



Opswise Automation Center 5.1

Tutorials

© 2012 by Stonebranch, Inc. All Rights Reserved.



1. Introduction to the Tutorials	3
2. Tutorial - Creating and Manually Launching a Simple Task	4
3. Tutorial - Launching a Task Automatically Using a Simple Time Trigger	6
4. Tutorial - Launching a Task Every Monday Except Holidays	8
5. Tutorial - Launching a Task Every Two Hours During Workday	13
6. Tutorial - Launching an Email Task Based on a File Monitor	15
7. Tutorial - Creating a Simple Workflow	20
8. Tutorial - Running a Workflow with a Conditional Path	24
9. Tutorial - Running a Workflow with Skipped Criteria	33
10. Tutorial - Using Variables in a Simple Task	35
11. Tutorial - Using Variables in a Workflow	37
12. Tutorial - Launching Tasks at a Future Date and Time	38
13. Tutorial - Launching Tasks Using a Cron Trigger	39
14. Tutorial - Launching an Email Task Based on a Task Monitor	40
15. Tutorial - Running a Windows Task	48
16. Tutorial - Selecting Gauges for the Home Page	51
17. Tutorial - Making a Gauge	53
18. Tutorial - Adding Options to the Navigation Pane	54
19. Tutorial - Opswise Groups	56
20. Tutorial - Assigning Records to Opswise Groups	57
21. Tutorial - Taking Advantage of Opswise Groups	58
22. Tutorial - Create a Report Based on Opswise Groups	59
23. Tutorial - Viewing Activity by Group	60
24. Tutorial - Creating a Report	61
25. Tutorial - Scheduling and Distributing Reports	63
26. Tutorial - Setting Up a Virtual Resource	64
27. Tutorial - Getting the Most from the Activity Screen	65
28. Tutorial - Aborting a Process Launched by a Task	69
29. Tutorial - Creating Users and Assigning Permissions	71
30. Tutorial - Creating Security Groups and Assigning Permission	74

Introduction to the Tutorials

These **tutorials** guide you on a path through the system's features. They also provide links to more detailed information about each aspect of the system. Once you've completed the tutorials, you should have a basic understanding of how system features work together to automate your environment.

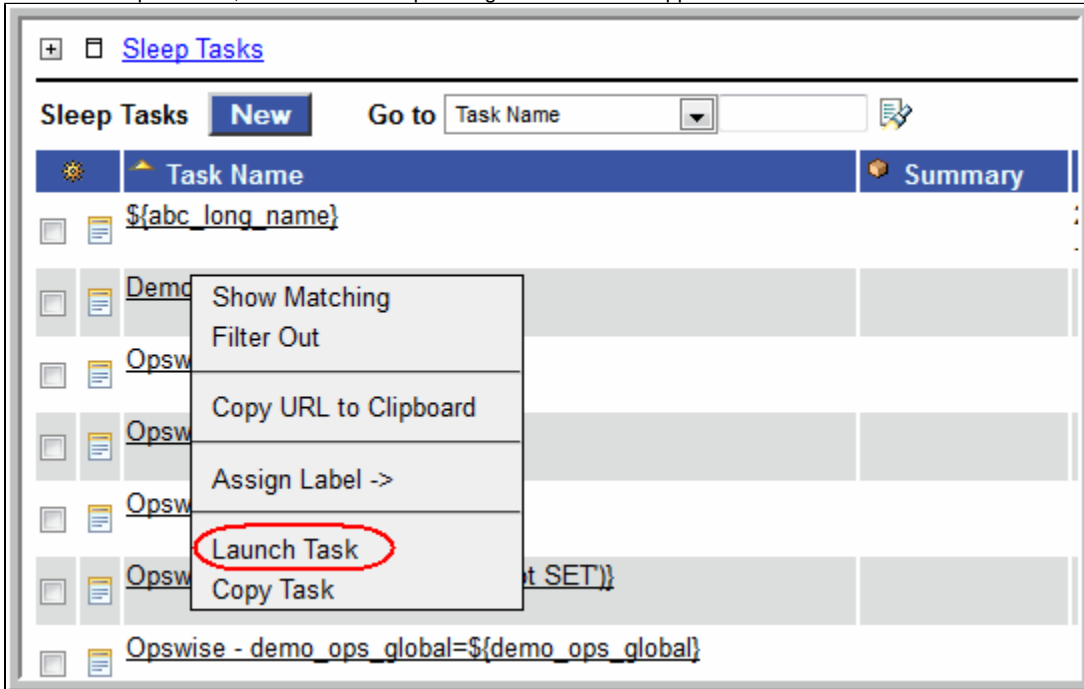
Before you begin, we recommend you watch a short (two minute) [movie](#) about how to navigate around Opswise Automation Center. Or, check out the [Getting Started](#) page.

