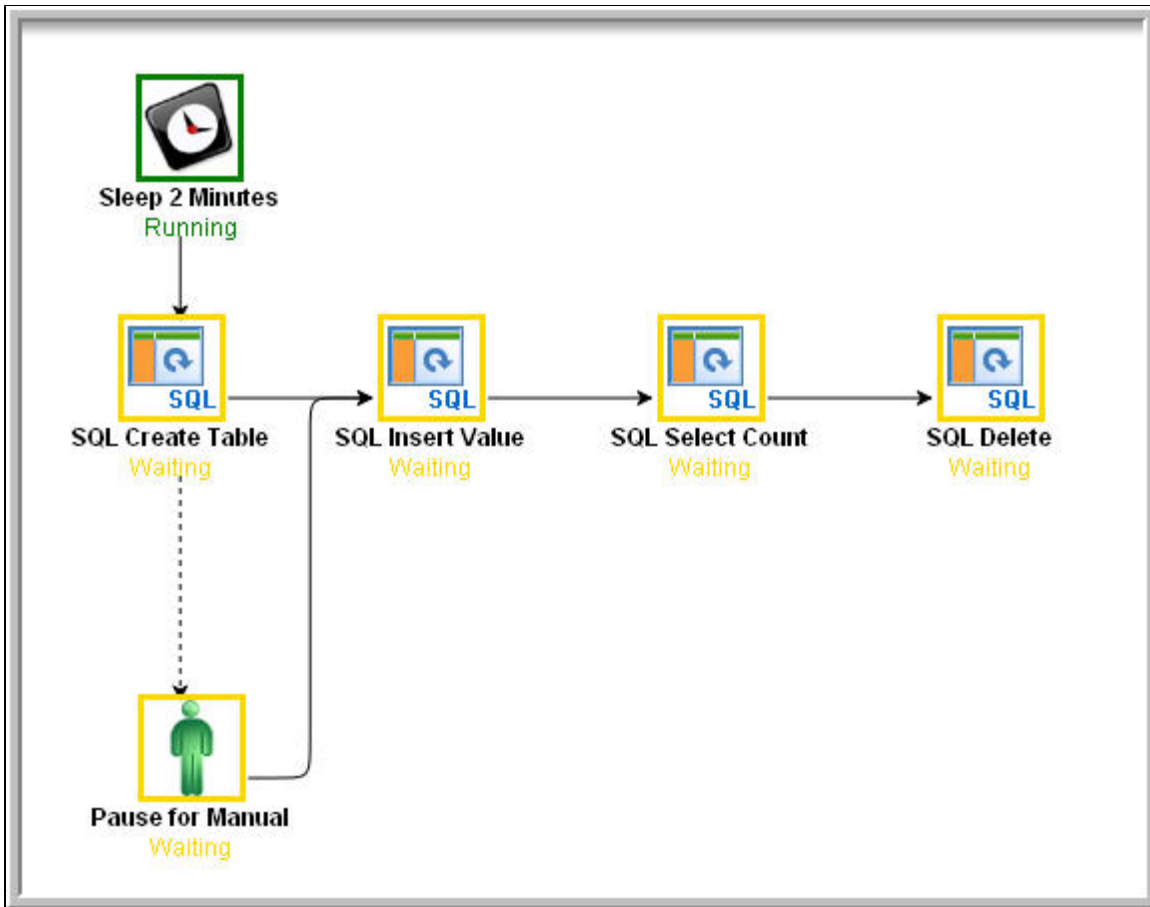
We are going to launch our workflow and view it from two different perspectives: from the Activity Screen view and the Workflow Monitor.

**Step 1** Launch the workflow manually.

**Step 2** Display the Activity screen. Because the Sleep 2 Minutes task is still running, your display should look similar to this:

Today's Task Instances by Created Time			
Instance Name	Type	Duration	Status
<a href="#">Pause for Manual</a>	Manual		Waiting
<a href="#">Sleep 2 Minutes</a>	Sleep		Running
<a href="#">Bigger Workflow</a>	Workflow		Running
<a href="#">SQL Insert Value</a>	SQL		Waiting
<a href="#">SQL Select Count</a>	SQL		Waiting
<a href="#">SQL Delete</a>	SQL		Waiting
<a href="#">SQL Create Table</a>	SQL		Waiting

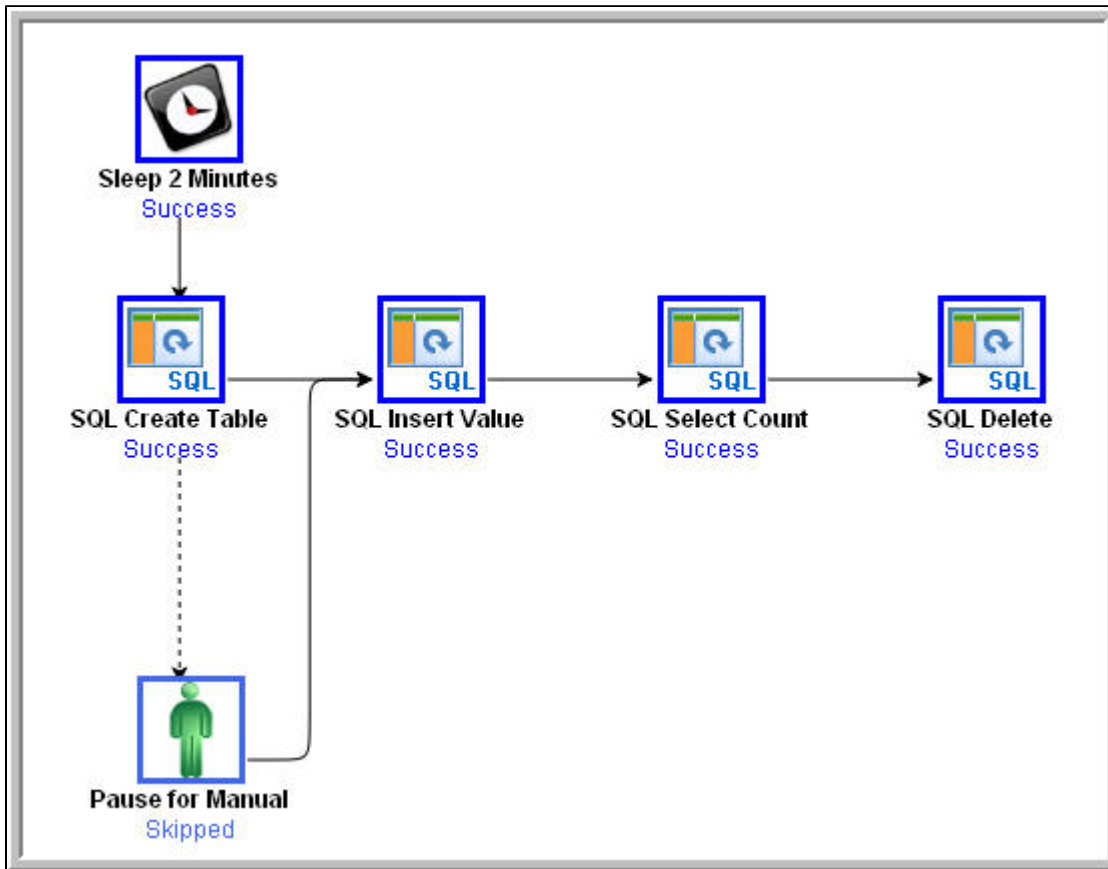
**Step 3** To view the running workflow from the Workflow Monitor, click on the Workflow task. The Workflow Monitor opens and shows progress on the task. The Workflow Monitor updates automatically with each status change.



When the Sleep task finishes, the SQL tasks execute. All tasks go to Success and the workflow goes to Success. The only task that



did not run is the conditional task, Pause for Manual.

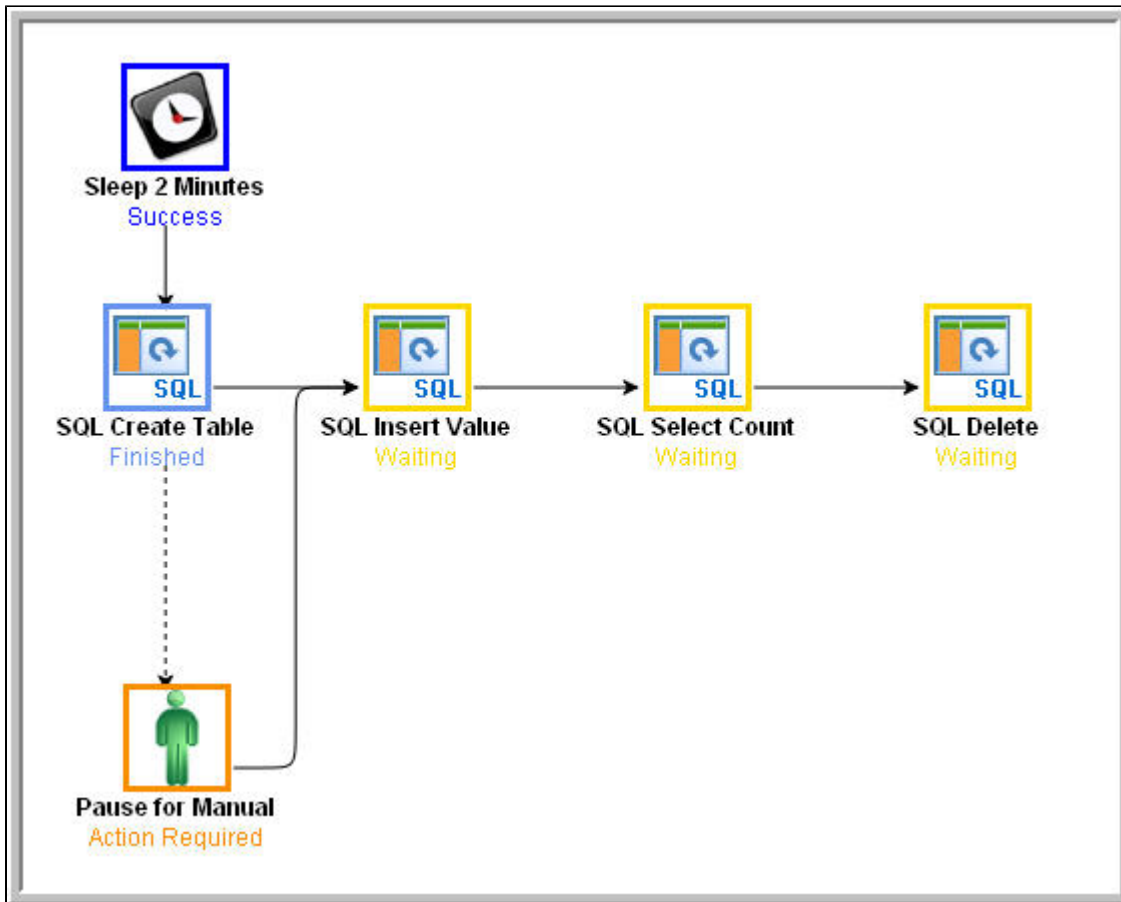


## Run the Workflow Down the Conditional Path

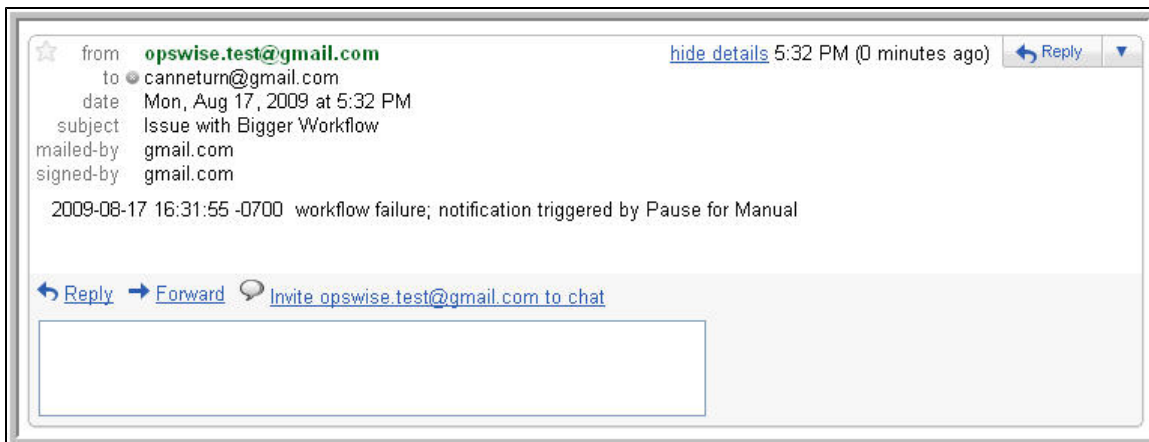
Recall that we inserted a date variable into the INSERT TABLE command. Thus, you can run this workflow every day and get a new table name each day, based on the date. For the purposes of our exercise, assuming you are performing it on the same day you did the previous exercise, the SQL Create Table task will fail this time because the table already exists.

**Step 1** Return to the Bigger Workflow task and launch it again.

**Step 2** From the Activity screen, click the Workflow name to view it from the canvas. This time, the workflow goes down the conditional path. Note that when you set up a conditional path, what would normally be a Failure status for the SQL Create Table task becomes a status of Finished. If you ran this task as a standalone task or without the conditional path, its status would say Failed.



**Step 3** When the Pause for Manual task launches, it generates the Email Notification we added earlier.

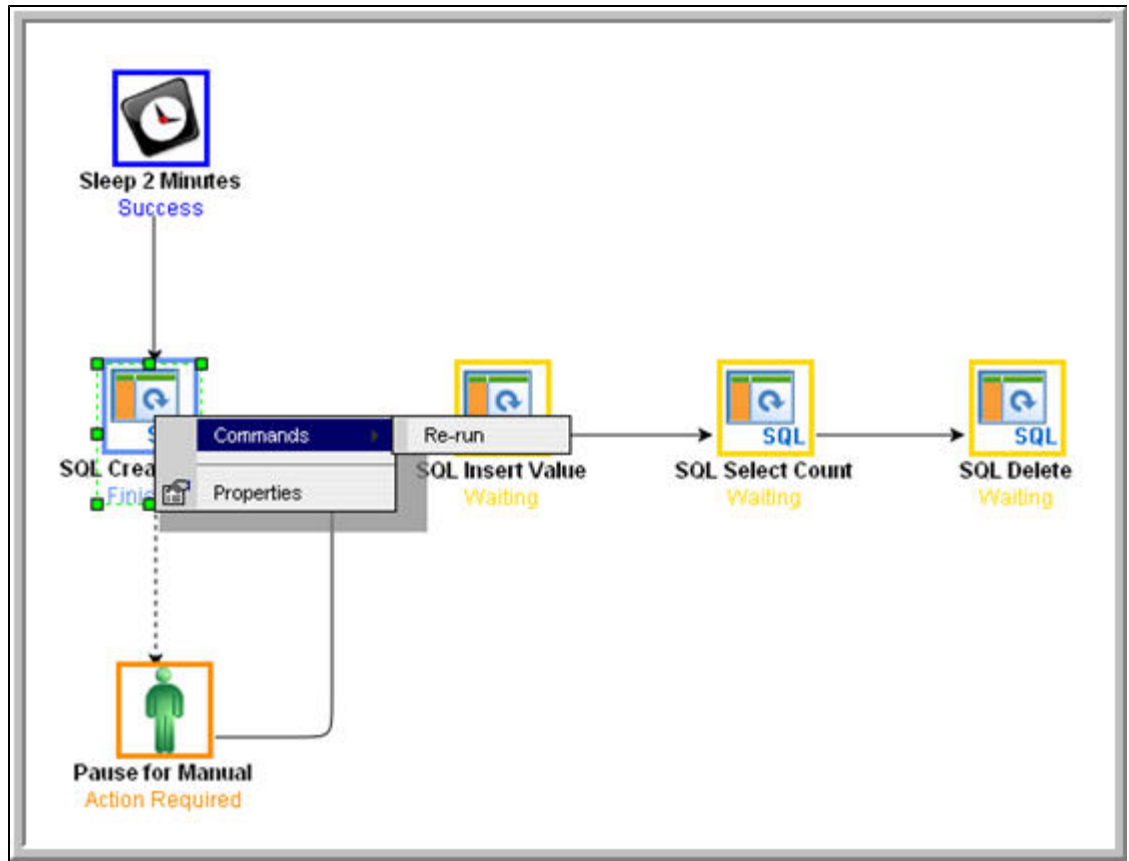


**Step 4** The user receives the email, which provides the name of the task that generated it (Pause for Manual in our case). The user might also be running a special Activity screen that displays only Manual tasks in the Action Required status. According to our scenario, the user opens the Pause for Manual task and checks the Notes to find out what action he or she is supposed to take. In our case, the Notes say to check the database and bring it back up.

**Step 5** At this point, the workflow processing could continue in either of two ways:

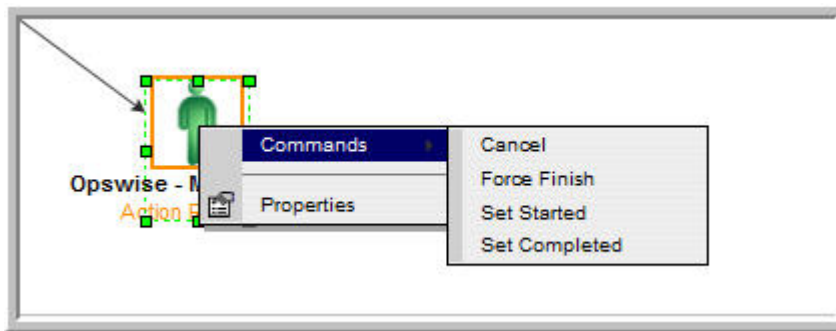
1. Re-run the task that failed and send the workflow down the success path.
  2. Set the Pause for Manual task to Completed status and continue the workflow from there.
- We will try both methods.

3. Re-run the failed task; right-click the task and select **Commands > Re-run**. In a real processing situation, this is the method you would use because you need to create the table before you can continue.
  - a. Right-click the task that failed and select **Commands > Re-run**.

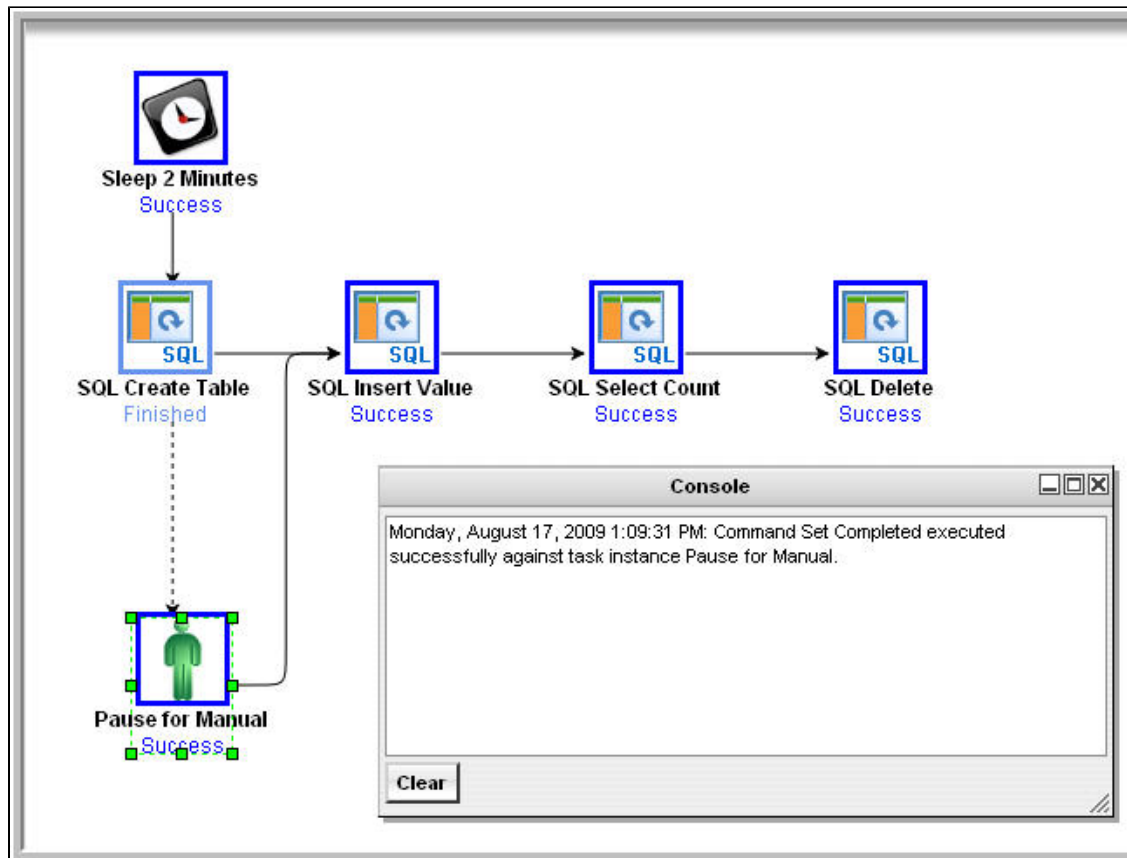


Note that we are still on the same day so the task fails again because the table already exists. In a real processing situation, the task would go to Success and the subsequent tasks would run as expected.

4. Set the Manual task to Complete status. For the purposes of our exercise, we will get the workflow going again by setting the Manual task to complete.
  - a. Right-click the Manual task.
  - b. Select **Commands > Set Completed**.



The Manual task goes to Success and the rest of the workflow runs.



- c. Because we set a two-minute Late Finish flag on our Manual task and we (presumably) took more than two minutes to complete it, the Manual task has been flagged as a late finish. To view the flag, go to the Activity screen and click the Manual task name. The details of the task display. Note that an enabled Finished Late field now displays. The Duration field indicates the duration of the task.

Status:	SUCCESS	Duration:	4 Minutes 16 Seconds
Status Description:			
Start Time:	2012-02-06 19:04:52 -0500	Shortest Estimated End Time:	2012-02-06 19:07:16 -0500
End Time:	2012-02-06 19:09:08 -0500	Longest Estimated End Time:	2012-02-06 20:03:28 -0500
User Estimated End Time:	2012-02-06 20:34:52 -0500	Late Finish Type:	Duration
Average Estimated End Time:	2012-02-06 19:35:22 -0500	Late Finish Duration:	00:02:00 m:mm:ss
Late Finish:	<input checked="" type="checkbox"/>		
Finished Late:	<input checked="" type="checkbox"/>		

Update View Parent Show Details Re-run Delete



---

For additional information, see:

- [Activity display](#)
- [Monitoring Workflows](#)
- [Database Connection](#)
- [SQL Task](#)
- [Manual Task](#)
- [Email Notification Actions](#)
- [Creating Notes](#)
- [Creating Conditional Paths](#)
- [Adding Skip/Run Criteria](#)

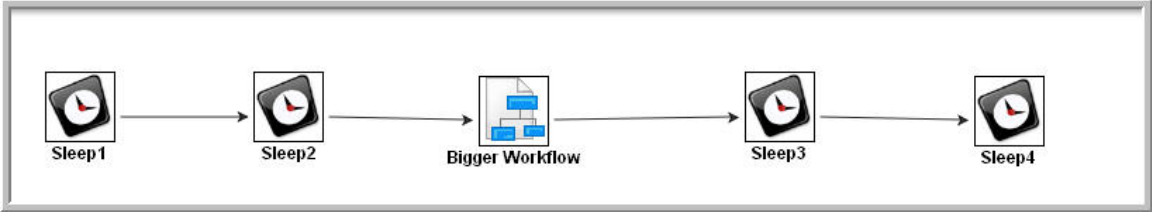

## Tutorial - Running a Workflow with Skipped Criteria

- [Introduction](#)
- [Create the Workflow](#)
- [Run the Workflow](#)
- [Check the Skipped Workflow's History](#)

### Introduction

In this exercise, we will create a daily workflow that includes a task that we want to skip on Fridays. We will also include a workflow within a workflow and later check the skipped workflow's status information.