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 re-enter data, do not delete the records.

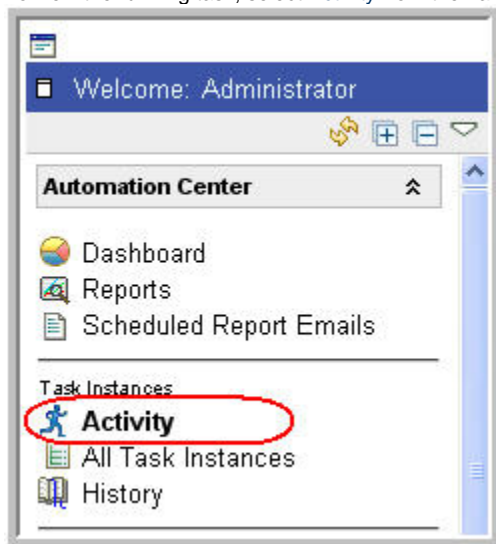
Tutorial - Creating and Manually Launching a Simple Task

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

1. From the navigation pane, select **Tasks > Sleep Task**.
2. Click **New**.
3. In the Task Name field, enter **Demo Sleep**.
4. In the Sleep Time field, enter **60**.
5. Click **Submit**.
6. From the Sleep Task list, locate Demo Sleep and right-click. A menu appears.



7. Select **Launch Task**.
8. To view the running task, select **Activity** from the navigation pane.



9. Locate the task instance Demo Sleep on the Activity display. When the task completes, the status changes from Running to Success.
10. To view more **details** about the task instance, click on the Instance name. The detailed record displays.


Sleep Task Instance		Update		Show Details		Re-run		Delete	
Instance Name:	Demo Sleep	Invoked By:	Manually Launched						
Task:	Demo Sleep	Member of Groups:							
Instance Reference Id:	1								
Sleep Type:	Seconds	Sleep Time (secs):	60						
Summary:									
Status:	SUCCESS								
Status Description:									
Start Time:	2009-08-11 14:30:44 -0400	End Time:	2009-08-11 14:31:44 -0400						
Duration:	1 Minute 0 Seconds								
Update		Show Details		Re-run		Delete			

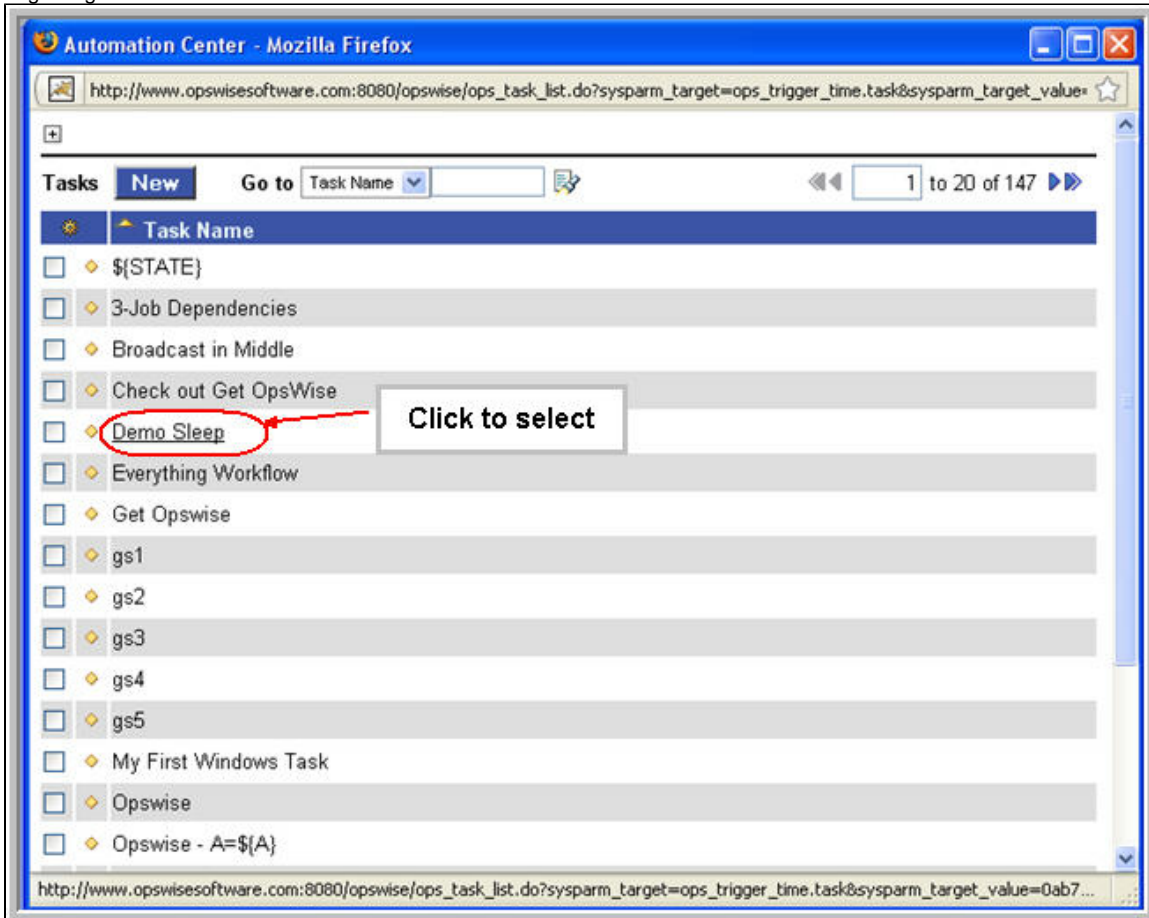
For additional information, see:

- Task Overview and Navigation (8 minute movie)
- Defining a Task (2 minute movie)
- Launching a Task (1 minute movie)
- Creating Tasks (user documentation)

Tutorial - Launching a Task Automatically Using a Simple Time Trigger

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

1. From the navigation pane, select **Triggers > Time Triggers**.
2. Click **New**.
3. In the Trigger Name field, type **Launch Sleep Every Minute**.
4. In the Task(s) field, click the lock icon.
5. Click the magnifying glass to **browse** for the Sleep task you created above. **Hint:** Type **D** and click the Search  icon to skip to tasks beginning with "D".



6. **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.
7. In the Time Style field, select **Time Interval**.
8. In the Time Interval field, type **1**.
9. In the Time Interval Units field, select **Minutes**.

Time Trigger Variables Versions

Time Trigger = Required field

Trigger Name: Launch Sleep Every Minute

Task(s): Demo Sleep

Calendar: System Default

Skip Count: 0

Skip Trigger if Active: ☐

Description:

Time Zone: System (US/Eastern)

Time Style: Time Interval

Time Interval: 1

Time Interval Units: Minutes

Starting Time: ☐

Restrict Times: ☐

Day Style: Simple

Daily: ☒

Enabled: ☐

Enabled By:

Forecast: ☐

Member of Groups: ☐

Version: 1

10. Click **Submit**.
11. By default, triggers are disabled. To enable this trigger:
 - a. From the trigger list, locate the Launch Sleep Every Minute trigger you just submitted.
 - b. Right-click the name of the trigger.
 - c. From the drop-down menu, select **Enable Trigger**. The system enables the trigger. Note that the Enabled column on the trigger list now displays a green check-mark for this trigger.
12. From the navigation pane, select **Activity**. Note that a new instance of Demo Sleep appears every one minute.
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:

[Triggers Overview \(user documentation\)](#)
[Time Trigger \(user documentation\)](#)
[Enabling and Disabling Triggers](#)

Tutorial - Launching a Task Every Monday Except Holidays

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

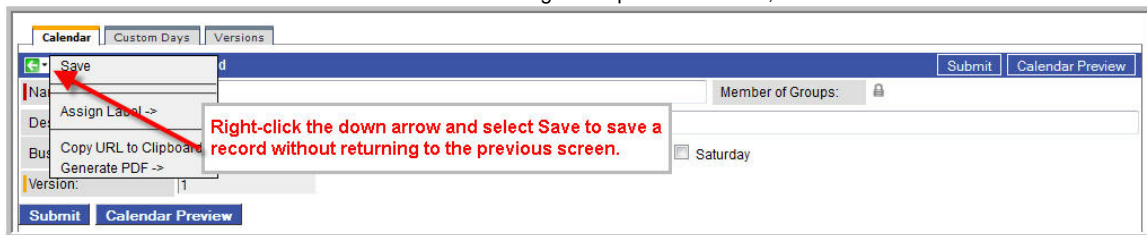
Introduction

In this tutorial, we will define a trigger for our Demo Sleep task 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 the system 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:

1. From the navigation pane, select **Calendars**.
2. Click the **New** button.
3. In the Name field, type My Holiday Calendar.
4. In the Comments field, type **Defines our company business days and holidays**.
5. Leave the default selections for business days (Monday through Friday).
6. 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. To save a record without returning to the previous screen, click the small down-arrow and select **Save**.