### Create the Workflow

<b>Step 1</b>	<p>Create a new workflow called Workflow with Skipped, access the Workflow Editor and add the following tasks created in previous tutorials:</p> <ul style="list-style-type: none"> <li>• Sleep1, Sleep2, Sleep3, Sleep4 (See the <a href="#">Creating a Simple Workflow</a> tutorial.)</li> <li>• Bigger Workflow (See the <a href="#">Running a Workflow with a Conditional Path</a> tutorial.)</li> </ul>
<b>Step 2</b>	<p>Organize the workflow using all Success conditions as shown in the following illustration.</p>  <pre> graph LR     Sleep1[Sleep1] --&gt; Sleep2[Sleep2]     Sleep2 --&gt; BiggerWorkflow[Bigger Workflow]     BiggerWorkflow --&gt; Sleep3[Sleep3]     Sleep3 --&gt; Sleep4[Sleep4]   </pre>
<b>Step 3</b>	Click Save 
<b>Step 4</b>	Right-click Bigger Workflow and select <b>View/Edit Run Criteria</b> . Opwise Controller navigates to the Run Criteria list.
<b>Step 5</b>	Click <b>New</b> . The Task Run Criteria screen displays.
<b>Step 6</b>	In the Type field, select <b>Skip Criteria</b> .
<b>Step 7</b>	Select <b>Specific Day(s)</b> .

**Step 8** Select the current day (that is, if today is Friday, select Fri).

(See [Task Run Criteria Field Descriptions](#) for more details.)

**Step 9** Click **Submit**.

## Run the Workflow

**Step 1** Right-click anywhere in the Workflow Editor and select **Launch Workflow**.

**Step 2** Display the Activity screen and note that the sub-workflow (Bigger Workflow) and all its sub-tasks have been skipped, as shown in the following example. Note also that this did not impact subsequent tasks, which ran as usual.

Today's Task Instances by Created Time						New Report	Edit Report
Instance Name	Type	Status	Start Time	End Time	Dur		
<a href="#">Pause for Manual</a>	Manual	Skipped		2009-08-27 11:26:07 -0700			
<a href="#">Sleep 2 Minutes</a>	Sleep	Skipped		2009-08-27 11:26:07 -0700			
<a href="#">SQL Create Table</a>	SQL	Skipped		2009-08-27 11:26:07 -0700			
<a href="#">SQL Delete</a>	SQL	Skipped		2009-08-27 11:26:07 -0700			
<a href="#">SQL Insert Value</a>	SQL	Skipped		2009-08-27 11:26:07 -0700			
<a href="#">SQL Select Count</a>	SQL	Skipped		2009-08-27 11:26:07 -0700			
<a href="#">Bigger Workflow</a>	Workflow	Skipped		2009-08-27 11:26:07 -0700			
<a href="#">Sleep4</a>	Sleep	Success	2009-08-27 11:26:37 -0700	2009-08-27 11:26:47 -0700	10 S		
<a href="#">Sleep3</a>	Sleep	Success	2009-08-27 11:26:27 -0700	2009-08-27 11:26:37 -0700	10 S		
<a href="#">Sleep2</a>	Sleep	Success	2009-08-27 11:26:17 -0700	2009-08-27 11:26:27 -0700	10 S		
<a href="#">Sleep1</a>	Sleep	Success	2009-08-27 11:26:07 -0700	2009-08-27 11:26:17 -0700	10 S		
<a href="#">Workflow with Skipped</a>	Workflow	Success	2009-08-27 11:26:07 -0700	2009-08-27 11:26:48 -0700	41 S		

## Check the Skipped Workflow's History

You can view the history of a task to find out why, for example, it has a status of Skipped. This information is available from the Activity screen for normal tasks by clicking on the task. For workflows, clicking on the workflow from the Activity screen displays the Workflow Monitor. Therefore, to view a workflow's history, use the **Task Instances** screen.

**Step 1** From the navigation pane, select **Task Instances > Task Instances**.

**Step 2** Locate the Bigger Workflow task with the Skipped status that you just ran and click the name to access the record. It should resemble the following:

The screenshot displays the 'Workflow Task Instance' form with the following fields and values:

Instance Name:	Bigger Workflow	Invoked By:	Manually Launched
Task:	Bigger Workflow	Member of Business Services:	
Instance Reference Id:	2		
Hold Reason:			
Task Description:			
Status:	SKIPPED		
Status Description:	Skipped due to run/skip criteria.		
Start Time:		Duration:	2 Minutes 10 Seconds
End Time:	2009-08-27 11:26:07 -0700		
User Estimated End Time:		Shortest Estimated End Time:	
Average Estimated End Time:		Longest Estimated End Time:	
Show/Hide Skipped Tasks:	Hide Skipped		
Virtual Resource Priority:	10	Hold Resources on Failure:	<input type="checkbox"/>

Buttons at the top: Update, Show Details, View Workflow, Delete. Buttons at the bottom: Update, View Parent, Show Details, View Workflow, Delete.

Note that the Status Description field indicates that the workflow was skipped due to run/skip criteria.

For additional information, see:

- [Activity display](#)
- [Monitoring Workflows](#)
- [Adding Skip/Run Criteria](#)



## Tutorial - Using Variables in a Simple Task

- Introduction
- Resolving Variable Using Value from Global Variable Table
- Resolving Variable Using Value from Task
- Resolving Variable Using Value from Trigger

### Introduction

**Note**

You need an [Email Connection](#) to perform this exercise.

In the [Launching an Email Task Based on a File Monitor](#) tutorial, a built-in variable called `${ops_trigger_name}` and a system variable called `${_date}` were included to pass information into an Email message. Those variables were resolved using system information when the email task instance was created.

In this exercise, we will create a new user-defined variable, use it in a task, and run the task both manually and via a trigger to illustrate how such variables are resolved.

### Resolving Variable Using Value from Global Variable Table

<b>Step 1</b>	From the navigation pane, select <b>Variables</b> and click <b>New</b> .
<b>Step 2</b>	Give the variable the name <b>Tutorial</b> and a value of <b>Global</b> .
<b>Step 3</b>	Click <b>Submit</b> .

**Step 4** Create a new Email task with the following values:

- Task Name=Email Tutorial
- Email Connection=your Email connection
- To=your Email address
- Subject=Variable demo
- Body=\${Tutorial}

**Step 5** Save and launch the Email task.

## Resolving Variable Using Value from Task

- |               |   |
|---------------|---|
| <b>Step 1</b> | Open the Email Tutorial Task and click the <b>Variables</b> tab.                                |
| <b>Step 2</b> | Click <b>New</b> and add a variable called Tutorial, with a value of <b>Task</b> , and save it. |
| <b>Step 3</b> | Update and launch the task.   |
| <b>Step 4</b> | Check the Email and note that the body of the message now says "Task".                          |

## Resolving Variable Using Value from Trigger

<b>Step 1</b>	Create a new Time trigger with the following values: <ul style="list-style-type: none"><li>• Trigger Name=Variable Demo</li><li>• Tasks=Email Tutorial</li><li>• Time=a couple minutes from now</li></ul>
<b>Step 2</b>	Save the Time trigger.
<b>Step 3</b>	Click the <b>Variables</b> tab and add the Tutorial variable with a value of <b>Trigger</b> .
<b>Step 4</b>	Enable the trigger.
<b>Step 5</b>	When the Email task runs, check the email. The body of the Email now says Trigger.

---

For additional information, see:

- [User-Defined Variables](#)
- [Variables \(five-minute movie\)](#)

## Tutorial - Using Variables in a Workflow



### Note

You need a working [Database Connection](#) to do this exercise.

For tasks executing within a workflow, the order of precedence for [resolving user-defined variables](#) differs.

As the following exercise demonstrates, the variable definition in the task takes precedence, then Opwise Controller looks within the workflow or parent workflow(s), with the global variable coming last.

<b>Step 1</b>	Create a SQL task called <b>SQL With Variable</b> with the following SQL command. <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <pre>CREATE TABLE \${tutorial}\${_date("yyyyMMdd",5) } (name varchar(128), value varchar(128));</pre> </div>
<b>Step 2</b>	Within the task, define a variable called <b>tutorial</b> , with a value of <b>task</b> .
<b>Step 3</b>	Submit the SQL task.
<b>Step 4</b>	Create a new workflow called <b>Variable Workflow</b> .
<b>Step 5</b>	Add the SQL With Variable task and save the workflow.
<b>Step 6</b>	Launch the workflow and open the SQL With Variable task instance on the Activity screen. Note that the SQL command resembles the following, with the value from the task variable. <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <pre>CREATE TABLE task20090913 (name varchar(128), value varchar(128));</pre> </div>
<b>Step 7</b>	Open the task and delete the task variable.
<b>Step 8</b>	Go back to Variable Workflow and add the following variable: <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <pre>tutorial/workflow</pre> </div>
<b>Step 9</b>	Open the task instance. The SQL command used the variable from the workflow because the task no longer had a variable. <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <pre>CREATE TABLE workflow20090913 (name varchar(128), value varchar(128));</pre> </div>

For additional information, see:

- [User-Defined Variables](#)
- [Variables \(five-minute movie\)](#)

## Tutorial - Creating Custom Days and Periods

- Introduction
- Create a Custom Day
- Create a Custom Period
- Assigning Custom Day and Custom Period to a Calendar
- Selecting a Custom Calendar for a Trigger

### Introduction

In this exercise, we will create a custom day and period for a calendar, and assign that custom calendar to a trigger.

### Create a Custom Day

In this procedure, we will create a custom day, which can be applied to any calendar.

<b>Step 1</b>	From the navigation pane, select <b>Automation Center &gt; Custom Days</b> .
<b>Step 2</b>	On the Custom Days List screen, click New.
<b>Step 3</b>	<p>Create a custom day for Thanksgiving:</p> <ol style="list-style-type: none"> <li>1. Select Holiday.</li> <li>2. In the Type field, select Relative Repeating Date.</li> <li>3. In the When, Day of Week, and Month field, define Thanksgiving as the 4th Thursday in November.</li> </ol> <div data-bbox="300 982 1453 1516" data-label="Form"> <p>The screenshot shows the 'Custom Days' form with the following fields and values:</p> <ul style="list-style-type: none"> <li><b>Name:</b> Thanksgiving</li> <li><b>Version:</b> 1</li> <li><b>Period:</b> <input type="checkbox"/></li> <li><b>Holiday:</b> <input checked="" type="checkbox"/></li> <li><b>Description:</b> Thanksgiving in U.S.</li> <li><b>Type:</b> Relative Repeating Date</li> <li><b>When:</b> 4th</li> <li><b>Day Of Week:</b> Thu</li> <li><b>Month:</b> Nov</li> </ul> </div>
<b>Step 4</b>	Click Submit.

### Create a Custom Period

In this procedure, we will create a custom period of days, which can be applied to any calendar.

<b>Step 1</b>	From the navigation pane, select <b>Automation Center &gt; Custom Days</b> .
<b>Step 2</b>	On the Custom Days List screen, click New.

**Step 3** Create a custom period for the fourth quarter of the year:

1. Select Period.
2. In the Type field, select List of Dates.
3. In the Date field, select October 1, 2012 and click Add.
4. In the Date field, select December 31, 2012 and click Add.

**Step 4** Click Submit.

## Assigning Custom Day and Custom Period to a Calendar

In this procedure, we will assign the custom day and custom period to a new calendar.

**Step 1** From the navigation pane, select **Automation Center > Calendars**.

**Step 2** On the Calendars List screen, click New.

**Step 3** Name the calendar Custom Calendar #1 and add a description.

**Step 4** Access the Action menu and click **Save**.

**Step 5** Click the Custom Days tab.

**Step 6** On the Has Custom Days screen, click Edit.

**Step 7** Add Q4 Period and Thanksgiving to the Has Custom Days List.

Name	Thanksgiving
Description	Thanksgiving in U.S.
Period	false
Holiday	true

Use "Add Filter" and "Run Filter" to isolate the records to pick from

**Step 8** Click Save.



**Note**

You also can assign a custom day/period from the Custom Days Definition screen by clicking the Used by Calendars tab.

## Selecting a Custom Calendar for a Trigger

In this procedure, we will assign the custom calendar to a trigger.

**Step 1** From the navigation pane, click **Automation Center > Triggers > Time Triggers**.

**Step 2** On the Time Triggers List screen, click New.

**Step 3** On the Time Trigger Definition screen, name the trigger Custom Trigger.

**Step 4** Select a task for the trigger.

**Step 5** In the Calendar field, select Custom Calendar #1.

**Step 6** In the Day Style field, select Complex.

**Step 7** In the Date Noun drop-down list, select the custom day that you applied to Custom Calendar #1, Thanksgiving.

**Step 8** Access the Action menu and click **Save**.



**Step 9**

Click the List Qualifying Times button to see that the trigger will run the task every year on Thanksgiving.

## Trigger Name: Custom Trigger

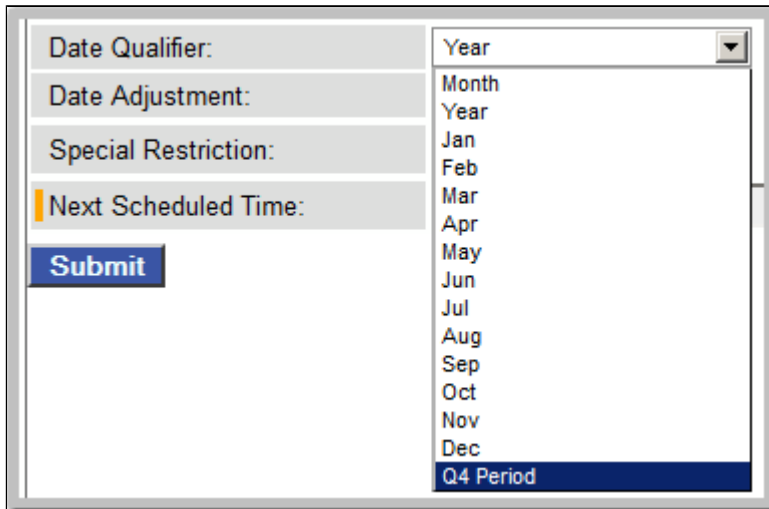
### Listing From "Tuesday, July 17, 2012 12:42:41 PDT -0700"

User/Trigger Timezone: US/Pacific
Thursday, November 22, 2012 00:00:00 PST -0800
Thursday, November 28, 2013 00:00:00 PST -0800
Thursday, November 27, 2014 00:00:00 PST -0800
Thursday, November 26, 2015 00:00:00 PST -0800
Thursday, November 24, 2016 00:00:00 PST -0800
Thursday, November 23, 2017 00:00:00 PST -0800
Thursday, November 22, 2018 00:00:00 PST -0800
Thursday, November 28, 2019 00:00:00 PST -0800
Thursday, November 26, 2020 00:00:00 PST -0800
Thursday, November 25, 2021 00:00:00 PST -0800
Thursday, November 24, 2022 00:00:00 PST -0800
Thursday, November 23, 2023 00:00:00 PST -0800
Thursday, November 28, 2024 00:00:00 PST -0800

**Step 10**

Set the Date Noun field to Business Day.

**Step 11** In the Date Qualifier drop-down list, select the custom period that you that you applied to Custom Calendar #1, Q4 Period.



The image shows a dialog box with several input fields and a dropdown menu. The fields are labeled: "Date Qualifier:", "Date Adjustment:", "Special Restriction:", and "Next Scheduled Time:". A blue "Submit" button is located below the "Next Scheduled Time:" field. The dropdown menu is open, showing a list of options: "Year", "Month", "Year", "Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec", and "Q4 Period". The "Q4 Period" option is highlighted with a dark blue background.

**Step 12** Access the [Action](#) menu and click **Save**.

**Step 13** Click the List Qualifying Times button to see that the trigger will run the task every business day during the custom period, October 1 to December 31.

## Trigger Name: Custom Trigger

### Listing From "Tuesday, July 17, 2012 13:55:09 PDT -0700"

User/Trigger Timezone: US/Pacific
Monday, October 01, 2012 00:00:00 PDT -0700
Tuesday, October 02, 2012 00:00:00 PDT -0700
Wednesday, October 03, 2012 00:00:00 PDT -0700
Thursday, October 04, 2012 00:00:00 PDT -0700
Friday, October 05, 2012 00:00:00 PDT -0700
Monday, October 08, 2012 00:00:00 PDT -0700
Tuesday, October 09, 2012 00:00:00 PDT -0700
Wednesday, October 10, 2012 00:00:00 PDT -0700
Thursday, October 11, 2012 00:00:00 PDT -0700
Friday, October 12, 2012 00:00:00 PDT -0700
Monday, October 15, 2012 00:00:00 PDT -0700
Tuesday, October 16, 2012 00:00:00 PDT -0700
Wednesday, October 17, 2012 00:00:00 PDT -0700

---

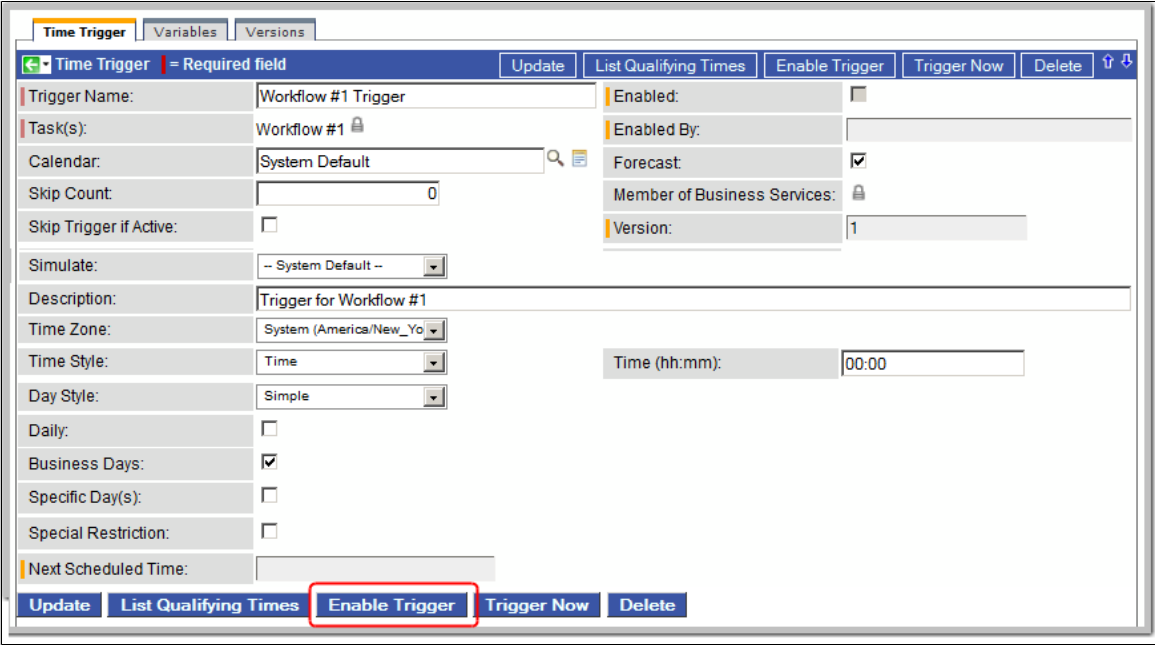
For additional information, see:

- [Creating Triggers](#)
- [Creating Calendars](#)
- [Creating Custom Days](#)
- [Custom Periods \(2 minute movie\)](#)

## Tutorial - Generating Forecast Data

In this exercise, we will:

- Add a workflow and its tasks to the Forecast calendar.
- Run the workflow and display its forecast information.
- Update a task in the workflow and then re-run the workflow.
- Recalculate the forecast information for the workflow.

<b>Step 1</b>	Create a workflow named Workflow #1 and add three tasks: Sleep 10, Sleep 30, and Sleep 60 that specify sleep times of 10, 30, and 60 seconds.
<b>Step 2</b>	<p>Create a Time trigger named Workflow #1 Trigger and:</p> <ul style="list-style-type: none"> <li>• Specify Workflow #1 as the trigger task.</li> <li>• Select the Business Days field.</li> <li>• Select the Forecast field.</li> </ul>
<b>Step 3</b>	<p>Access the <a href="#">Action menu</a> and click <b>Save</b>, and then click the <b>Enable Trigger</b> button. (A trigger must be enabled in order to generate forecast data for it.)</p>  <p>The screenshot shows the configuration page for a 'Time Trigger'. The 'Trigger Name' is 'Workflow #1 Trigger', 'Task(s)' is 'Workflow #1', and 'Calendar' is 'System Default'. The 'Forecast' checkbox is checked. The 'Enable Trigger' button at the bottom is highlighted with a red box.</p>

**Step 4**

From the navigation pane, select **Automation Center > Triggers > Forecast Calendar**. The Forecast Calendar identifies Workflow #1 on the days in the forecast period, for the current month, when it will be launched by Workflow #1 Trigger. (By default, tasks within a workflow are not displayed.)

Week	Sun	Mon	Tue	Wed	Thu	Fri	Sat
31	29	30	31	August 1	2	3	4
32	5	6	7	8	9	10	11
33	12	13	14	15	16	17	18
34	19	20	21 Workflow #1	22 Workflow #1	23 Workflow #1	24 Workflow #1	25
35	26	27 Workflow #1	28 Workflow #1	29 Workflow #1	30 Workflow #1	31 Workflow #1	September 1

**Step 5**

Hover your cursor over any Workflow #1 icon to display a pop-up of forecast and agent information for that workflow. Note that the Launch Time and End Time are identical; since the workflow has never been run, there is no data to support estimated times.

Week	Sun	Mon	Tue	Wed	Thu	Fri	Sat
31	29	30	31	August 1	2	3	4
32	5	6	7	8	9	10	11
33	12	13	14	15	16	17	18
34	19	20	21 Workflow #1	22 Workflow #1	23 Workflow #1	24 Workflow #1	25
35	26	27 Workflow #1	28 Workflow #1	29 Workflow #1	30 Workflow #1	31 Workflow #1	September 1

**Forecasts**

Task:	Workflow #1	Trigger:	Workflow #1 Trigger
Task Type:	Workflow	Launch Time:	2012-08-21 00:00:00 -0400
Workflow:		End Time:	2012-08-21 00:00:00 -0400
Member of Business Services:		Run Criteria Evaluation:	Run
Agent:		Agent Cluster:	
Agent Variable:		Agent Cluster Variable:	
		Cluster Broadcast:	

**Step 6**

From the navigation pane, select **Automation Center > Triggers > Forecast List**. The Forecast List identifies Workflow #1 and every task in Workflow #1, as well as their Launch Times and End Times, for every day in the forecast period when Workflow #1 will be launched by Workflow #1 Trigger.

Reports > Forecast - All - List with Run Criteria Evaluation

Forecasts 1 to 92 of 92

Trigger	Task	Task Type	Workflow	Launch Time	End Time	Run Criteria Evaluation
Workflow #1 Trigger	Workflow #1	Workflow		2012-08-21 00:00:00 -0400	2012-08-21 00:00:00 -0400	Run
Workflow #1 Trigger	Sleep 10	Sleep	Workflow #1	2012-08-21 00:00:00 -0400	2012-08-21 00:00:00 -0400	Run
Workflow #1 Trigger	Sleep 30	Sleep	Workflow #1	2012-08-21 00:00:00 -0400	2012-08-21 00:00:00 -0400	Run
Workflow #1 Trigger	Sleep 60	Sleep	Workflow #1	2012-08-21 00:00:00 -0400	2012-08-21 00:00:00 -0400	Run
Workflow #1 Trigger	Workflow #1	Workflow		2012-08-22 00:00:00 -0400	2012-08-22 00:00:00 -0400	Run
Workflow #1 Trigger	Sleep 10	Sleep	Workflow #1	2012-08-22 00:00:00 -0400	2012-08-22 00:00:00 -0400	Run
Workflow #1 Trigger	Sleep 30	Sleep	Workflow #1	2012-08-22 00:00:00 -0400	2012-08-22 00:00:00 -0400	Run
Workflow #1 Trigger	Sleep 60	Sleep	Workflow #1	2012-08-22 00:00:00 -0400	2012-08-22 00:00:00 -0400	Run
Workflow #1 Trigger	Workflow #1	Workflow		2012-08-23 00:00:00 -0400	2012-08-23 00:00:00 -0400	Run
Workflow #1 Trigger	Sleep 10	Sleep	Workflow #1	2012-08-23 00:00:00 -0400	2012-08-23 00:00:00 -0400	Run
Workflow #1 Trigger	Sleep 30	Sleep	Workflow #1	2012-08-23 00:00:00 -0400	2012-08-23 00:00:00 -0400	Run
Workflow #1 Trigger	Sleep 60	Sleep	Workflow #1	2012-08-23 00:00:00 -0400	2012-08-23 00:00:00 -0400	Run
Workflow #1 Trigger	Workflow #1	Workflow		2012-08-24 00:00:00 -0400	2012-08-24 00:00:00 -0400	Run
Workflow #1 Trigger	Sleep 10	Sleep	Workflow #1	2012-08-24 00:00:00 -0400	2012-08-24 00:00:00 -0400	Run
Workflow #1 Trigger	Sleep 30	Sleep	Workflow #1	2012-08-24 00:00:00 -0400	2012-08-24 00:00:00 -0400	Run
Workflow #1 Trigger	Sleep 60	Sleep	Workflow #1	2012-08-24 00:00:00 -0400	2012-08-24 00:00:00 -0400	Run
Workflow #1 Trigger	Workflow #1	Workflow		2012-08-27 00:00:00 -0400	2012-08-27 00:00:00 -0400	Run
Workflow #1 Trigger	Sleep 10	Sleep	Workflow #1	2012-08-27 00:00:00 -0400	2012-08-27 00:00:00 -0400	Run
Workflow #1 Trigger	Sleep 30	Sleep	Workflow #1	2012-08-27 00:00:00 -0400	2012-08-27 00:00:00 -0400	Run
Workflow #1 Trigger	Sleep 60	Sleep	Workflow #1	2012-08-27 00:00:00 -0400	2012-08-27 00:00:00 -0400	Run

**Step 7**

Return to the Workflow #1 Trigger screen and click Trigger Now to run Workflow #1.

Time Trigger Variables Versions

Time Trigger = Required field Update List Qualifying Times Disable Trigger Trigger Now Delete

Trigger Name:	Workflow #1 Trigger	Enabled:	<input checked="" type="checkbox"/>
Task(s):	Workflow #1	Enabled By:	ops.admin
Calendar:	System Default	Forecast:	<input checked="" type="checkbox"/>
Skip Count:	0	Member of Business Services:	<input type="checkbox"/>
Skip Trigger if Active:	<input type="checkbox"/>	Version:	1
Simulate:	-- System Default --		
Description:	Trigger for Workflow #1		
Time Zone:	System (America/New_Yo)	Time (hh:mm):	00:00
Time Style:	Time		
Day Style:	Simple		
Daily:	<input type="checkbox"/>		
Business Days:	<input checked="" type="checkbox"/>		
Specific Day(s):	<input type="checkbox"/>		
Special Restriction:	<input type="checkbox"/>		
Next Scheduled Time:	2012-08-21 00:00:00 -0400		

Update List Qualifying Times Disable Trigger **Trigger Now** Delete

**Step 8** When Workflow #1 has completed, recalculate the forecast information for Workflow #1: access the Action menu and click **Recalculate Forecast**.

**Note**  
 You also can recalculate the forecast, by the same method, from the Workflow #1 Workflow Task Definition screen.

**Step 9** Return to the Forecast Calendar and hover your cursor over any Workflow #1 icon. The pop-up of forecast and agent information for Workflow #1 now contains estimated Launch Time and End Time information based on the workflow run just completed.

**Step 10** Open the Sleep Task Definition screen for Sleep 10 and update the Sleep Time (secs) and Task Description fields from 10 to 50, and then click Update.

The screenshot shows the 'Sleep Task' configuration interface. The 'Task Name' is 'Sleep 60'. The 'Version' is '1'. The 'Sleep Type' is set to 'Duration'. The 'Task Description' is 'Sleep for 60 seconds.'. The 'Sleep Duration' is '00:01:00'. The 'Late Finish' checkbox is checked, and the 'Late Finish Duration' is '00:00:45'. The 'Virtual Resource Priority' is '10'. The 'First Time Ran' and 'Last Time Ran' are both '2013-09-20 08:29:31 -0700'. The 'Number of Instances' is '1'. Buttons for 'Update', 'Launch Task', 'View Instances', and 'Delete' are visible at the bottom.

**Step 11** Rerun Workflow #1.

**Step 12** From the Workflow #1 Trigger or Workflow #1 Definition screen, recalculate the forecast; since the updated Sleep Time for the Sleep 10 task affected the Workflow #1 End Time (the time it took to run the workflow), the Forecast information for Workflow #1 is now obsolete.

**Step 13** Re-open the Forecast Calendar and hover your cursor over any Workflow #1 task. The estimated run time, which based on the first run was 1 Minute 40 Seconds, is now 2 Minutes 0 Seconds, since the second run - in which 40 seconds was added to the Sleep 10 task - took 2 Minutes 20 seconds to run.

The screenshot shows the 'Forecast - Calendar' for August 2012. The calendar grid shows dates from 29 to 25. On August 20, 21, 22, and 23, there are icons for 'Workflow #1'. A 'Forecasts' panel is open, showing details for a 'Workflow #1' task. The 'Task' is 'Workflow #1', the 'Trigger' is 'Workflow #1 Trigger', the 'Launch Time' is '2012-08-21 00:00:00 -0400', and the 'End Time' is '2012-08-21 00:02:00 -0400'. The 'Run Criteria Evaluation' is 'Run'. The 'Agent' and 'Agent Variable' fields are empty. The 'Cluster Broadcast' field is also empty.

For additional information, see:



- [Creating and Maintaining Workflows](#)
- [Creating Triggers](#)
- [Displaying Trigger Forecast Information](#)
- [Forecasting \(3 minute movie\)](#)

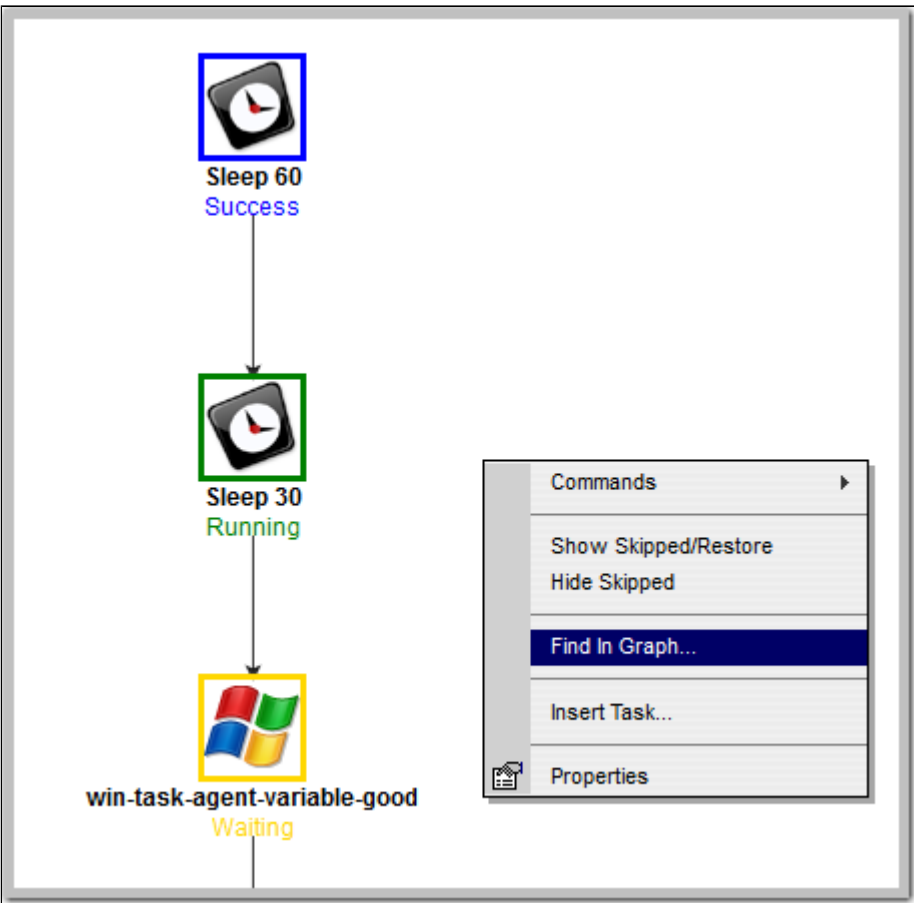
## Tutorial - Finding and Inserting Tasks in an Active Workflow

In this exercise, we will run a workflow and:

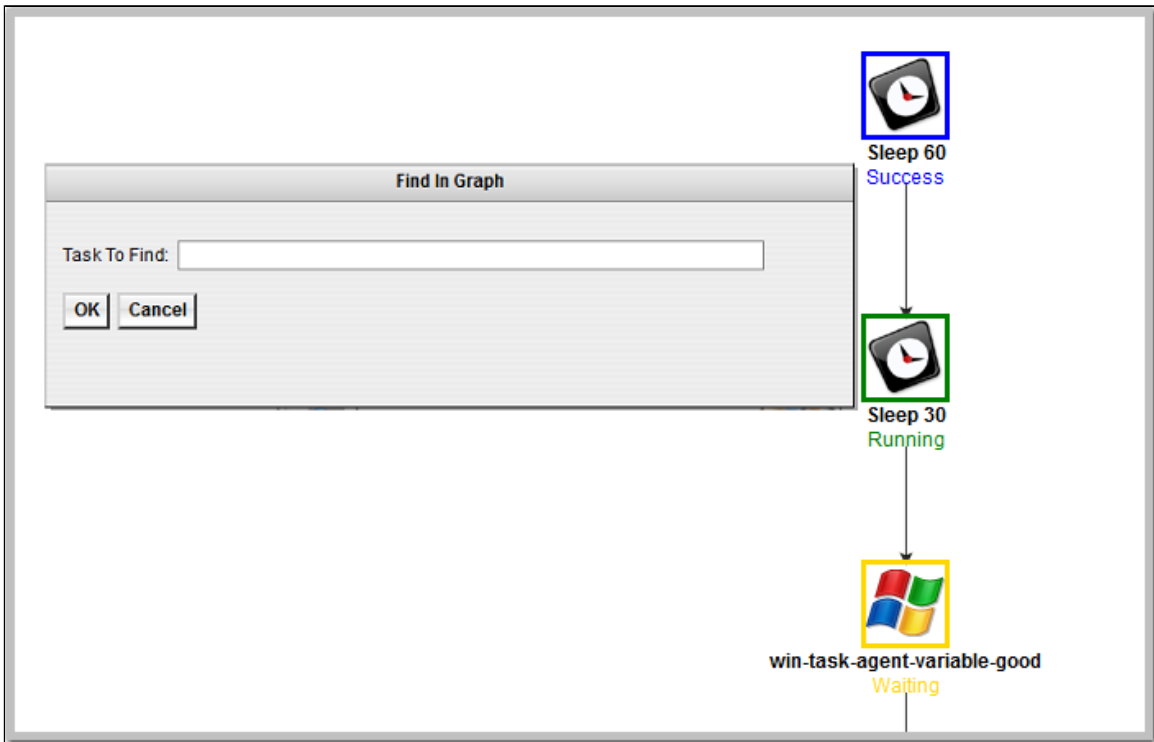
- Find a specific task within the workflow. (You also can find a task in a workflow that has not yet been launched or one that has run and completed.)
- Insert a task as a predecessor to another task in the workflow.
- Insert a task as a successor to another task in the workflow.

**Note**

You may find it easier to run a workflow three different times, once for each step in this exercise.

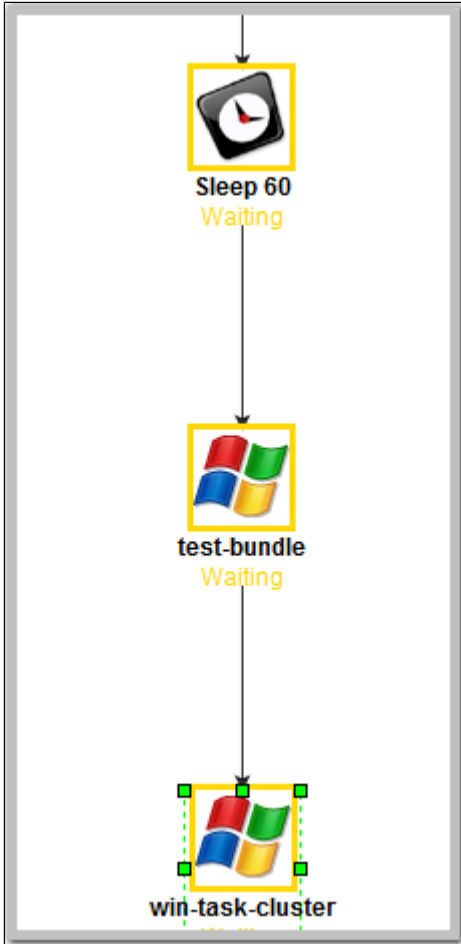
<b>Step 1</b>	Create a workflow so that all of its tasks cannot display on the Workflow Monitor at the same time.
<b>Step 2</b>	Right-click in the Workflow Monitor canvas. A pop-up menu displays. 

**Step 3** Click Find in Graph... to displays the Find in Graph dialog.

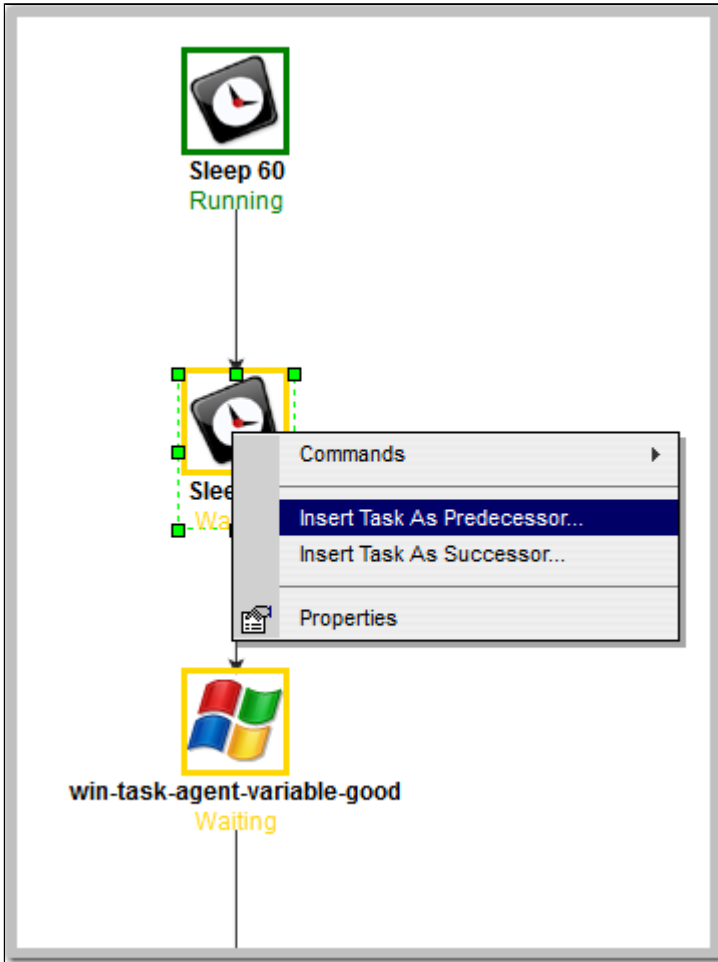


**Step 4**

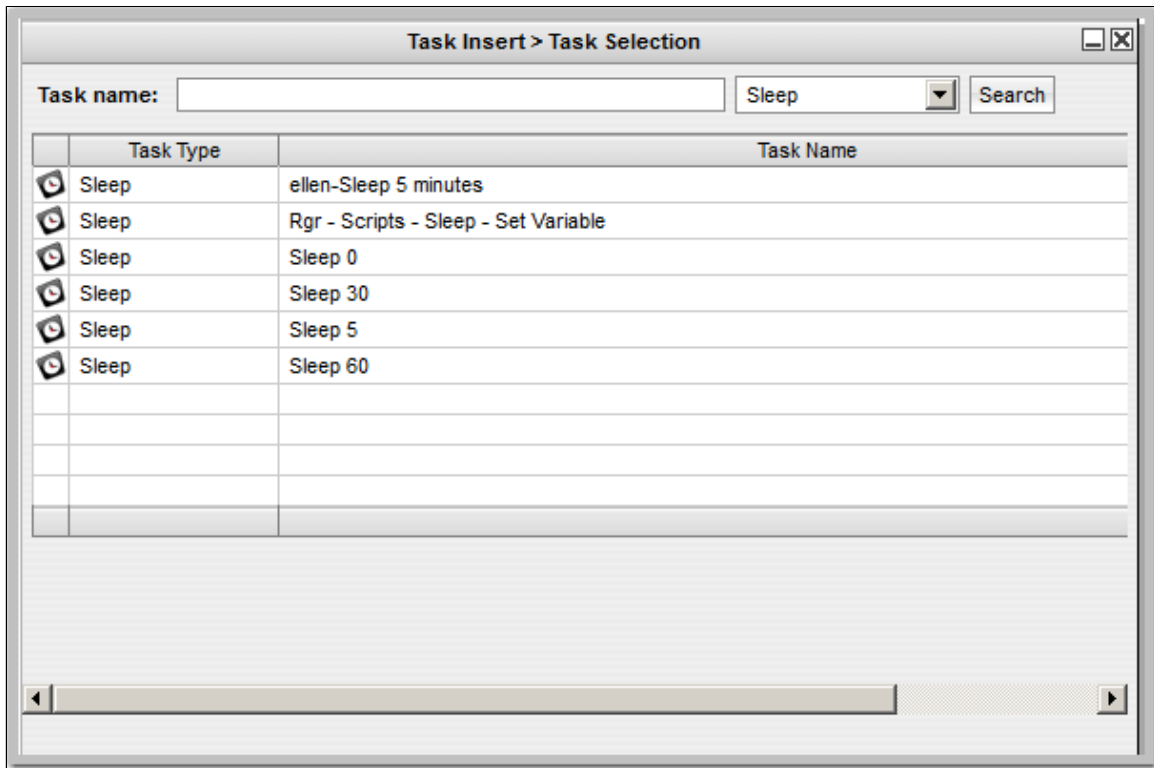
In the Task to Find field, enter the name of a task in the workflow that you want to find, and then click OK. Opswise Controller finds the task in the workflow, highlights the task, and moves the Workflow Monitor display so that the found task can be seen.



**Step 5** Right-click the Sleep 30 task in the Waiting status.



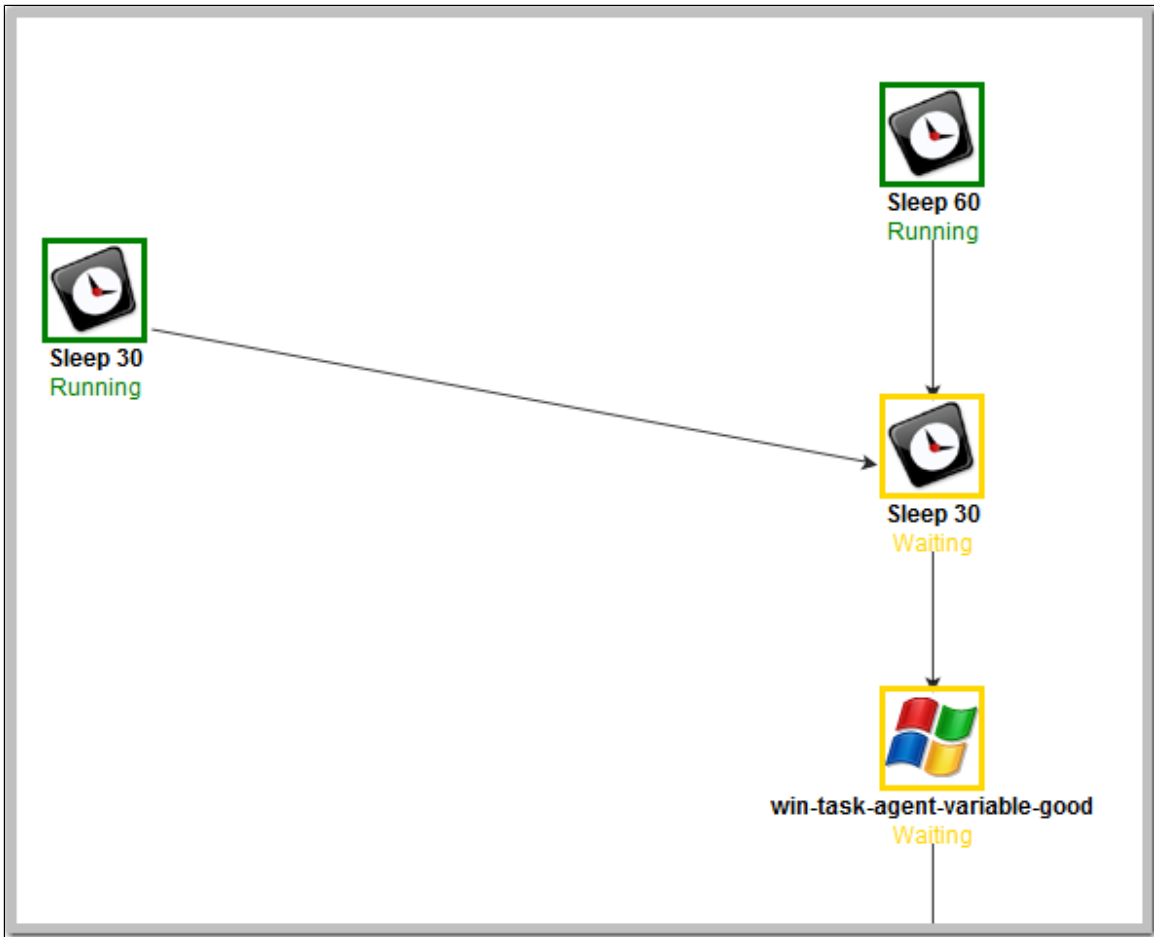
**Step 6** Click **Insert Task As Predecessor....** The Task Insert > Task Selection dialog displays.



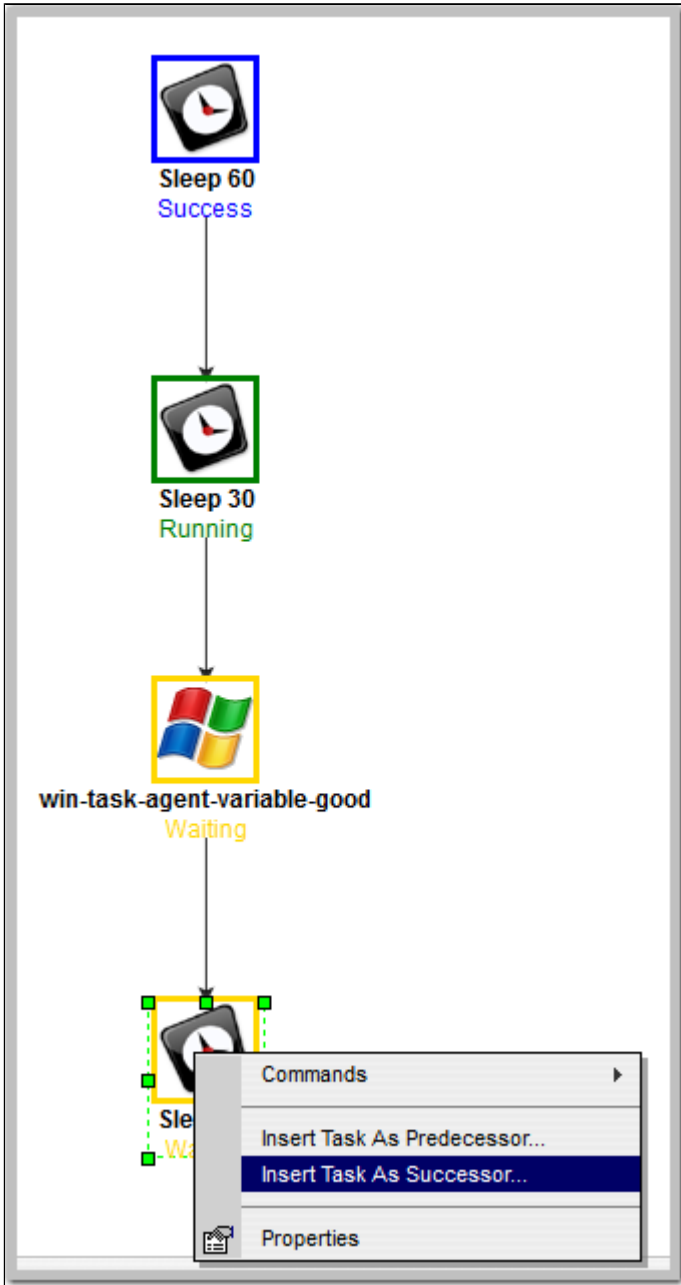
**Step 7** Find a task you want to insert and drag the task's icon to the workflow canvas. In this example, we will select the Sleep 30 task.

**Step 8**

The Controller inserts the Sleep 30 task as a precedent to the Waiting Sleep 30 task, and the inserted Sleep 30 task begins running. When the inserted Sleep 30 completes, the Sleep 30 in Waiting begins running.

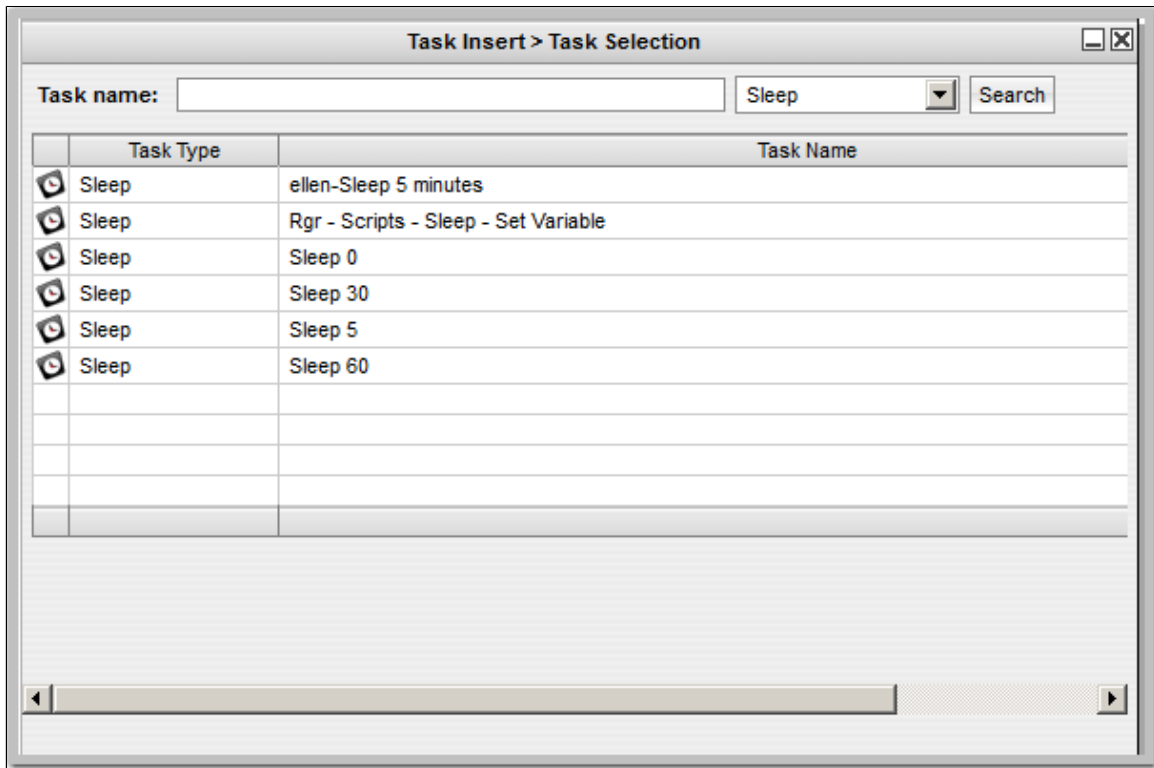


**Step 9** Right-click the Sleep 30 task in the Waiting status.



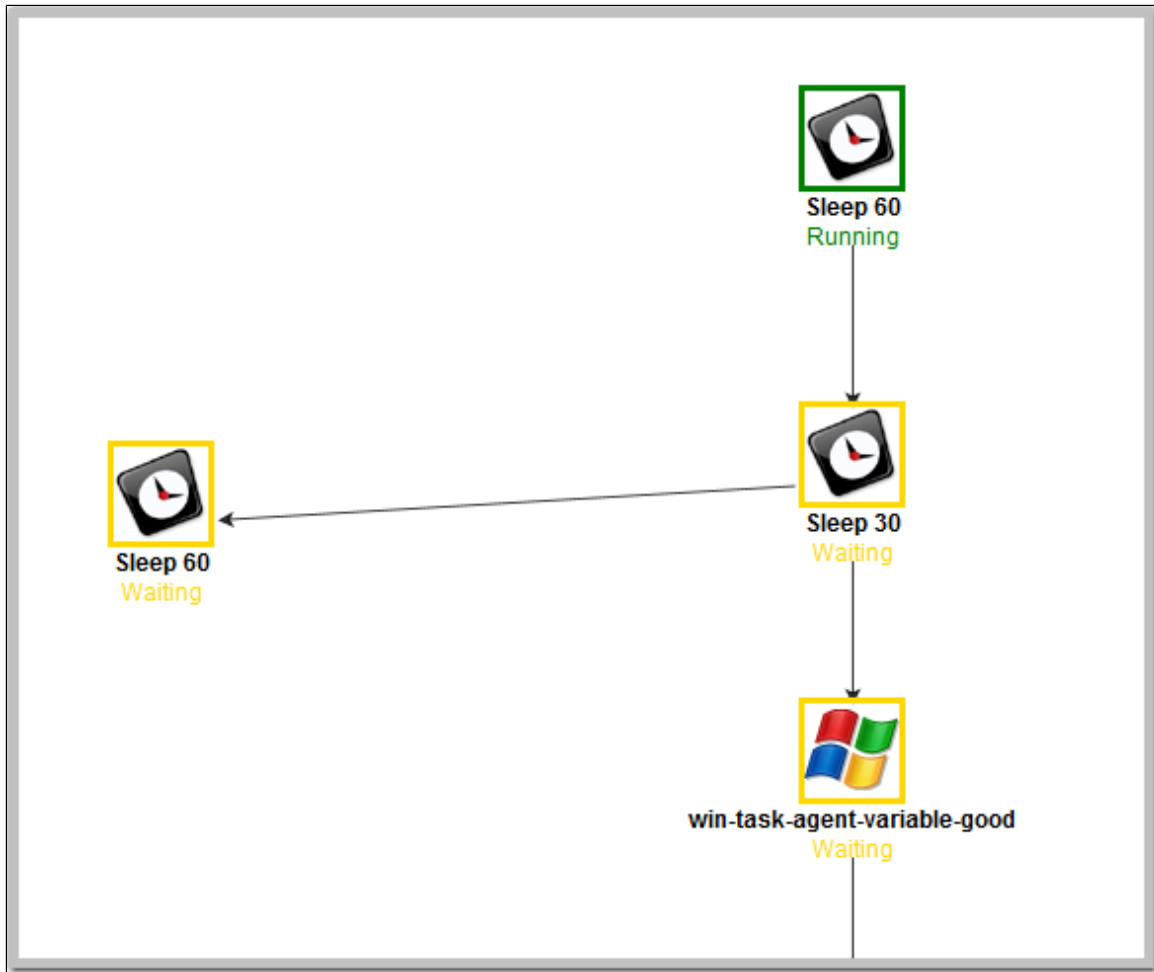


**Step 10** Select **Insert Task As Successor....** The Task Insert > Task Selection dialog displays.



**Step 11** Find the task you want to insert and drag the task's icon to the workflow canvas. In this example, we will select the Sleep 60 task.

**Step 12** The Controller inserts the Sleep 60 task as a successor to the Waiting Sleep 30 task. When Sleep 30 completes successfully, the inserted Sleep 60 task begins running.



For additional information, see:

- [Finding a Task in a Workflow](#)
- [Inserting a Task in a Workflow](#)
- [Searching for and Adding Tasks](#)
- [Workflow Improvements \(2 minute movie\)](#)

## Tutorial - Skipping, Unskipping, and Showing-Hiding Skipped Task Instances

You can skip (and unskip) individual task instances and task instances within a workflow that have been launched but have not yet started to run. For skipped tasks within a workflow, you can choose to show or hide those tasks in the Workflow Monitor.

You also can skip a task instance so that all dependent task instances of that task instance automatically are skipped as well.

Although there are several methods for skipping, unskipping, and showing/hiding skipped task instances, in this exercise we will:

- Skip a task instance in a workflow.
- Unskip a previously skipped task instance in a workflow.
- Show and hide a skipped task instance in a workflow.



### Note

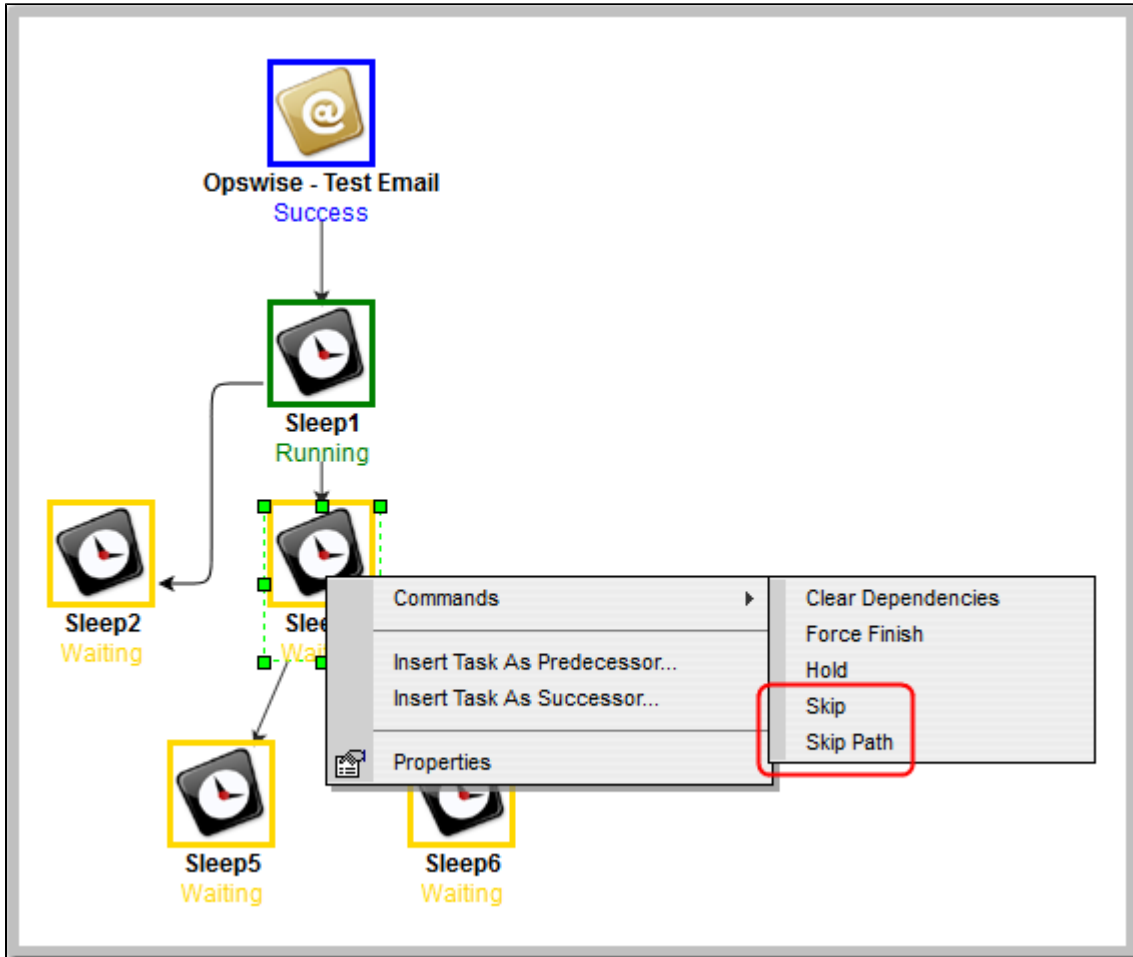
There also are methods for skipping a task and showing/hiding a skipped task before it becomes a task instance (that is, before it or the workflow in which it resides has been launched). See the links at the bottom of this page.

<b>Step 1</b>	From the navigation pane, select <b>Automation Center &gt; Tasks &gt; Workflow Tasks</b> . The Workflow Tasks List screen displays a list of all workflow tasks.
<b>Step 2</b>	Right-click <b>Simple Workflow</b> (created in the <a href="#">Creating a Simple Workflow</a> tutorial), and on the <b>Action menu</b> , click <b>Launch Task</b> .
<b>Step 3</b>	On the Activity screen, select Active Workflow Task Instances from the drop-down list.
<b>Step 4</b>	Click <b>Simple Workflow</b> . The Workflow Monitor displays for this running workflow.

**Step 5** Right-click the Sleep3 task while it is in Waiting status and, from the pop-up menu that displays, click Commands / Skip.

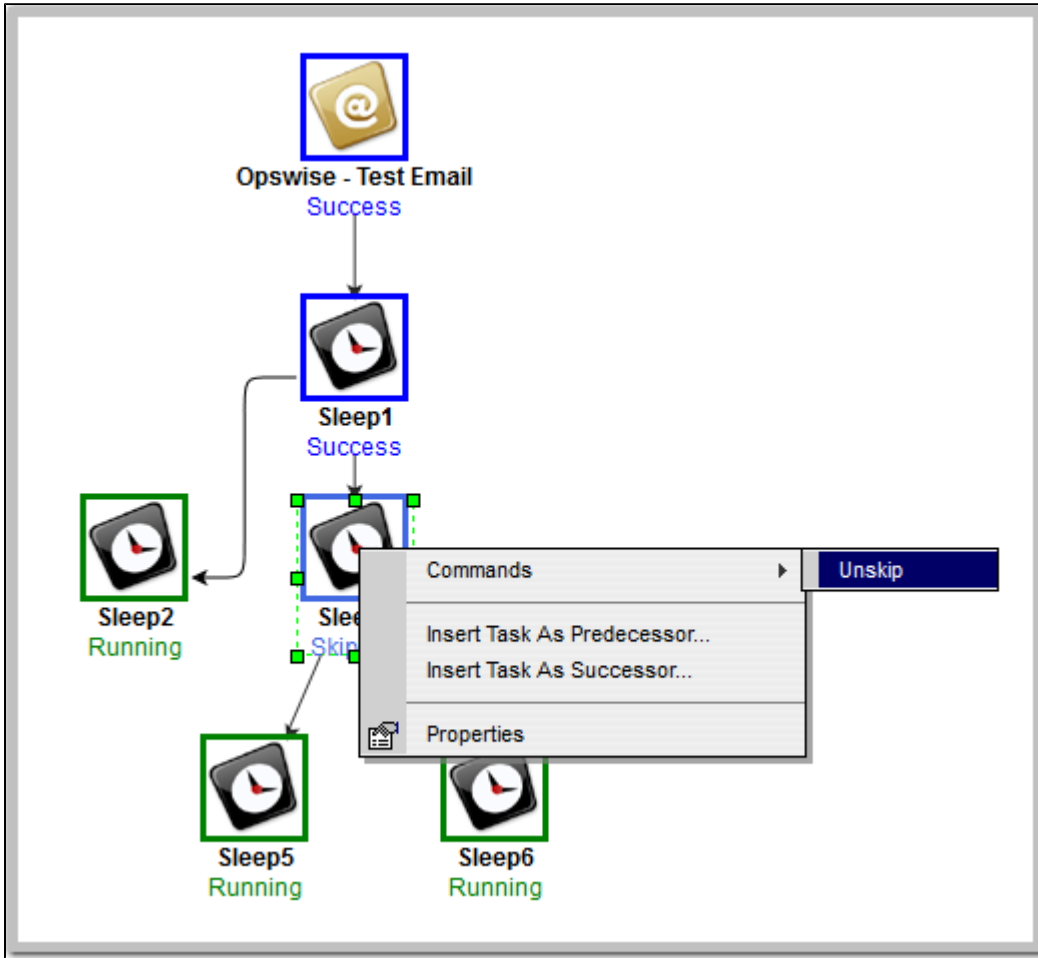
**Note**

If you wanted to skip the Sleep3 task and its dependent Sleep5 and Sleep6 tasks, you would click Commands / Skip Path.



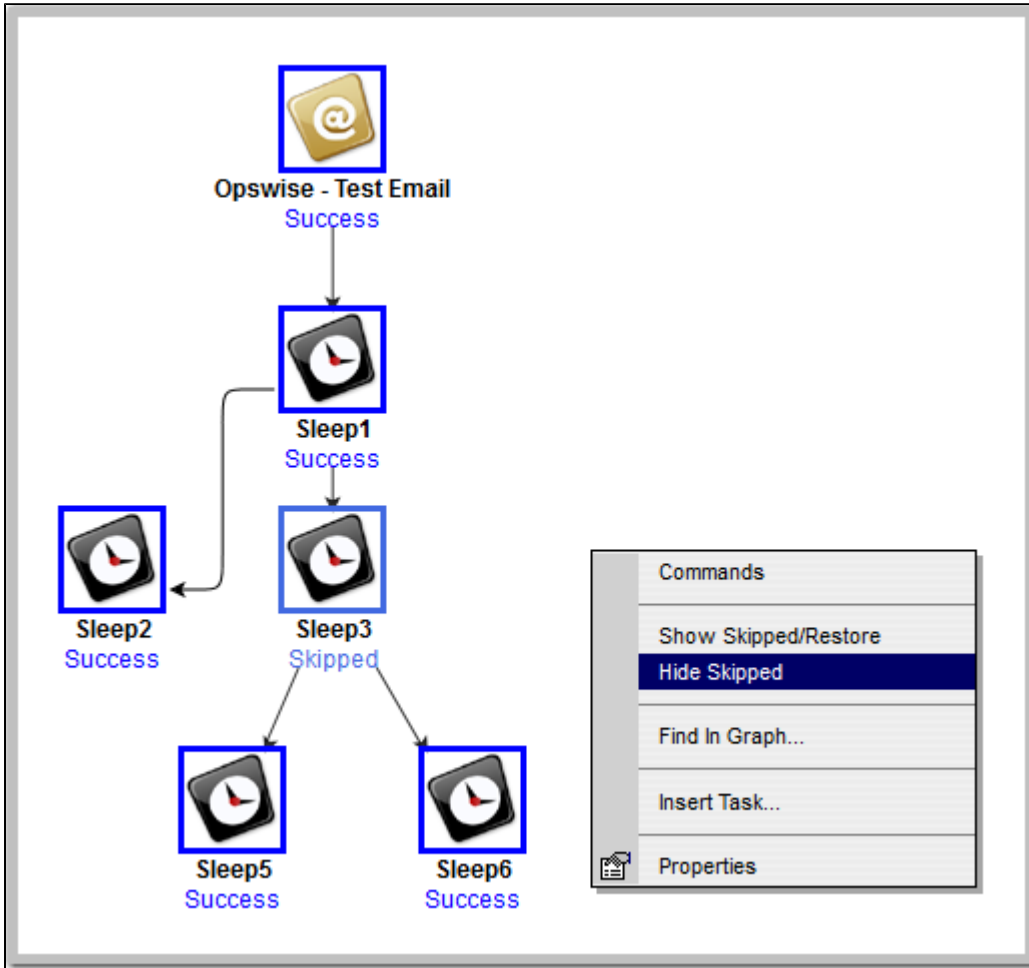
The Sleep3 status changes from Waiting to Skipped. When Sleep1 completes successfully, Opwise Controller will skip Sleep3 and start running Sleep5 and Sleep6.

**Step 6** To un-skip Sleep3, right-click it and, from the pop-up menu that displays, click Commands / Unskip.

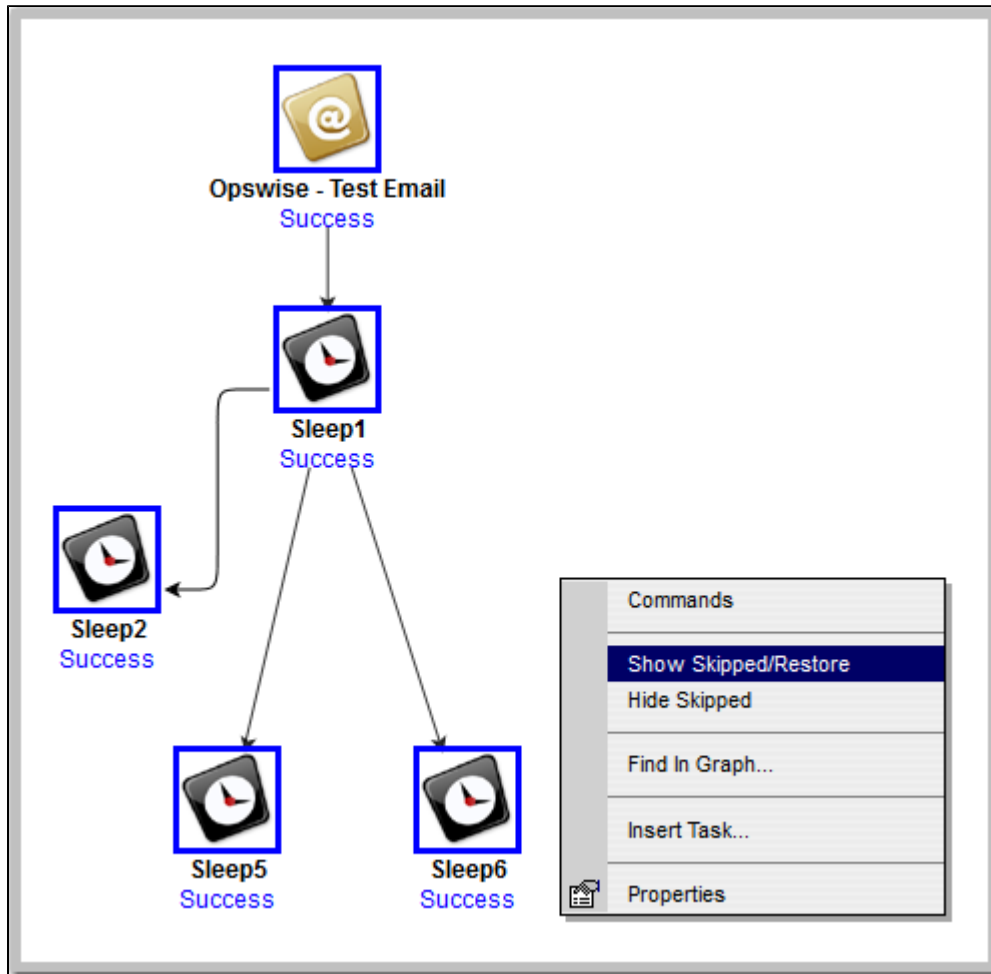


The task status changes from Skipped to Waiting. When Sleep1 completes successfully, the Controller will not skip Sleep3.

**Step 7** To hide the skipped Sleep3 task in the Workflow Editor, right-click in the Workflow Monitor canvas and click Hide Skipped.



**Step 8** To show the skipped Sleep3 task in the Workflow Monitor, right-click in the Workflow Editor canvas and click Show Skipped/Restore.



For additional information, see:

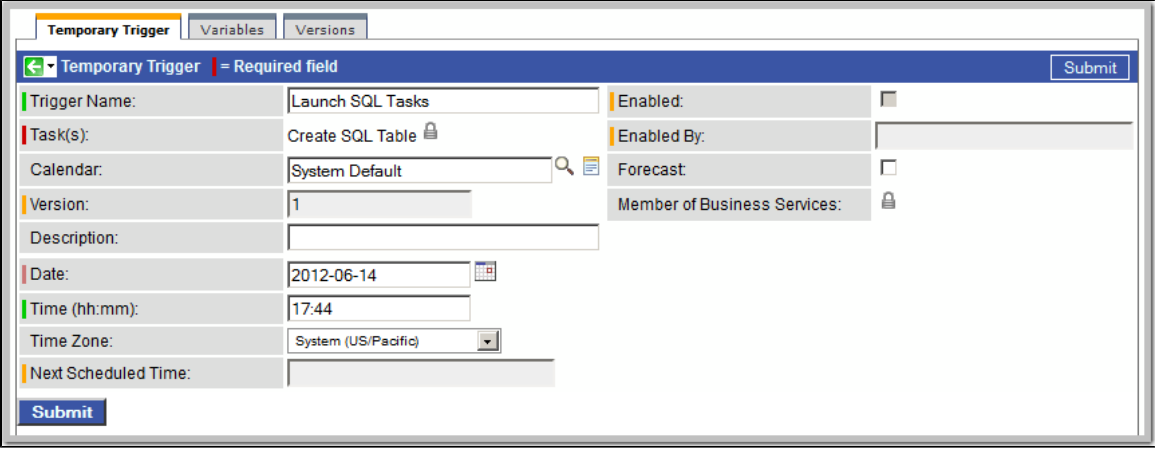
- [Skipping a Task](#)
- [Unskipping a Task](#)
- [Showing or Hiding Skipped Tasks](#)
- [Adding Skip/Run Criteria for Specific Tasks](#)
- [Workflow Improvements \(2 minute movie\)](#)

## Tutorial - Launching Tasks at a Future Date and Time

In this exercise, we will create a trigger that will launch multiple tasks at the same time in the future (in two minutes). For this exercise, we will use the SQL tasks created in the [Running a Workflow with a Conditional Path](#) tutorial.

(A future date also can be selected, but to see now that the tasks have been launched, keep the current date.)

### Create a Temporary Trigger

<b>Step 1</b>	From the navigation pane, select <b>Triggers &gt; Temporary Triggers</b> and click <b>New</b> .
<b>Step 2</b>	In the Trigger Name field, type <b>Launch SQL Tasks</b> .
<b>Step 3</b>	In the Tasks field, click the Lock. In the Search field (with the magnifying glass), type SQL. The four SQL tasks you created pop up. Select the first one, then repeat the process for the other three.
<b>Step 4</b>	In the Date field, leave the default set to the current date.
<b>Step 5</b>	In the Time field, enter a time two minutes in the future, using 24-hour time.
<b>Step 6</b>	In the Time Zone field, select your time zone.
	
<b>Step 7</b>	Access the <b>Action</b> menu and click <b>Save</b> .
<b>Step 8</b>	Click the <b>Enable Trigger</b> button.

### View the Activity Screen

Navigate to the Activity screen. At the designated time, the four tasks are loaded and run.

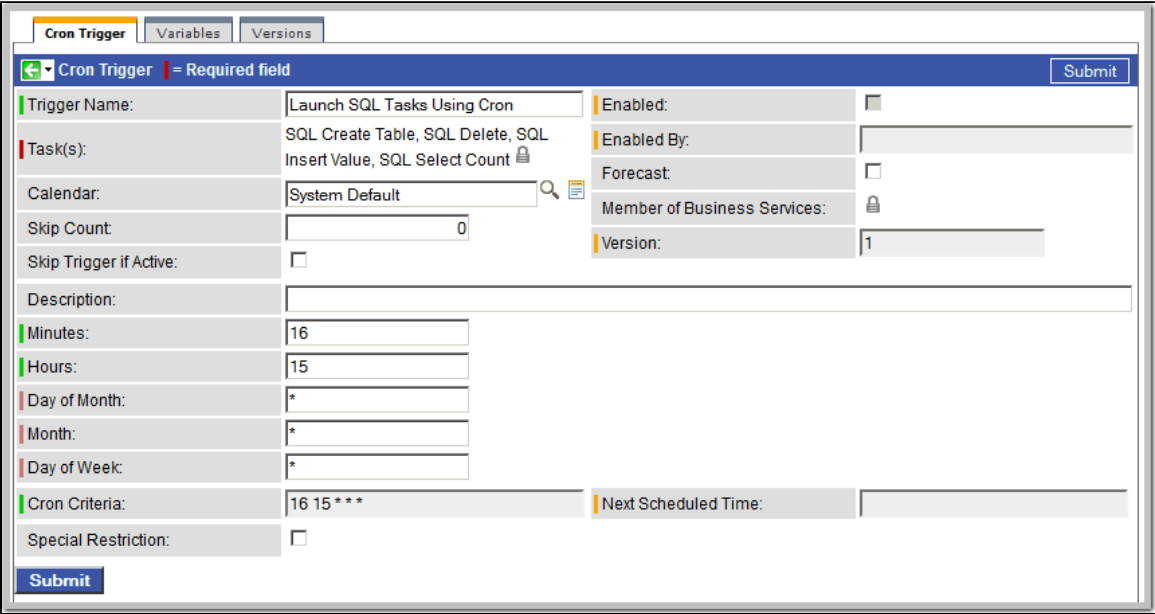
For additional information, see:

- [Temporary Trigger](#)



## Tutorial - Launching Tasks Using a Cron Trigger

### Create a Cron Trigger

<b>Step 1</b>	From the navigation pane, select <b>Triggers &gt; Cron Triggers</b> and click <b>New</b> .
<b>Step 2</b>	In the Trigger Name field, type <b>Launch SQL Tasks Using Cron</b> .
<b>Step 3</b>	In the Tasks field, click the Lock icon. In the Search field (with the magnifying glass), type <b>SQL</b> . The four SQL tasks created in the <a href="#">Running a Workflow with a Conditional Path</a> tutorial appear. Select the first one, then repeat the process for the other three.
<b>Step 4</b>	In the Minutes field, type the number of minutes past the hour you want the tasks to run. Use a time a few minutes from now. For example, for 3:16, type 16 in this field.
<b>Step 5</b>	In the Hours field, type the hour in 24-hour time that you want the tasks to run. For example, for 3:16, type 15 in this field. Opwise Controller uses the time zone of the Controller server.
<b>Step 6</b>	Keep the asterisks (*) in the remaining fields.
	
<b>Step 7</b>	Access the <a href="#">Action menu</a> and click <b>Save</b> .
<b>Step 8</b>	Click the <b>Enable Trigger</b> button.

### View the Activity Screen

Navigate to the Activity screen. At the designated time, the four tasks are loaded and run.

For additional information, see:

- [Cron Trigger](#)

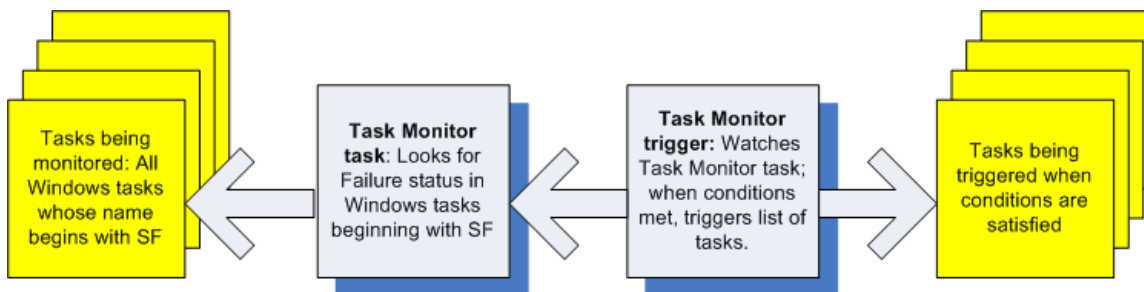
## Tutorial - Launching an Email Task Based on a Task Monitor

- Introduction
- Creating an Activity Screen Filter
- Creating an Email Template
- Creating an Email Task Using an Email Template
- Creating the Task Monitor Task
- Creating the Task Monitor Trigger
- Running the Task Monitor

### Introduction

In this exercise, we will set up a Task Monitor and Task Monitor trigger. The Task Monitor will monitor all tasks for a status that indicates some sort of problem. When the trigger is satisfied, Opwise Controller will launch an Email task that notifies a user that there is a problem. We will also create an Email template for use in our Email task and create an Activity screen filter that displays only problem tasks.

The following illustration shows the various components used to trigger tasks based on the status of other tasks.



### Creating an Activity Screen Filter

The first task we will do is create a filter for the Activity screen that will be used by our imaginary tech support person. When this user receives an email indicating there is a problem, he or she can view this Activity screen to find out more information. In the Controller, this filter is referred to as a report. For more detailed explanations of each field, see [Opwise Controller - Reports](#).

**Step 1** From the Navigation pane, select **Automation Center > Reports**. The main report screen displays.

This screen provides a list of all reports that your userid has access to, along with a New button that allows you to create a new report.

**Step 2** Under My Saved reports, click **New**. A blank reports screen displays.

Reports > New report

Run Report Save Insert Delete Make Gauge Schedule

Name:

Visible to: Me

Type: Bar chart

Table: Activity [ops\_exec]

Group by: Agent

Stacked Field: -- None --

Sum Field: -- None --

Chart size: Large

Other threshold: System Default

Filter and Order:

**Step 3** In the Name field, enter **Problem Task Instances**.

**Step 4** In the Visible to field, select **Everyone**.

**Step 5** In the Type field, select **List**.

**Step 6** In the Table field, select **Activity(ops\_exec)**, which is the table used for all Activity reports and contains information about all task activity.

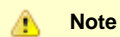
**Step 7** Skip the **Group by** field.

**Step 8** In the Filter and Order fields, click the plus sign and select the following:

1. In choose field, select **Status**.
2. In the operator field, select **is one of**.
3. In the last field, use CTRL+Click to select all of: **Undeliverable, Running/Problems, In Doubt, Start Failure, Cancelled, Failed**.

**Step 9** The Columns section displays a list of available fields in the Available section, and the selected fields in the Selected section. A number of default fields appear in the Selected section. For this tutorial, move the following fields to the Selected list in the order shown:

- Status
- Type
- Instance Name
- Invoked By
- Start Time
- End Time



**Note**

To select and deselect fields:

- On the Selected list, double-click fields to remove them from the report.
- On the Available list, double-click fields to add them to the report.

**Step 10** To save the report, click **Save**.

Reports > Problem Task Instances

Run Report Update Save Insert Delete Publish Make Gauge Schedule

Name: Problem Task Instances

Visible to: Everyone

Type: List

Table: Activity [ops\_exec]

Group by: -- None --

Filter and Order:

Columns

Available

- Agent (+)
- Attempt
- Average Estimated End Time
- CPU Time
- Calendar (+)
- Class
- Created
- Created by

Selected

- Status
- Type
- Instance Name
- Invoked By
- Start Time
- End Time

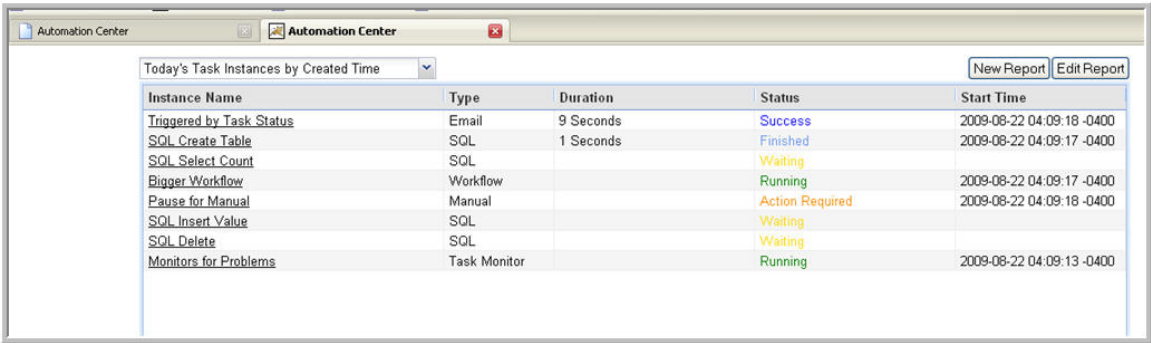
Filter and Order: Status is one of CANCELLED, FAILED, SKIPPED, FINISHED

Now we'll open a new tab containing only our new Activity screen:

**Step 1** Open a new browser tab.

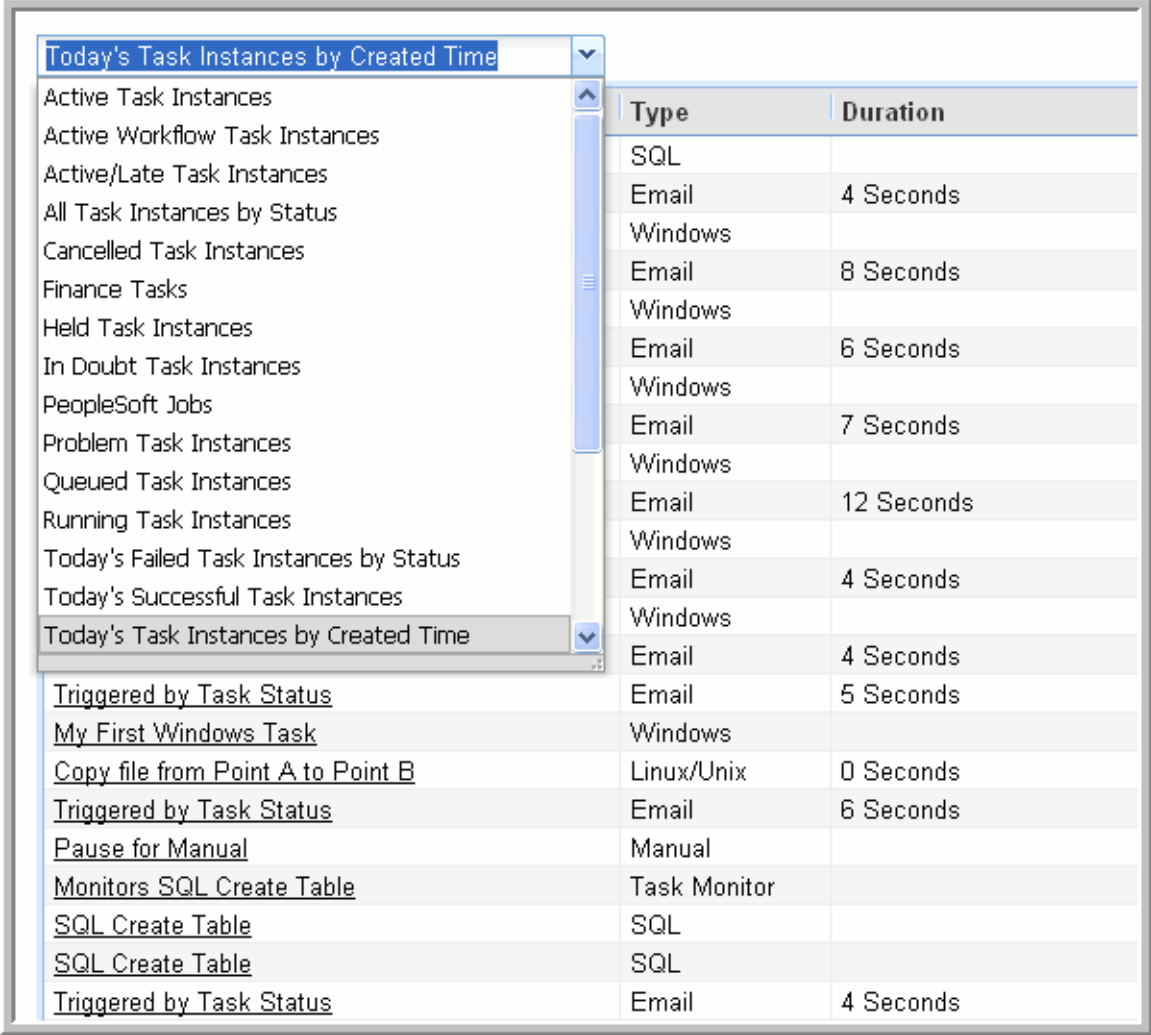
**Step 2** From the first tab, click and drag the Activity option to the new tab.

**Step 3** Click the new tab. Note that the new tab contains only the Activity screen, without the navigation pane and other application options.



Instance Name	Type	Duration	Status	Start Time
Triggered by Task Status	Email	9 Seconds	Success	2009-08-22 04:09:18 -0400
SQL Create Table	SQL	1 Seconds	Finished	2009-08-22 04:09:17 -0400
SQL Select Count	SQL		Waiting	
Bigger Workflow	Workflow		Running	2009-08-22 04:09:17 -0400
Pause for Manual	Manual		Action Required	2009-08-22 04:09:18 -0400
SQL Insert Value	SQL		Waiting	
SQL Delete	SQL		Waiting	
Monitors for Problems	Task Monitor		Running	2009-08-22 04:09:13 -0400

**Step 4** Click the scroll bar at the top that shows the current report name.



Type	Duration
SQL	
Email	4 Seconds
Windows	
Email	8 Seconds
Windows	
Email	6 Seconds
Windows	
Email	7 Seconds
Windows	
Email	12 Seconds
Windows	
Email	4 Seconds
Windows	
Email	4 Seconds
Email	5 Seconds
Windows	
Linux/Unix	0 Seconds
Email	6 Seconds
Manual	
Task Monitor	
SQL	
SQL	
Email	4 Seconds

**Step 5** Select the report you just created, **Problem Task Instances**. Your Activity screen will now show only problem task instances, sorted by status (the leftmost column).

Status	Type	Instance Name	Invoked By
Cancelled	Manual	<a href="#">Pause for Manual</a>	Workflow: Bigger Workflow
Cancelled	Manual	<a href="#">Pause for Manual</a>	Workflow: Bigger Workflow
Failed	SQL	<a href="#">SQL Create Table</a>	Manually Launched
Failed	SQL	<a href="#">SQL Create Table</a>	Manually Launched
Failed	SQL	<a href="#">SQL Create Table</a>	Manually Launched
Failed	SQL	<a href="#">SQL Create Table</a>	Manually Launched
Failed	SQL	<a href="#">SQL Create Table</a>	Manually Launched
Failed	SQL	<a href="#">SQL Create Table</a>	Workflow: Bigger Workflow2
Failed	SQL	<a href="#">SQL Delete</a>	Workflow: Bigger Workflow
Failed	SQL	<a href="#">Opwise - SQL Create</a>	Manually Launched
Failed	FTP	<a href="#">Opwise - FTP Download</a>	Workflow: A Workflow with UNIX and Stored Procedures
Failed	Linux/Unix	<a href="#">Linux Failure Exit Code 10</a>	Workflow: Opwise - Conditional
Failed	Linux/Unix	<a href="#">Linux Failure Exit Code 10</a>	Workflow: Opwise - Conditional
Failed	Linux/Unix	<a href="#">Linux Failure Exit Code 10</a>	Workflow: Opwise - Conditional

## Creating an Email Template

Email Templates allow you to create a "canned" Email task that you can then refer to when creating an Email task. This is useful if you have a large number of common parameters on Email tasks but still require separate tasks.

**Step 1** From the navigation pane, select **Automation Center Resources > Email Templates**.

**Step 2** For Template Name, enter **Notification based on status**.

**Step 3** For Email Connection, select the connection you defined for earlier exercises.

**Step 4** For the To field, enter a usable email account that you can use for the tutorial.

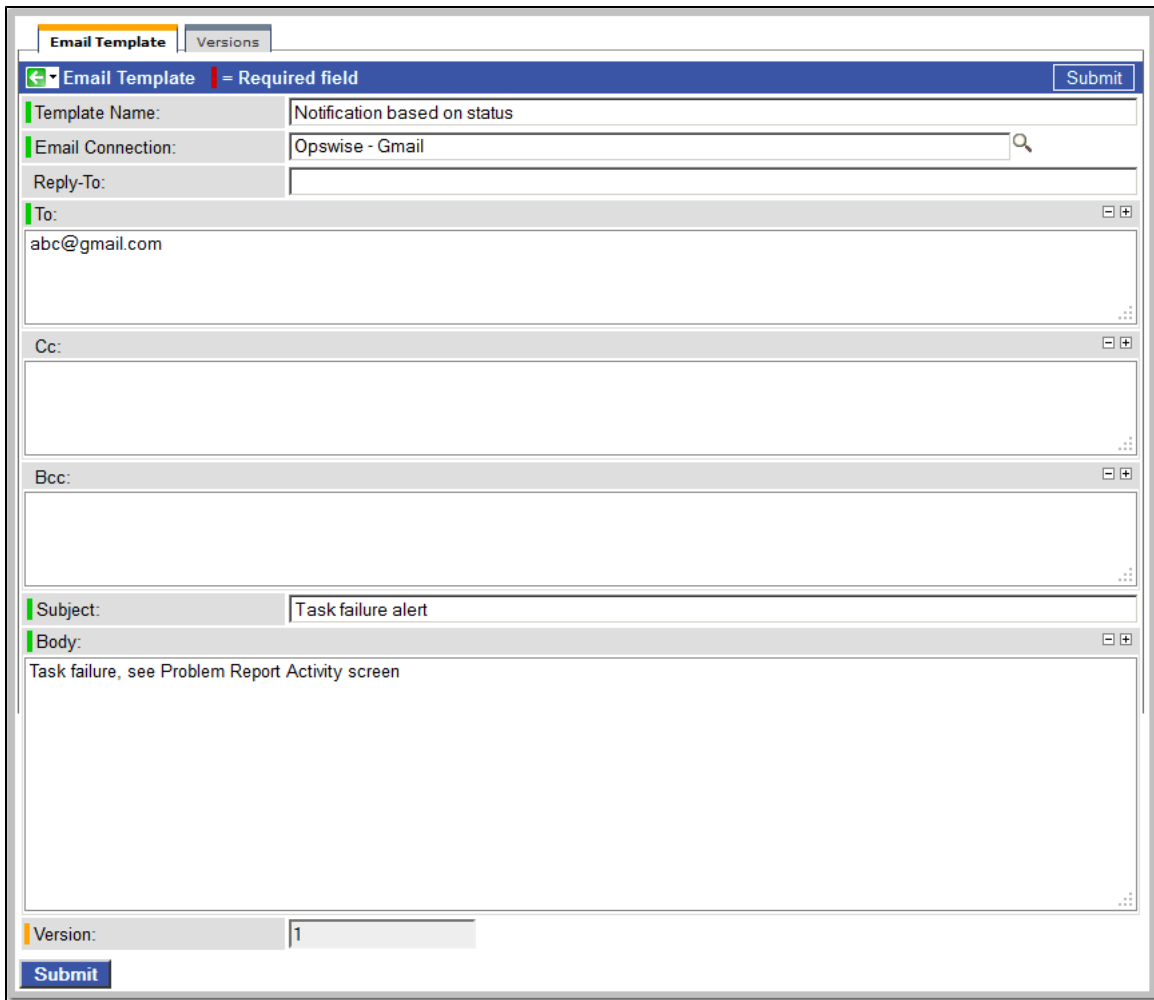
**Step 5** For Subject, enter:

Task failure alert

**Step 6** For Body, enter:

Task failure, see Problem Report Activity screen

**Step 7** Click **Submit**. Your Email Template should resemble the following sample.



The screenshot shows the 'Email Template' configuration page. At the top, there are tabs for 'Email Template' and 'Versions'. Below the tabs is a blue header bar with a back arrow, the text 'Email Template', a red exclamation mark icon, and the text '= Required field', and a 'Submit' button on the right. The form fields are as follows:

- Template Name:** Notification based on status
- Email Connection:** Opwise - Gmail
- Reply-To:** (empty)
- To:** abc@gmail.com
- Cc:** (empty)
- Bcc:** (empty)
- Subject:** Task failure alert
- Body:** Task failure, see Problem Report Activity screen
- Version:** 1

A 'Submit' button is located at the bottom left of the form.

## Creating an Email Task Using an Email Template

**Step 1** From the navigation pane, select **Tasks > Email Tasks** and click **New**.

**Step 2** In the Task Name field, type **Triggered by Task Status**.

- Step 3** In the Email Template field, select the Email Template you just created, **Notification based on status**. For our purposes, we will leave the remaining fields blank since we want to use the information from the template. If you fill in any of the duplicate fields, the information from the task overrides the information from the template.

The screenshot shows the 'Email Task' configuration window. The 'Task Name' is 'Triggered by Task Status' and the 'Email Template' is 'Notification based on status'. The 'Version' is set to 1. The 'Task Description' is empty. The 'Member of Business Services' is locked. The 'Email Connection' is empty. The 'Reply-To', 'To', 'Cc', 'Bcc', 'Subject', and 'Body' fields are empty. There are checkboxes for 'Late Start', 'Late Finish', and 'Early Finish', all of which are unchecked. The 'Virtual Resource Priority' is set to 10. The 'Hold Resources on Failure' checkbox is unchecked. A 'Submit' button is located at the bottom left of the form.

- Step 1** Click **Submit**.

## Creating the Task Monitor Task

For this Task Monitor, we will monitor the status of all other tasks. If any task has a status that indicates there is some sort of problem, we will generate an email. Once the Task Monitor is launched by the Task Monitor trigger, it remains active, launching an Email every time any of its conditions are met. The Task Monitor task remains active until the Task Monitor trigger is disabled or until a user manually stops it.

- |               |  |
|---------------|--|
| <b>Step 1</b> | From the navigation pane, select *Tasks > Task Monitors *, and click *New *.   |
| <b>Step 2</b> | In the Task Name field, type <b>Monitors for Problems</b> .  |
| <b>Step 3</b> | In the Status area, enable the following statuses: *Undeliverable, Running/Problems, In Doubt, Start Failure, Cancelled, Failed.* (For a description of each status, see <a href="#">Task Status Descriptions</a> ). |
| <b>Step 4</b> | In Monitoring Type, select <b>General Tasks</b> . Note that you can also select <b>Specific Task</b> and browse for a task name.   |
| <b>Step 5</b> | In Task type to Monitor, we will leave all task types selected for our exercise.   |

**Step 6** Click **Submit**.

**Task Monitor** = Required field Submit

Task Name: Monitoring for Problems Member of Business Services:

Version: 1

Task Description:

Status To Monitor:

- DEFINED
- HELD
- RESOURCE WAIT
- UNDELIVERABLE
- SUBMITTED
- STARTED
- RUNNING/PROBLEMS
- START FAILURE
- CANCELLED
- SKIPPED
- SUCCESS
- WAITING
- RESOURCE REQUESTED
- EXECUTION WAIT
- QUEUED
- ACTION REQUIRED
- RUNNING
- IN DOUBT
- CONFIRMATION REQUIRED
- FAILED
- FINISHED

Monitoring Type: General Task(s)

Task Name To Monitor Condition: All Tasks

Task Type To Monitor:

- Workflow
- Windows
- File Monitor
- Email
- Sleep
- FTP File Monitor
- Stored Procedure
- System Monitor
- SAP
- Linux/Unix
- Manual
- Task Monitor
- z/OS
- File Transfer
- SQL
- Indesca
- Application Control

Workflow Condition: -- None --

Time Scope: -- None --

Late Start:

Late Finish:

Early Finish:

Virtual Resource Priority: 10

Hold Resources on Failure:

Submit

## Creating the Task Monitor Trigger

As the last step in our setup process, we will create the Task Monitor trigger, which controls when the Task Monitor task is started and stopped.

- |               |  |
|---------------|--|
| <b>Step 1</b> | From the navigation pane, select <b>Triggers &gt; Task Monitor Triggers</b> and click <b>New</b> .   |
| <b>Step 2</b> | In Trigger Name, type <b>Controls Monitors for Problems</b> .  |
| <b>Step 3</b> | In the Task Monitor field, browse to the Task Monitor task you just created (Monitors for Problems). |



- Step 4** In the Tasks field, click on the lock icon and type a **T** in the browse field. This will jump to a list of tasks whose names begin with "T". (Click [here](#) for more information about selecting records.)

The screenshot shows the 'Task Monitor Trigger' configuration window. The 'Task(s):' field is open, displaying a list of tasks starting with 'T'. The selected task is 'Triggered by Task Status'. The interface includes fields for Trigger Name, Task Monitor, Calendar, Skip Count, Skip Trigger if Active, Description, Restrict Times, and Special Restriction. Buttons for Update, Enable Trigger, Trigger Now, and Delete are visible at the bottom.

- Step 5** Select **Triggered by Task Status**, which you created earlier.

- Step 6** Access the [Action](#) menu and select **Save**.

- Step 7** Click the **Enable Trigger** button.

The screenshot shows the 'Task Monitor Trigger' configuration window after completion. The 'Task(s):' field is now 'Triggered by Task Status'. The 'Enabled' checkbox is checked. The 'Enabled By' field is 'ops.admin'. The 'Version' field is '2'. Buttons for Update, Disable Trigger, Trigger Now, and Delete are visible at the bottom.

## Running the Task Monitor

To test our setup, we need to run a task to one of the failure statuses that will trigger the email. To do so, we will launch the Manual task we created earlier and force it into Failed status.

- Step 1** From the navigation pane, select **Tasks > Manual Tasks**.
- Step 2** Right-click the Pause for Manual task and, on the [Action](#) menu, select **Launch Task**.
- Step 3** Display the Activity screen with the Today's Task Instances by Created Times filter.

**Step 4** First note that the Monitor for Problems task is running. It was launched when you enabled the trigger.

Today's Task Instances by Created Time				
Instance Name	Type	Status	Start Time	End
<a href="#">Pause for Manual</a>	Manual	Action Required	2012-07-10 14:16:21 -0400	
<a href="#">Monitors for Problems</a>	Task Monitor	Running	2012-07-10 13:58:42 -0400	

**Step 5** Locate the **Pause for Manual** task instance, which will be in Action Required status.

**Step 6** Right-click on **Pause for Manual**. This displays a list of commands that are currently available for this task instance.

Today's Task Instances by Created Time				
Instance Name	Type	Status	Start Time	End
<a href="#">Pause for Man</a>	Manual	Action Required	2012-07-10 14:16:21 -0400	
<a href="#">Monitors for Pr</a>	Task Monitor	Running	2012-07-10 13:58:42 -0400	

- Cancel
- Force Finish
- Force Finish/Cancel
- Set Started
- Set Completed

**Step 7** Select **Cancel**. Pause for Manual goes to Cancelled status and an Email task is launched by the Task Monitor trigger.

**Step 8** Check the Email account where you sent the notification.

**Task failure alert** Inbox | X

---

☆ from **opswise.test@gmail.com** [hide details](#) 4:26 PM (0 minutes ago) [Reply](#)  
 to abc@gmail.com  
 date Thu, Aug 20, 2002 at 4:26 PM  
 subject Task failure alert  
 mailed-by gmail.com  
 signed-by gmail.com

Task failure, see Problem Report Activity screen

[Reply](#) [Forward](#) [Invite opswise.test@gmail.com to chat](#)

**Step 9** Once the user receives the email, he or she can quickly check for more information by looking at the Activity screen using the Cancelled Task Instances filter and clicking on the Instance Name of the problem task. As shown in the illustration below, additional information about the issue is displayed in the Status Description field.

Manual Task Instance		Update	Show Details	Force Finish	Re-run	Delete
Instance Name:	Pause for Manual	Invoked By:	Manually Launched			
Task:	Pause for Manual	Member of Business Services:				
Instance Reference Id:	1					
Hold Reason:						
Task Description:	A Manual task run at 2012-07-10 14:16:21 -0400					
Status:	CANCELLED					
Status Description:	State was cancelled from ACTION REQUIRED to CANCELLED					
Start Time:	2012-07-10 14:16:21 -0400	Duration:				
End Time:	2012-07-10 14:30:04 -0400					
User Estimated End Time:						
Average Estimated End Time:	2012-07-10 14:16:21 -0400	Shortest Estimated End Time:				
Late Finish:	<input checked="" type="checkbox"/>	Longest Estimated End Time:				
Finished Late:	<input checked="" type="checkbox"/>	Late Finish Type:	Duration			
Virtual Resource Priority:	10	Late Finish Duration:	00 : 02 : 00 hh:mm:ss			
		Hold Resources on Failure:	<input type="checkbox"/>			
		Update	Show Details	Force Finish	Re-run	Delete

To view the status description and other information about a workflow instance, use **Task Instances > Task Instances**.

For additional information, see:

- Email Task
- Email Connection
- Email Template
- Opwise Controller - Reports
- Activity Table
- Activity display
- Task Monitor Task
- Task Monitor Trigger
- Command Quick Reference
- Cancelling a Task Run
- Modifying Activity Screen Reports (2 minute movie)

## Tutorial - Force Finishing, Force Finish-Cancelling, and Cancelling a Task

In this exercise, we will force finish, force finish/cancel, and cancel tasks within a workflow from three areas:

- Workflow Monitor
- Activity screen
- Task Instances screen

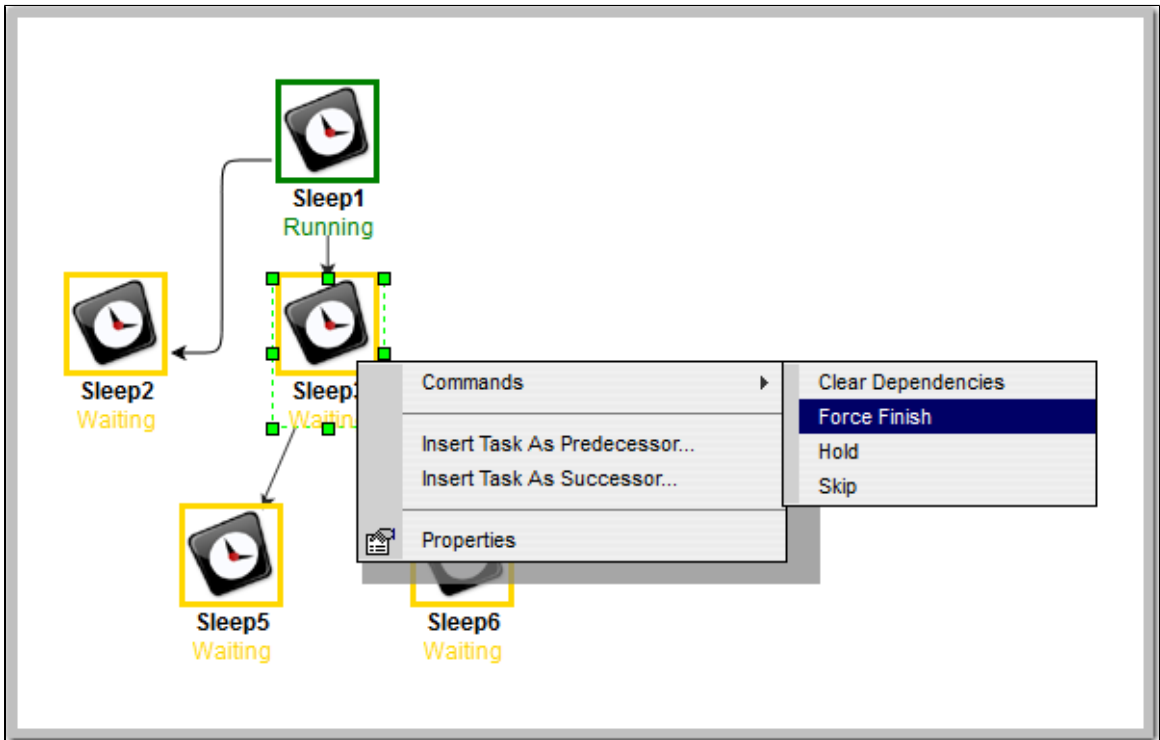
You can run any of these three commands from any of these three screens. For stand-alone tasks, you can run these commands only from the Activity and Task Instances screens.



### Note

You can force finish, force finish/cancel, and cancel any task in Running status, but you only can force finish a task in Waiting status. See [Manually Running and Controlling Tasks](#) for a complete list of task statuses for each command.

<b>Step 1</b>	From the navigation pane, select <b>Automation Center &gt; Tasks &gt; Workflow Tasks</b> .
<b>Step 2</b>	On the Workflow Tasks List screen, right-click <b>Simple Workflow</b> , which you created in the <a href="#">Creating a Simple Workflow</a> tutorial, to display an <b>Action</b> menu.
<b>Step 3</b>	Click <b>Launch Task</b> to run the workflow.
<b>Step 4</b>	<p>Force Finish a task from the Workflow Monitor:</p> <ol style="list-style-type: none"> <li>1. From the navigation pane, select <b>Automation Center &gt; Task Instances &gt; Task Instances</b> to display the Activity screen which displays, by default, a list of Active Task Instances.</li> <li>2. Click Simple Workflow to display its Workflow Monitor.</li> <li>3. Right-click a Waiting task and, from the list of Commands on the pop-up menu, click Force Finish. The status of the task changes from Waiting to Finished, and all successor task instances waiting for successful completion of this task instance will start.</li> </ol>



**Step 5** Force Finish/Cancel a task from the Activity screen:

1. Return to the Activity screen.
2. Right-click a Simple Workflow task that is in the Running status and, on the **Action** menu, click Force Finish/Cancel. The status of the task changes from Running to Finished, and all successor task instances waiting for successful completion of this task instance will start.

Instance Name	Type	Status	Start Time	End
Simple Workflow	Workflow	Running	2012-07-20 06:55:22 -0700	
Sleep	Sleep	Running	2012-07-20 06:55:24 -0700	
Sleep	Sleep	Waiting		
Sleep	Sleep	Waiting		
Sleep	Sleep	Waiting		
Sleep	Sleep	Waiting		
Opswise - Test Email	Email	Success	2012-07-20 06:55:22 -0700	2012

**Step 6** Cancel a task from the Task Instances screen:

1. From the navigation pane, select **Automation Center > Task Instances > Task Instances** to display the Task Instances screen.
2. Right-click a Simple Workflow task in the Running status and, on the **Action** menu, click Cancel. The status of the task changes from Running to Cancelled, and all successor task instances waiting for successful completion of this task will remain in the Waiting status. The status of Simple Workflow changes from Running to Running/Problems.

Instance Name	Instance Reference Id	Type	Status	Invoked By	Agent	S
Sleep6	13	Sleep	WAITING	Workflow: Simple Workflow		
Sleep5	13	Sleep	WAITING	Workflow: Simple Workflow		
Sleep	13	Sleep	RUNNING	Workflow: Simple Workflow		2012 06:55
Sleep	13	Sleep	RUNNING	Workflow: Simple Workflow		2012 06:55
Sleep	13	Sleep	SUCCESS	Workflow: Simple Workflow		2012 06:55
Simple Workflow	13	Workflow	RUNNING	Manually Launched		2012 06:55
Opswise - Test Email	6	Email	SUCCESS	Workflow: Simple Workflow		2012 06:55
Ops	5	Unix	SUCCESS	Manually Launched	server.opswise.com - AGNT0002	2012 10:24
Ops	1	Unix	FAILED	Manually Launched	server.opswise.com - AGNT0002	2012 10:14
MR	6	Manual	SUCCESS	Manually Launched		2012 07:17

For additional information, see:

- [Force Finishing a Task](#)
- [Cancelling a Task](#)
- [Force Finish / Cancelling a Task](#)
- [Monitoring Activity from the Activity Screen](#)
- [Monitoring Activity from the Task Instances Screen](#)
- [Monitoring Workflows](#)

## Tutorial - Selecting Widgets for the Home Page

- Introduction
- Removing Widgets
- Adding Widgets
- Setting the Refresh Rate

### Introduction

The home page is associated with a login ID. It is the first page you see when you log in to Opwise Controller.

Shown below is a sample home page (your home page may be different). Each window on the page is a widget (filter, gadget, gauge, label, scroller, system application, or world clock) containing a specific set of information.

In this exercise, we are going to remove all of the widgets from the home page and add new widgets of the same type.

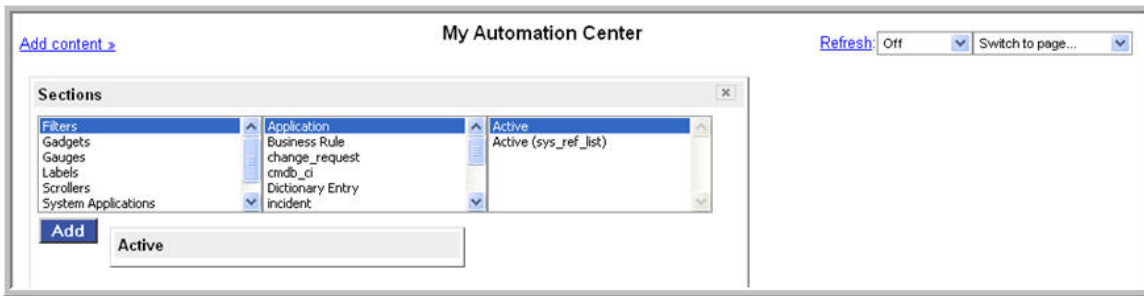
### Removing Widgets

For each widget, click the small X in its upper right corner to remove it from the home page.

### Adding Widgets

Now we are going to add the widgets to the home page.

**Step 1** Click **Add content**. The following menu displays:



(Click [here](#) for a description of each option on the menu.)

**Step 2** To add the Overview gauge, select **Gadgets > System Information > Overview** and click **Add**. The Overview panel appears on the home page.

**Step 3** To add the World Clocks, select **World Clocks > World Clocks** and click **Add**.

**Step 4** To add Task Activity Status, click **Gauges > Activity > Task Instances by Status** and click **Add**.

**Step 5** To add Active Task Instances By Type, click **Gauges > Activity > Active Task Instances By Type** and click **Add**.

**Step 6** To add Agent Connection Status, click **Gauges > Agent > Agent Connection Status** and click **Add**.

**Step 7** We are finished adding windows, so click the **X** in the upper corner of the **Add Content > Sections** menu.

**Step 8** Now we will move the windows around to match the sample. For each window, click the grey title bar and drag it to the appropriate location, as indicated by the sample.

## Setting the Refresh Rate

The default refresh rate is no refresh. Set it to 5 minutes by clicking the drop-down menu (upper right) and select **5 minutes**.


For additional information, see:

- [Home Page, Widgets, Dashboard \(3 minute movie\)](#)
- [Home Page, Dashboard, and Gauges](#)



## Tutorial - Creating a Gauge

In this exercise, we will use an existing report to create a new gauge for our home page.

<b>Step 1</b>	Select <b>Automation Center &gt; Reports</b> .
<b>Step 2</b>	Scroll down to the Activity section, locate <b>All Task Instances by Status</b> , and click it to open it.
<b>Step 3</b>	Click <b>Make Gauge</b> .
<b>Step 4</b>	Click the home  icon (upper right) to return to the home page.
<b>Step 5</b>	Click <b>Add content</b> .
<b>Step 6</b>	Select <b>Gauges &gt; Activity &gt; All Task Instances by Status</b> and click <b>Add</b> . The gauge is added to the home page.

**My Automation Center** Refresh: Off | Switch to page...

**All Task Instances by Status** 20 Task Instances

Task Instances Grouped By Status

- Status
- Status: FINISHED (12)
- Status: SUCCESS (8)

**Task Activity Status**

Running	0
Problem	0
Held/Action Required	0
Pending	0
Success/Finished	20
Skipped	0

**Agent Connection Status**

Active (4) (1.00%)

**Connector Status**

Active (2) (1.00%)

**Overview**

License	[Agents: 4/5] [Triggers: Unlimited] [Tasks: Unlimited] [Days: 1/45]
Node	server.opswissoftware.com:8080-opwise [Mode: Active]
Node Time	2013-02-05 13:33:03 -0800
Release	5.1.1.5
Build	build.6
Build date	02-04-2013_1250
DBMS Type	mysql
DB uri	jdbc:mysql://localhost/
DB Name	opwise
DB Connections	In Use: 0, Total: 4
Active sessions	2
Memory max	455MB
Memory allocated	331MB
Memory used	207MB
Memory free	37% of allocated memory, 54% of max memory

**Cluster Node Status**

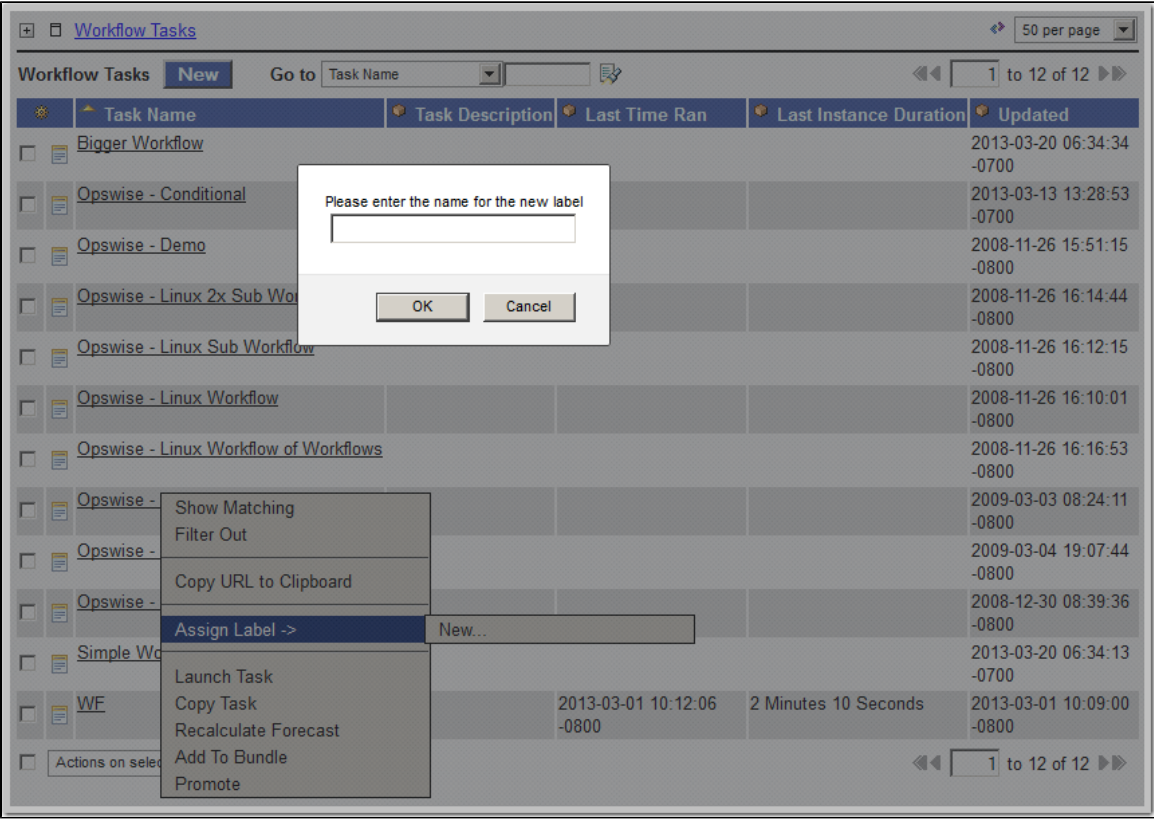
Active (1) (1.00%)

For additional information, see:

- [Home Page, Widgets, Dashboard \(3 minute movie\)](#)
- [Home Page, Dashboard, and Gauges](#)

## Tutorial - Adding Options to the Navigation Pane

In this exercise, we will add a new section to the navigation pane for a hypothetical business unit called Collections. We will then add records to the new section. This process does not create any new records; it merely creates links to records in the navigation pane.

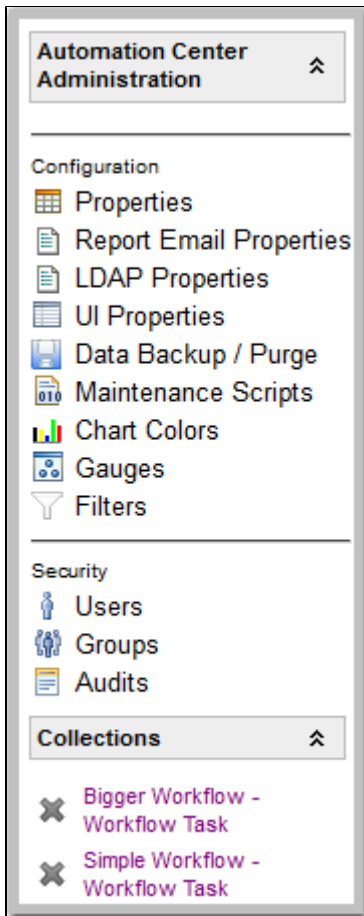
<b>Step 1</b>	Display the Workflow Tasks list and locate the <b>Simple Workflow</b> workflow created in the <a href="#">Creating a Simple Workflow</a> tutorial.
<b>Step 2</b>	<p>Right-click the <b>Simple Workflow</b> Task Name and, on the <b>Action menu</b>, hover your cursor over <b>Assign Label -&gt;</b> and click <b>New...</b> A pop-up dialog displays requesting the name of a new label.</p> 
<b>Step 3</b>	Enter <b>Collections</b> and click <b>OK</b> . A <b>Collections</b> label displays at the bottom of the navigation pane.
<b>Step 4</b>	Locate the <b>Bigger Workflow</b> workflow created in the <a href="#">Running a Workflow with a Conditional Path</a> tutorial.

**Step 5** Right-click the **Bigger Workflow** Task Name and, on the **Action** menu, hover your cursor over **Assign Label ->**, and click **Collections**

The screenshot shows the 'Workflow Tasks' management interface. At the top, there is a 'New' button and a search field labeled 'Go to Task Name'. Below this is a table with columns: Task Name, Task Description, Last Time Ran, Last Instance Duration, and Updated. The table contains several workflow tasks, with 'Bigger Workflow' at the top. A context menu is open over the 'Bigger Workflow' task, listing actions such as 'Show Matching', 'Filter Out', 'Copy URL to Clipboard', 'Assign Label ->', 'Launch Task', 'Copy Task', 'Recalculate Forecast', 'Add To Bundle', and 'Promote'. The 'Assign Label ->' option is highlighted, and a sub-menu is visible with 'Collections' selected. Other tasks listed include 'Opwise - Linux workflow of workflows', 'Opwise - SQL Result Processing', 'Opwise - Stored Procedure Workflow', 'Opwise - Workflow of Sleep Tasks', 'Simple Workflow', and 'WF'. The interface also includes pagination controls showing '1 to 12 of 12' items and a '50 per page' dropdown.

Task Name	Task Description	Last Time Ran	Last Instance Duration	Updated
Bigger Workflow				2013-03-20 06:34:34 -0700
Opwise				2013-03-13 13:28:53 -0700
Opwise				2008-11-26 15:51:15 -0800
Opwise				2008-11-26 16:14:44 -0800
Opwise				2008-11-26 16:12:15 -0800
Opwise				2008-11-26 16:10:01 -0800
Opwise - Linux workflow of workflows				2008-11-26 16:16:53 -0800
Opwise - SQL Result Processing				2009-03-03 08:24:11 -0800
Opwise - Stored Procedure Workflow				2009-03-04 19:07:44 -0800
Opwise - Workflow of Sleep Tasks				2008-12-30 08:39:36 -0800
Simple Workflow				2013-03-20 06:34:13 -0700
WF		2013-03-01 10:12:06 -0800	2 Minutes 10 Seconds	2013-03-01 10:09:00 -0800

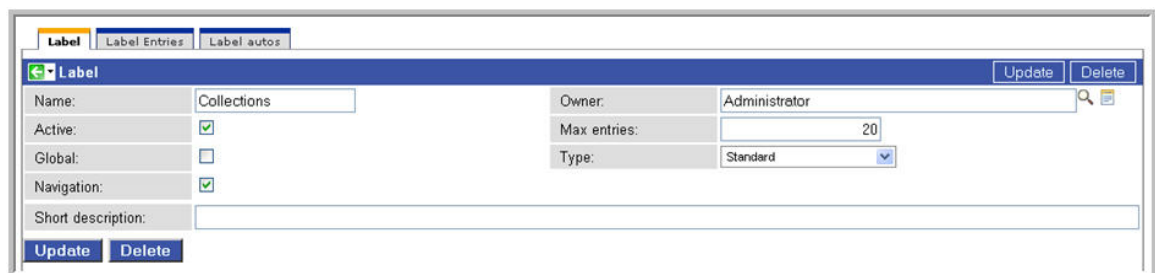
**Step 6** In the navigation pane, click the down arrows next to the **Collections** label to see the two workflows added to that Label.



You can view either Workflow record under **Collections** by clicking the record name.

**Step 7** Now we will remove both of the records, along with the new menu option.

1. Click the **X** next to each record under the **Collections** Label. The records disappear from the list.
2. Click the **Collections** Label. The Label record appears.



3. Click the **Delete** button. The Label disappears from the navigation pane.

## Tutorial - Creating Business Services

Business Services are used to group records into business functions.

In this exercise, we will create two hypothetical Business Services: Tech Support and Operations.

<b>Step 1</b>	On the navigation pane, select <b>Automation Center -&gt; Business Services</b> . The Business Services list screen displays.
<b>Step 2</b>	Click <b>New</b> . The Business Service definition screen displays.
<b>Step 3</b>	In the Name field, enter <b>Tech Support</b> .
<b>Step 4</b>	In the Description field, enter <b>This is the Tech Support business service</b> .
<b>Step 5</b>	Click <b>Submit</b> .
<b>Step 6</b>	Repeat steps 2 to 5 for a Business Service called <b>Operations</b> .

Business Services		
Name	Description	Version
Operations	This is the Operation business service.	1
Tech Support	This is the Tech Support business service.	1



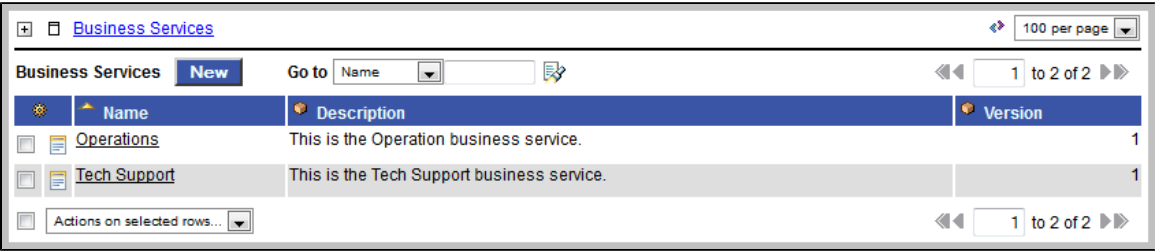
For additional information, see:

- [Business Services](#)

## Tutorial - Assigning Records to Business Services

In this exercise, we will assign the Sleep and Simple Workflow tasks created in the [Creating a Simple Workflow](#) tutorial to the Operations Business Service, and the SQL task and Bigger Workflow tasks created in the [Running a Workflow with a Conditional Path](#) tutorial) to the Tech Support Business Service.

See the [Creating Business Services](#) tutorial to see how these Business Services were created.

<b>Step 1</b>	Open the task called Sleep1, which you created for an earlier exercise.
<b>Step 2</b>	Click the lock  icon next to <b>Member of Business Services</b> .
<b>Step 3</b>	Click the search icon  and click <b>Operations</b> , which is the group you just added.
	 <p>The screenshot shows a table titled 'Business Services' with columns for Name, Description, and Version. The 'Operations' service is highlighted, and its description is 'This is the Operation business service.' The 'Tech Support' service is also listed with the description 'This is the Tech Support business service.' The table has a 'New' button and a 'Go to' dropdown menu. The page number is 1 of 2.</p>
<b>Step 4</b>	Click <b>Update</b> to save your changes.
<b>Step 5</b>	Repeat steps 1 to 4 for the other Sleep tasks you created earlier, Sleep2 to Sleep6, along with the Workflow we put them in, <a href="#">Simple Workflow</a> .
<b>Step 6</b>	Open the SQL Create Table task you created for an earlier exercise.
<b>Step 7</b>	Click the Member of Business Services lock, assign the task to the Tech Support group, and click <b>Update</b> .
<b>Step 8</b>	Repeat the previous step for the other three SQL tasks you created earlier, along with the workflow we put them in, <a href="#">Bigger Workflow</a> .

For additional information, see:

- [Business Services](#)

## Tutorial - Taking Advantage of Business Services



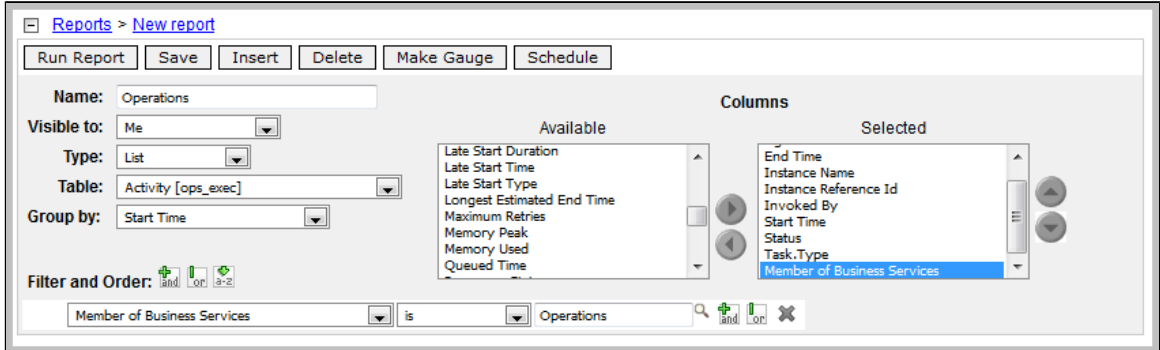
The advantage to Business Services is demonstrated in the following additional tutorials:

- [Create a Report Based on Business Services.](#)
- [Viewing Activity by Business Service.](#)
- [Creating Users and Assigning Permissions based on Business Services.](#)



## Tutorial - Creating a Report Based on Business Services

In this exercise, we will create two Activity reports so that users from our hypothetical Operations and Tech Support departments (see the [Creating Business Services](#) tutorials) can view activity related to their organizations.

<b>Step 1</b>	Select <b>Automation Center &gt; Reports</b> and click <b>New</b> .
<b>Step 2</b>	In the Name field, type <b>Operations</b> .
<b>Step 3</b>	For now, leave the <b>Visible to</b> field with the default value.
<b>Step 4</b>	For type, select <b>List</b> .
<b>Step 5</b>	For Table, leave the default <b>Activity [ops_exec]</b> , which is the table that contains all Task Instances.
<b>Step 6</b>	In Group by, select <b>Start Time</b> .
<b>Step 7</b>	In Filter and Order, click the <b>+and</b> icon. The choose field options appear.
<b>Step 8</b>	Select: <ul style="list-style-type: none"> <li>• Member of Business Services</li> <li>• is</li> <li>• Operations (Hint : Type an <b>O</b> into the field and "Operations" pops up.)</li> </ul>
<b>Step 9</b>	<p>Modify the Available and Selected Columns to display the following fields. Hint: Use  and  to reorder the columns.</p> <ul style="list-style-type: none"> <li>• Instance Name</li> <li>• Type</li> <li>• Status</li> <li>• Start Time</li> <li>• End Time</li> <li>• Invoked by</li> <li>• Agent</li> <li>• Member of Business Services</li> </ul> 
<b>Step 10</b>	Click <b>Save</b> . Note that the new report is saved into a separate "Activity" section under "My Saved reports." This is because it is visible only to "Me."
<b>Step 11</b>	Repeat the above sets for a report called <b>Tech Support</b> , filtered by "Member of Business Services is Tech Support." Hint: Keep the Operations report displayed, change the name and filter specifications, then click <b>Insert</b> .

For additional information, see:

- [Opswise Controller - Reports](#)

## Tutorial - Viewing Activity by Business Service

- [Introduction](#)
- [Create Some Activity](#)
- [Create Activity Screens Based on Business Services](#)

### Introduction

In this exercise, we will create some activity by launching two workflow, open two new browser tabs and drag the Activity function into each one of them, and apply each of the Activity reports created in the [Creating a Report Based on Business Services](#) tutorial.

### Create Some Activity

Launching the following two workflows:

- Simple Workflow (See the [Creating a Simple Workflow](#) tutorial.)
- Bigger Workflow (See the [Running a Workflow with a Conditional Path](#) tutorial.)

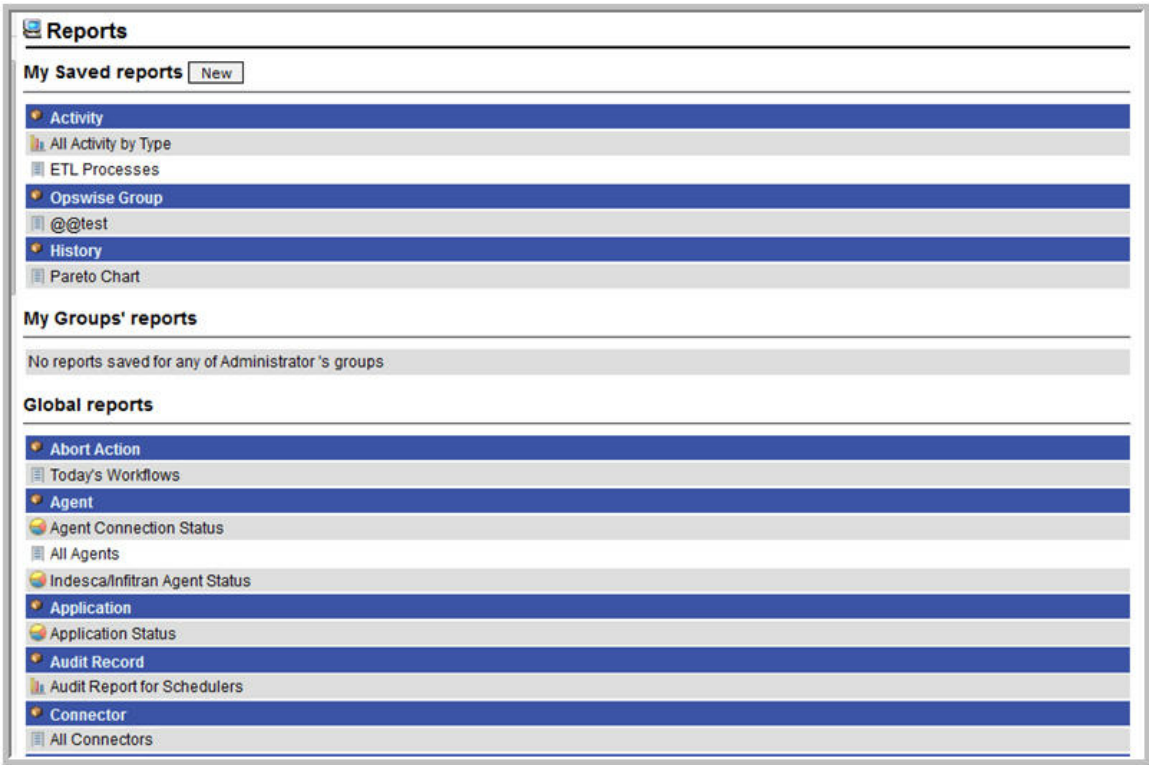
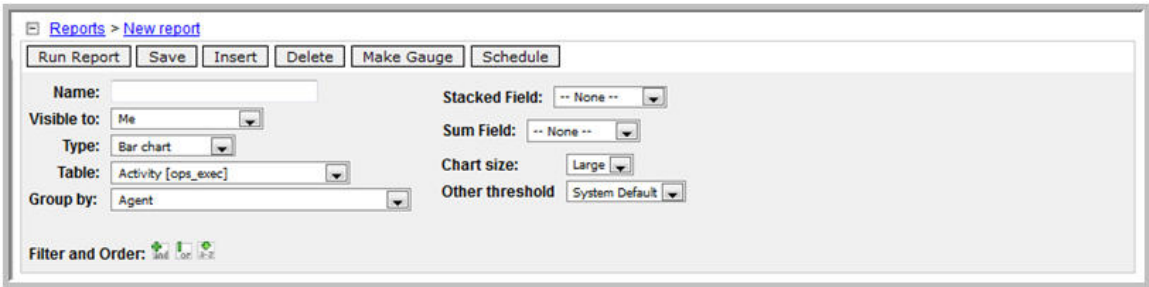
### Create Activity Screens Based on Business Services



<b>Step 1</b>	Open two new browser tabs.
<b>Step 2</b>	Return to the first tab. Click and drag <b>Task Instances &gt; Activity</b> to each of the new tabs.
<b>Step 3</b>	In the first tab, select <b>Operations</b> in the Activity Report drop-down. You will now see only activity for tasks belonging to the Operations Business Service.
<b>Step 4</b>	In the second tab, do the same for the Tech Support report.

## Tutorial - Creating a Report

In this exercise, we will create a new report.

Follow the steps below to create a sample Activity report that shows all tasks instances, sorted by resource.

<p><b>Step 1</b></p>	<p>From the navigation pane, select <b>Automation Center -&gt; Reports</b>. The Reports list screen displays.</p>  <p>This screen provides a list of all reports that your userid has access to, along with a <b>New</b> button that allows you to create a new report.</p>
<p><b>Step 2</b></p>	<p>Under My Saved reports, click <b>New</b>. A blank reports screen displays.</p> 
<p><b>Step 3</b></p>	<p>In the Name field, enter <b>Tasks Sorted by Resource</b>.</p>
<p><b>Step 4</b></p>	<p>In the <b>Visible to</b> field, select <b>Me</b>. This identifies who will have access to the report as follows:</p> <ul style="list-style-type: none"> <li>• Me — Only you can view/run the report.</li> <li>• Everyone — All users can view/run the report.</li> <li>• All other options — The remainder of the drop-down list consists of the groups that have been entered by selecting <b>Security &gt; Groups</b> from the Navigation pane.</li> </ul>
<p><b>Step 5</b></p>	<p>In the <b>Type</b> field, select <b>List</b>.</p>

<b>Step 6</b>	In the <b>Table</b> field, select the <a href="#">Opswise Controller report table</a> you want to use. For this tutorial, select <b>Activity (ops_exec)</b> , which is the table used for all Activity reports and contains information about all task activity.
<b>Step 7</b>	In <b>Group by</b> , select <b>Agent</b> . This field specifies that the report should be displayed in sections by agent.
<b>Step 8</b>	The Columns section displays a list of available fields in the Available section, and the selected fields in the Selected section. A number of default fields appear in the Selected section. For this tutorial, we will select Instance Name, Agent, Status, Type, and Duration. Select and deselect fields as follows: <ul style="list-style-type: none"> <li>• On the Selected list, double click fields to remove them from the report.</li> <li>• On the Available list, double click fields to add them to the report.</li> </ul>
<b>Step 9</b>	To run the report, click <b>Run Report</b> . Opswise Controller creates a list of Task Types, as specified in the report.
<b>Step 10</b>	To expand all sections of the report, click the plus  sign.
<b>Step 11</b>	Now you will filter the report to select only tasks that ran today: <ol style="list-style-type: none"> <li>1. Under Filter and Order, click on the box with a plus sign  and the word "and".</li> <li>2. You are prompted to select filtering criteria.</li> <li>3. For <b>choose field</b>, select Created.</li> <li>4. Select 'on'.</li> <li>5. Keep the default 'Today'.</li> <li>6. Click <b>Run Report</b> again. The report is re-run with the filter applied, removing those task instances that ran before today.</li> </ol>
<b>Step 12</b>	To save the report, click <b>Save</b> .
<b>Step 13</b>	The report is now listed in your section (My Saved Reports) of the Reports list. <ul style="list-style-type: none"> <li>• To view the report, click <b>Reports</b> from the Navigation Pane. It has also been added to the list of Activity reports on the Activity screen.</li> <li>• To print the report, save it first and then print it (see <a href="#">Viewing, Running, Printing an Existing Report</a>).</li> </ul>

---

For additional information, see:

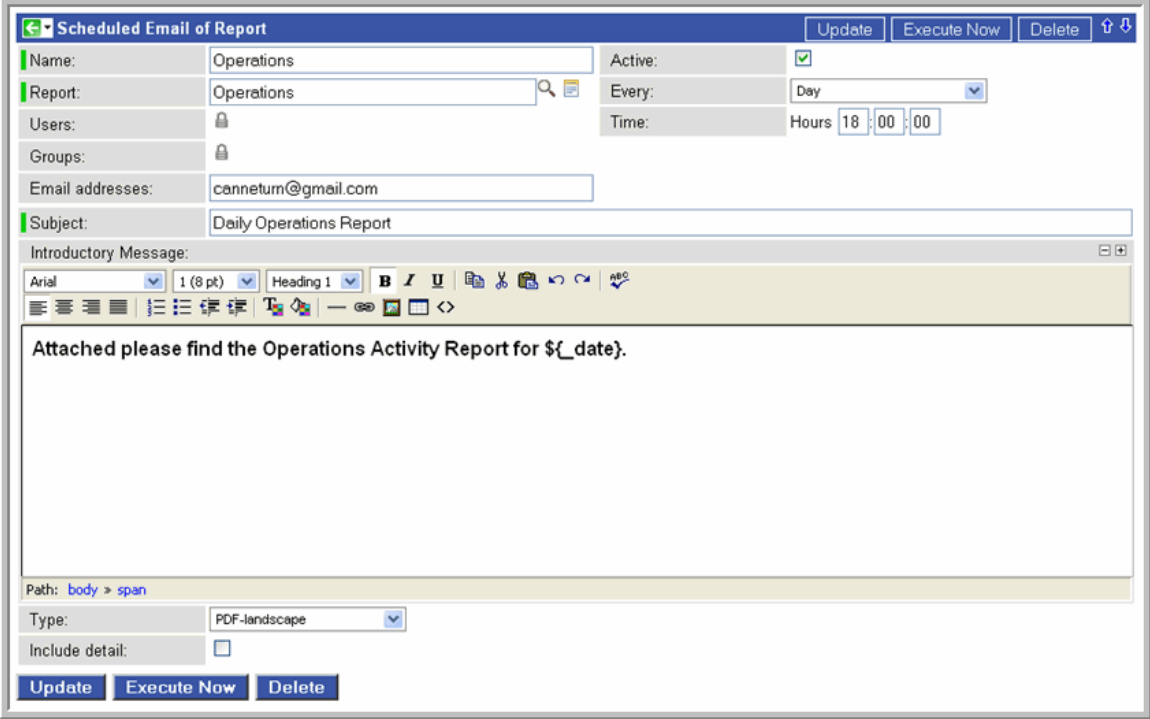
- [Reporting Overview \(5 minute movie\)](#)
- [Opswise Controller - Reports](#)

## Tutorial - Scheduling and Distributing Reports



### Note

Before you can distribute reports, you must configure the email server used for this purpose.

<b>Step 1</b>	Open the Operations report that you created in the <a href="#">Creating a Report Based on Business Services</a> tutorial.
<b>Step 2</b>	Click <b>Schedule</b> .
<b>Step 3</b>	<p>Complete the fields as follows:</p> <ul style="list-style-type: none"> <li>• Name = Operations</li> <li>• Report = Operations</li> <li>• Every = Day</li> <li>• Time = (specify a few minutes from now)</li> <li>• Email addresses = (your email address)</li> <li>• Subject = Daily Operations Report</li> <li>• Message = Attached please find the Operations Activity Report for \${_date}.</li> </ul>
	
<b>Step 4</b>	Click <b>Submit</b> .
<b>Step 5</b>	At the time you indicated, check your inbox for a message containing the report in PDF format.

For additional information, see:

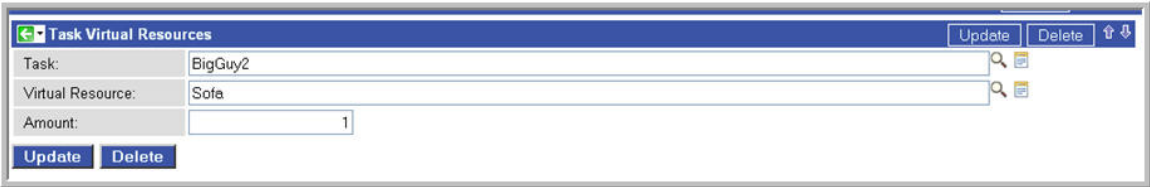
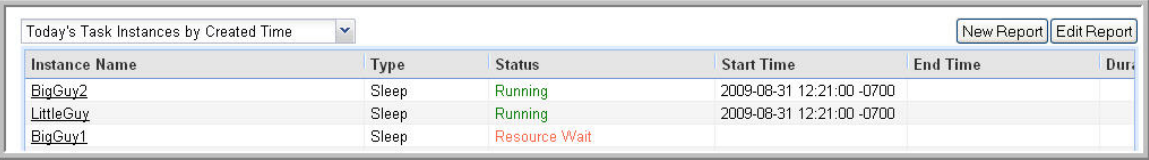
- [Opwise Controller - Reports](#)

## Tutorial - Setting Up a Virtual Resource

In this exercise, we will set up an imaginary resource and three imaginary tasks.

In this scenario, two of our tasks are resource-intensive, and they run on the same machine. Therefore, if one is already running when the other is launched, we want the second task to wait until the first is finished before running. However, our third task is not so resource-intensive, so we will allow this one to run at the same time as either of the other two.

To simplify the exercise, we will use Sleep tasks.

<b>Step 1</b>	Select <b>Automation Center -&gt; Automation Center Resources &gt; Virtual Resources</b> . The Virtual Resources list screen displays.																								
<b>Step 2</b>	Click <b>New</b> . The Virtual Resource definition screen displays.																								
<b>Step 3</b>	Name the Virtual Resource <b>Sofa</b> , change the Resource Limit to 5, and click <b>Submit</b> .																								
<b>Step 4</b>	<p>Set up the Sleep tasks as follows:</p> <ol style="list-style-type: none"> <li>Add the following Sleep task: <ul style="list-style-type: none"> <li>Task Name=BigGuy1</li> <li>Sleep Time=60</li> </ul> </li> <li>Save the task and click the <b>Task Virtual Resources</b> tab.</li> <li>Click <b>Edit</b>.</li> <li>Select <b>Sofa</b>.</li> <li>Sofa is added to the list with a default resource amount of 1. Click the Resource Type name for Sofa to open the record.</li> </ol>  <ol style="list-style-type: none"> <li>Change the Amount to 4, which means that this task will consume most of the resource.</li> <li>Repeat the above steps for another Sleep task called BigGuy2 with a sleep time of 30 seconds. Give BigGuy2 a resource amount of 4 as well.</li> <li>Add a third Sleep task called LittleGuy with a sleep time of 20, assign it to the Sofa resource and leave the resource amount at 1.</li> </ol>																								
<b>Step 5</b>	Display the Sofa Virtual Resource and click the <b>Task Virtual Resources</b> tab. All tasks assigned to this virtual resource are listed, along with the resource amount that each one uses.																								
<b>Step 6</b>	We want all three of our tasks to launch at once, so use a simple <a href="#">time trigger</a> to launch all three in a minute or two. (Don't forget to enable the trigger.)																								
<b>Step 7</b>	When the three tasks are launched, only one of the Big Guys can actually run, along with Little Guy. When the first Big Guy finishes, there is room for the second to run. Display the Activity screen and note that the second Big Guy task is waiting in Resource Wait status.																								
	 <table border="1"> <thead> <tr> <th>Instance Name</th> <th>Type</th> <th>Status</th> <th>Start Time</th> <th>End Time</th> <th>Dur.</th> </tr> </thead> <tbody> <tr> <td>BigGuy2</td> <td>Sleep</td> <td>Running</td> <td>2009-08-31 12:21:00 -0700</td> <td></td> <td></td> </tr> <tr> <td>LittleGuy</td> <td>Sleep</td> <td>Running</td> <td>2009-08-31 12:21:00 -0700</td> <td></td> <td></td> </tr> <tr> <td>BigGuy1</td> <td>Sleep</td> <td>Resource Wait</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Instance Name	Type	Status	Start Time	End Time	Dur.	BigGuy2	Sleep	Running	2009-08-31 12:21:00 -0700			LittleGuy	Sleep	Running	2009-08-31 12:21:00 -0700			BigGuy1	Sleep	Resource Wait			
Instance Name	Type	Status	Start Time	End Time	Dur.																				
BigGuy2	Sleep	Running	2009-08-31 12:21:00 -0700																						
LittleGuy	Sleep	Running	2009-08-31 12:21:00 -0700																						
BigGuy1	Sleep	Resource Wait																							
<b>Step 8</b>	On the Virtual Resource Sofa, click the <b>Currently In Use By</b> tab and observe which tasks are running on this virtual resource. Note that this display does not automatically refresh.																								

For additional information, see:

[Virtual Resources](#)



## Tutorial - Accessing Task Instance Details

**Step 1** Navigate to the Activity screen.



- Step 2** Click the **Instance Name** of any task instance to display its Task Instance screen. The screen contains many fields, which are not displayed on the Task Definition screen for this task, that provide information about this run of the task (this task instance) and all runs of the task, including the first time it was run; the last time it was run; the number of times it has run; and the least, average, and most amount of time it has ever taken to run.

The screenshot shows the 'Linux/Unix Task Instance' screen with the following details:

- Instance Name:** Opwise - Linux Failure Exit Code 10
- Task:** Opwise - Linux Failure Exit Code 10
- Reference Id:** 10
- Member of Business Services:**
- Agent:** server.opswissoftware.com - AGN
- Agent Variable:**
- Hold Reason:**
- Task Description:**
- Status:** FINISHED
- Exit Code:** 10
- Status Description:** State was forced from FAILED to FINISHED
- Queued Time:** 2013-09-16 09:34:06 -0700
- Start Time:** 2013-09-16 09:34:21 -0700
- End Time:** 2013-09-16 09:34:22 -0700
- Process ID:** 1380
- CPU Time:** 199
- Duration:** 1 Seconds
- Command or Script:** Command
- Command:** exit
- Parameters:** 10
- Runtime Directory:**
- Exit Code Processing:** Success Exitcode Range
- Exit Codes:** 0
- Automatic Output Retrieval:** Standard Output
- Start Line:**
- Number of Lines:**
- Scan Text:**
- Task Priority:** MEDIUM
- Maximum Retries:** 0
- Retry Interval (Seconds):** 60
- Current Retry Count:** 0
- Retry Indefinitely:**
- User Estimated End Time:** 2013-09-17 06:34:21 -0700
- Average Estimated End Time:** 2013-09-16 09:34:21 -0700
- Virtual Resource Priority:** 10
- Shortest Estimated End Time:**
- Longest Estimated End Time:**
- Hold Resources on Failure:**

- Step 3** To view all details stored in the **Opwise Activity table** (ops\_exec) for this task instance, click the **Show Details** button. A new browser tab opens, displaying the task instance details.

- Left column shows each field name from the Activity table.

- Right column shows the current value for each field for this task instance.

agent	2792d1e5d861e5e400d10cb35ea55b40
agent_cluster	da8318b0c0a8016501f15de66f0f40d1
agent_cluster_var_check	false
agent_id	2792d1e5d861e5e400d10cb35ea55b40
agent_var_check	false
attempt_count	1
avg_estimated_end	2013-09-16 16:34:21
calendar	77171434c0a801c9016d5b2b5d17ddee
command	exit
command_or_script	Command
cpu_time	199
credentials_var_check	false
desktop_interact	false
duration	1000
duration_seconds	1
early_finish	false
ef_enabled	false
ef_type	TIME
end_time	2013-09-16 16:34:22
exclusive_state	Initial
execution_user	ops.admin
exit_code	10
exit_code_processing	Success Exitcode Range
exit_codes	0
forced	true
hold_resources	false
invoked_by	Workflow: Opwise - Linux Workflow
io_other	0
io_reads	0
io_writes	0
late_finish	false
late_start	false
launch_time	2013-09-16 16:33:37
lf_enabled	false

lf_type	TIME
ls_enabled	false
ls_type	TIME
memory_peak	0
memory_used	0
name	Opwise - Linux Failure Exit Code 10
output_return_type	STDOUT
output_type	STDOUT
parameters	10
priority	MEDIUM
process_id	1380
queued_time	2013-09-16 16:34:06
res_priority	10
res_state	Initial
resources_consumed	false
retry_counter	0
retry_indefinitely	false
retry_interval	60
retry_maximum	0
run_as_sudo	false
run_called	true
run_criteria_rt	false
run_criteria_tt	false
security_name	Opwise - Linux Failure Exit Code 10
start_held	false
start_time	2013-09-16 16:34:21
state_changed_time	2013-09-16 16:34:22
status_code	FINISHED
status_description	State was forced from FAILED to FINISHED
sys_class_name	ops_exec_unix
sys_created_by	ops.admin
sys_created_on	2013-09-16 16:33:38
sys_id	279ec323d861e5e4007347c6c5aa734e
sys_mod_count	9

sys_updated_by	glide.maint
sys_updated_on	2013-09-16 16:34:22
task_id	64e831fac0a802ba00a36c96a0187359
task_ref_count	10
type	Linux/Unix
user_estimated_end	2013-09-17 13:34:21
vertex_id	21
wait_for_exclusive	false
wait_for_resources	false
workflow_definition_id	699e9650c0a800030059a24b880a768a
workflow_id	279ec11bd861e5e40127c2e5f8308662
workflow_start_time	2013-09-16 16:33:52

## Tutorial - Aborting a Process Launched by a Task

You can use the Abort Actions tab to instruct Opwise Controller to abort a process under certain conditions. For example, you may want to abort a task if it is running too long.

For demonstration purposes, we will set a Sleep task to run for 60 seconds and specify an Abort action once the task runs longer than 45 seconds.

**Step 1** Create a new Sleep task that will run for 60 seconds:

1. Enable Late Finish.
2. Specify a Late Finish Type of Duration.
3. Specify a Late Finish Duration of 45 seconds. This means that the task will be considered late finishing if it takes longer than 45 seconds.

The screenshot shows the configuration window for a 'Sleep Task'. The task name is 'Sleep 60' and the version is '1'. The sleep type is set to 'Duration' with a sleep duration of 00:01:00. The task description is 'Sleep for 60 seconds.' The 'Late Finish' checkbox is checked, and the 'Late Finish Type' is set to 'Duration' with a 'Late Finish Duration' of 00:00:45. The 'Early Finish' checkbox is unchecked. The 'Virtual Resource Priority' is set to 10. The 'First Time Ran' and 'Last Time Ran' are both 2013-09-20 08:29:31 -0700. The 'Number of Instances' is 1. The 'Member of Business Services' is locked, and 'Hold on Start' is unchecked. The 'Hold Resources on Failure' checkbox is also unchecked.

**Step 2** Save the task and click the **Actions** tab.

**Step 3** Click **New** and select **Abort Action** from the Action wizard.

**Step 4** Enable On Late Finish and click **Submit**.

**Step 5** Launch the Sleep task.

**Step 6** Navigate to the Activity screen. After 45 seconds, the task goes to Finished status.

**Step 7** Click on the task instance to display details. Note the status description indicates:

State was forced from RUNNING to FINISHED

For additional information, see:

- [Setting Up Abort Actions](#)



## Tutorial - Creating Users and Assigning Permissions

- Introduction
- Create New Users
- Assign Permissions Via User Roles
- Assign Specific Permissions to a User
- Assign Permissions to Groups of Users

### Introduction

In this exercise, we will create some users related to the Operations and Tech Support departments created in the [Creating Business Services](#) tutorial. We also will assign access and management rights via user roles and Opwise Controller permissions:

- **Permissions** control who can add, change, delete and control Controller tasks, task instances, triggers, Agents, calendars, and credentials.
- **Roles** are pre-defined groups of permissions that control access to users, reports (filters), gauges, bundles, and promotions.

### Create New Users


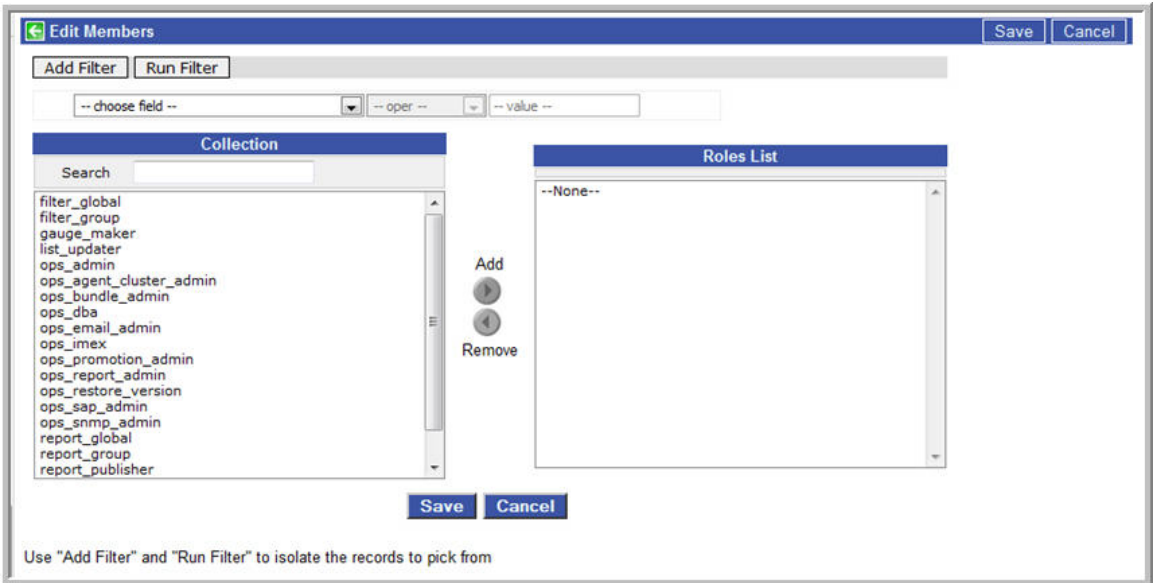
In this exercise, we will create three new users.

<b>Step 1</b>	From the navigation pane, select <b>Security &gt; Users</b> . You will see the default user, ops.admin.
<b>Step 2</b>	Click <b>New</b> .
<b>Step 3</b>	In the Name field, type OpMan (short for Operations Manager).
<b>Step 4</b>	Fill in the remaining fields as follows: <ul style="list-style-type: none"> <li>• First name=Ben</li> <li>• Lastname=Bernanke</li> <li>• Password=abc</li> </ul>
<b>Step 5</b>	Click <b>Submit</b> .
<b>Step 6</b>	Add two more users: <ul style="list-style-type: none"> <li>• TSMAN, name=Tim Geithner, password=abc</li> <li>• OpClerk, name=Jimmy Smith, password=abc</li> </ul>
<b>Step 7</b>	To test one of the new users: <ol style="list-style-type: none"> <li>1. Click the <b>Logout</b> button in the upper right corner of the display.</li> <li>2. Log back in using the OpMan/abc ID and password.</li> <li>3. Click on several options in the navigation pane. Since we have not assigned any permissions to OpMan, this user is prohibited from the following: <ul style="list-style-type: none"> <li>• Viewing tasks, tasks instances, or triggers</li> <li>• Changing or deleting the dashboard or home page items, email or database connections, calendars or custom days, and Business Services.</li> <li>• The Security and administrative options do not display on the navigation pane.</li> </ul> </li> <li>4. When you are finished, click <b>Logout</b> and log back in using ops.admin.</li> </ol>

### Assign Permissions Via User Roles

In this exercise, we will assign global permissions to a user by using roles.

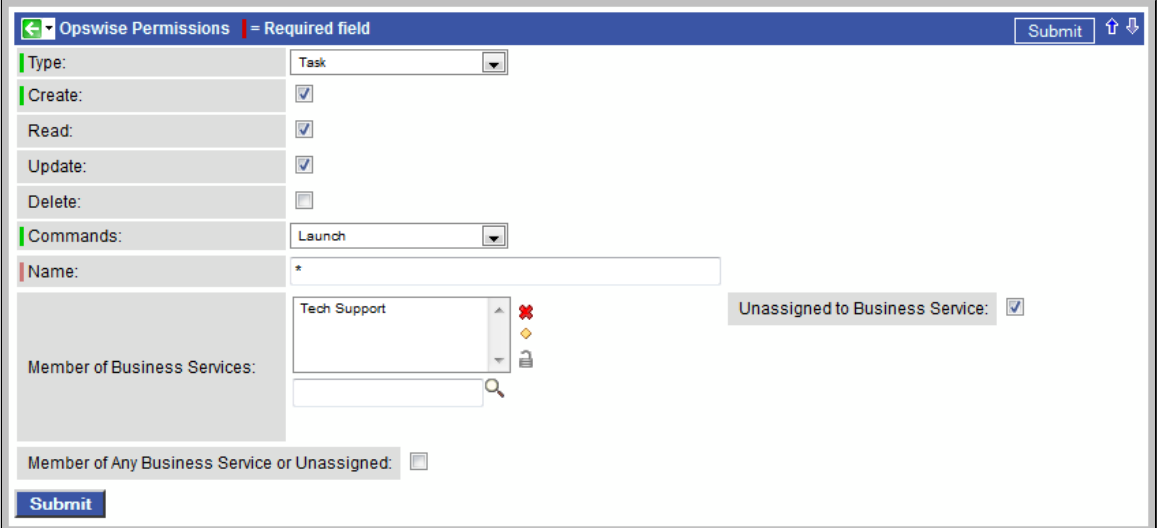
<b>Step 1</b>	Open the OpMan user and click the <b>User Roles</b> tab.
---------------	--

<b>Step 2</b>	Click <b>Edit</b> .
<b>Step 3</b>	<p>This user is a manager so we will give him the <b>admin roles</b>. Ctrl-Click <b>ops_admin</b> and <b>ops_report_admin</b>, and click the right-arrow .</p> 
<b>Step 4</b>	Click <b>Save</b> .
<b>Step 5</b>	Logout and log back in as OpMan. Note that this user has full permissions on everything because it was given the admin roles.

## Assign Specific Permissions to a User

In this exercise, we are going to give TSMAN limited permission, including rights to add, delete, change records that belong to the Tech Support group.

<b>Step 1</b>	Open the User, TSMAN.
<b>Step 2</b>	Click the <b>Opswise Permissions</b> tab and click <b>New</b> .
<b>Step 3</b>	First we will give permission to view, add, update, and manually launch tasks.
<b>Step 4</b>	In Type, select <b>Task</b> .
<b>Step 5</b>	Select Create, Read, and Update.
<b>Step 6</b>	In Business Services, select <b>Tech Support</b> .
<b>Step 7</b>	Deselect Unassigned to Business Service (see <a href="#">Permissions Field Descriptions</a> for more information).

<p><b>Step 8</b></p>	<p>In Commands, select <b>Launch</b>.</p> 
<p><b>Step 9</b></p>	<p>Click <b>Submit</b>.</p>
<p><b>Step 10</b></p>	<p>Test the permissions by logging out again and logging back in as TSMAN/abc. TSMAN will be able to see the tasks assigned to the Tech Support group, and launch those tasks. However, TSMAN cannot see them on the Activity screen because TSMAN was not given permissions on Task Instances.</p>

## Assign Permissions to Groups of Users

In this exercise we will assign our last user to a group, then assign permissions to the group instead of to the user.

<p><b>Step 1</b></p>	<p>Open the user record OpClerk.</p>
<p><b>Step 2</b></p>	<p>Click the <b>Group Members</b> tab.</p>
<p><b>Step 3</b></p>	<p>Click <b>New</b> to add a new group.</p>
<p><b>Step 4</b></p>	<p>Name the group Operations Clerks.</p>
<p><b>Step 5</b></p>	<p>Access the <b>Action</b> menu and click <b>Save</b>.</p>
<p><b>Step 6</b></p>	<p>While still displaying the Operations Clerks group, click the <b>Opwise Permissions</b> tab and click <b>New</b>.</p>
<p><b>Step 7</b></p>	<p>Assign the following permissions:</p> <ul style="list-style-type: none"> <li>• Type = Task Instance</li> <li>• Read, Update, Delete = Enabled</li> <li>• Business Services = Operations</li> <li>• Unassigned to Business Service= Disabled</li> <li>• Commands = All</li> </ul> <p>These permissions give the Operations Clerks full permissions on all activity (task instances) related to the Operations Business Service. In future, any users you assign to this user group will inherit these permissions.</p>
<p><b>Step 8</b></p>	<p>Click <b>Submit</b>.</p>
<p><b>Step 9</b></p>	<p>Log out and log back in as OpClerk and check the permissions.</p>

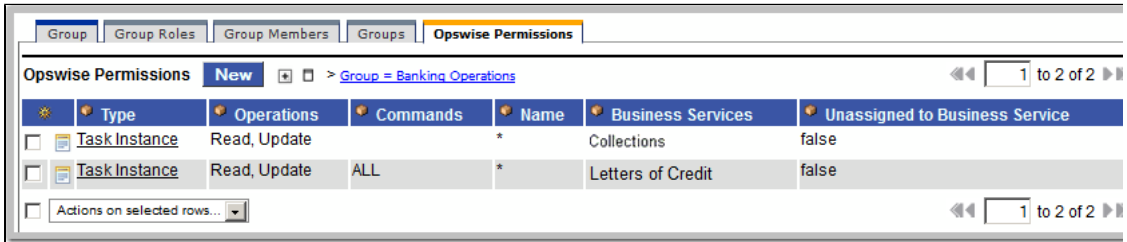
For additional information, see:

- [Users and Groups](#)



## Tutorial - Creating Security Groups and Assigning Permission

In this exercise, we will create users and user groups, then assign permissions to the groups instead of directly to the users.

<b>Step 1</b>	<p>Create the following three users:</p> <ul style="list-style-type: none"> <li>• AmyH/Amy Hempel/abc</li> <li>• TobyW/Tobias Wolff/abc</li> <li>• RayC/Ray Carver/abc</li> </ul>																		
<b>Step 2</b>	<p>Select <b>Users &gt; Groups</b> and add a group called <b>Data Center</b> and save it.</p> <ol style="list-style-type: none"> <li>1. Click the <b>Opwise Permissions</b> tab and add the following permission: <ul style="list-style-type: none"> <li>• Type=Task Instance</li> <li>• Read</li> <li>• All Business Services</li> </ul> </li> <li>2. Click the <b>Group Members</b> tab and click <b>Edit</b>.</li> <li>3. Add Ray Carver to the group.</li> </ol>																		
<b>Step 3</b>	<p>Add another group called <b>Banking Operations</b> and save it.</p> <ol style="list-style-type: none"> <li>1. Add the following two permissions: <ul style="list-style-type: none"> <li>• Type=Task Instance</li> <li>• Read</li> <li>• Update</li> <li>• Business Services=Collections</li> <li>• Unassigned to Business Service=deselect</li> <li>• Commands=None</li> </ul> <p>and</p> <ul style="list-style-type: none"> <li>• Type=Task Instance</li> <li>• Read</li> <li>• Update</li> <li>• Business Services=Letters of Credit</li> <li>• Unassigned to Business Service=deselect</li> <li>• Commands=All</li> </ul> </li> </ol>  <p>The screenshot shows the 'Opwise Permissions' tab for the 'Banking Operations' group. It displays a table with two rows of permissions:</p> <table border="1"> <thead> <tr> <th>Type</th> <th>Operations</th> <th>Commands</th> <th>Name</th> <th>Business Services</th> <th>Unassigned to Business Service</th> </tr> </thead> <tbody> <tr> <td>Task Instance</td> <td>Read, Update</td> <td>*</td> <td></td> <td>Collections</td> <td>false</td> </tr> <tr> <td>Task Instance</td> <td>Read, Update</td> <td>ALL</td> <td>*</td> <td>Letters of Credit</td> <td>false</td> </tr> </tbody> </table>	Type	Operations	Commands	Name	Business Services	Unassigned to Business Service	Task Instance	Read, Update	*		Collections	false	Task Instance	Read, Update	ALL	*	Letters of Credit	false
Type	Operations	Commands	Name	Business Services	Unassigned to Business Service														
Task Instance	Read, Update	*		Collections	false														
Task Instance	Read, Update	ALL	*	Letters of Credit	false														
	<ol style="list-style-type: none"> <li>2. Use the Group Members tab and <b>Edit</b> button to add Amy Hempel to the group.</li> </ol>																		

**Step 4** Add a third group called **IT Personnel** and save it.

1. Add the following two permissions: Type=Trigger
  - Create
  - Read
  - Update
  - Delete
  - All Business Services
  - Commands=None
2. Repeat the above permissions for Type=Task.

Type	Operations	Commands	Name	Business Services	Unassigned to Business Service
Trigger	Create, Read, Update, Delete	*	*	*	true
Task	Create, Read, Update, Delete	*	*	*	true

3. Add Tobias Wolff to the group.

**Step 5** Log in as each of the users and note that each is limited to those functions assigned to his or her group.

For additional information, see:

- [Users and Groups](#)