7. Click the **Custom Days** tab.
8. A number of Custom Days definitions are included in the default database. First we will select several of those for our calendar:
 - a. Click **Edit**. The Edit Members screen is available in many areas of Opwise. 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.
 - b. Press the CTRL key and click several of the holidays listed in the Collection list.
 - c. Click the **Add** button (right arrow). The days you selected are added to the Custom Days list.

Edit Members

Add Filter Run Filter

-- choose field --

Collection

Search

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
Ops - Veterans Day

Has Custom Days List

My Holiday Calendar

--None--

Add
Remove

Save Cancel

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

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

- d. Click **Save**. The holidays you selected are now shown in the Custom Days tab.
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:
 - a. From the Custom Days list, click **New**.
 - b. In the Name field, type **Inventory Day**.
 - c. 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.
 - d. In the Type field, select **Relative Repeating Date**. Three additional fields appear: When, Day of Week, and Month.
 - e. Make the following selections:
 - When = 3rd
 - Day of Week = Monday
 - Month = January

Custom Days Used by Calendars Versions

Custom Days Required field Submit

Name: Inventory Day Version: 1

Holiday: ☒

Description: Inventory performed on the third Monday of January

Type: Relative Repeating Date

When: 3rd
Day Of Week: Mon
Month: Jan

Submit

- f. Click **Submit** to add this custom day to the calendar.
- g. 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
- h. Click **Submit** to add this day to the calendar.
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

1. From the navigation pane, select **Time Trigger**.
2. Select **New**.
3. In the Trigger Name field, type **Every Monday at One, Tuesday on Holidays**.
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 window. The 'Trigger Name' field contains 'Every Monday at one, Tuesday on Holidays'. The 'Task(s)' field has a search box with 'D' entered, and a list of tasks is displayed below it, with 'Demo Sleep' selected. A red circle highlights the search box and the list. A callout box on the right says: 'Type "D" and a list of tasks beginning with "D" displays. Type as many letters as needed.'

5. In the Calendar field, select **My Holiday Calendar**.
6. In the Description field, type **Run every Monday at one, roll to Tuesday if Monday is a holiday**.
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.
8. First we will define a simple restriction. In the Day Style field, leave the default, **Simple**.
9. Enable the Specific Day field and select **Monday**.
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 system 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)

The screenshot shows the 'Time Trigger' configuration window with the following details:

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

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 **Complex Restriction** and make the following selections:

- Situation = On Holiday
- Restriction Mode = Or
- Restriction Adjective = Last
- Restriction Noun = Monday
- Restriction Qualifier = Year

Click **Submit**.

To enable this trigger, right-click on the trigger name in the list and select **Enable Trigger**.

List Qualifying Times

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

1. Display the Time Trigger you just created called **Every Monday at One, Tuesday on Holidays**.
2. Click **List Qualifying Times**. The system calculates the next twenty dates that this trigger will be satisfied and lists them in a new browser tab. In the example shown below, 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:

[Triggers Overview](#)

[Time Trigger](#)

[Calendars](#)

[Displaying Trigger Forecast Information](#)

Tutorial - Launching a Task Every Two Hours During Workday

In this exercise, you will define a recurring task that runs every two hours, limited to business hours -- Monday through Friday from nine a.m. to five p.m. You will also instruct the system 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 system will be instructed not to run the next instance.

1. From the navigation pane, select **Time Triggers**.
2. Click **New**.
3. In the Trigger Name field, type **Every Two Hours During Business Hours**.
4. Select our Demo Sleep task in the Task(s) field and My Holiday Calendar in the Calendar field.
5. Enable the **Skip Trigger if Active** field. This tells the system not to trigger the task if the previous instance of the task is still active.
6. In the Description field, type **Run every two hours, Monday thru Friday, 9 to 5; Do not run if holiday**.
7. Now we will specify a recurring time trigger of every two hours. In the Time Style field, select **Time Interval** and specify the following:
 - Time Interval = 2
 - Time Interval Units = Hours
8. 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:
 - Restrict Times = Enabled
 - Enabled Start = 09:00
 - Enabled End = 17:00
9. To specify that this trigger should run on week days, you could select either of the following:

Day Style = Complex
 Date Adjective = Every
 Date Noun = Business Day
 Date Qualifier = Year
 Both methods use the business days specified in the calendar.

Or
 Day Style = Simple
 Business Days = Enabled

To tell the system not to run the task on holidays, specify:

- Special Restriction = Enabled
- Situation = On Holiday

- Action = Do Not Trigger

Right-click the down-arrow and click **Save** to save the new trigger, then click **Enable Trigger** to activate it. Click **List Qualifying Times** to display the next 20 scheduled instances.

For additional information, see:

[Triggers Overview \(user documentation\)](#)

[Time Trigger \(user documentation\)](#)

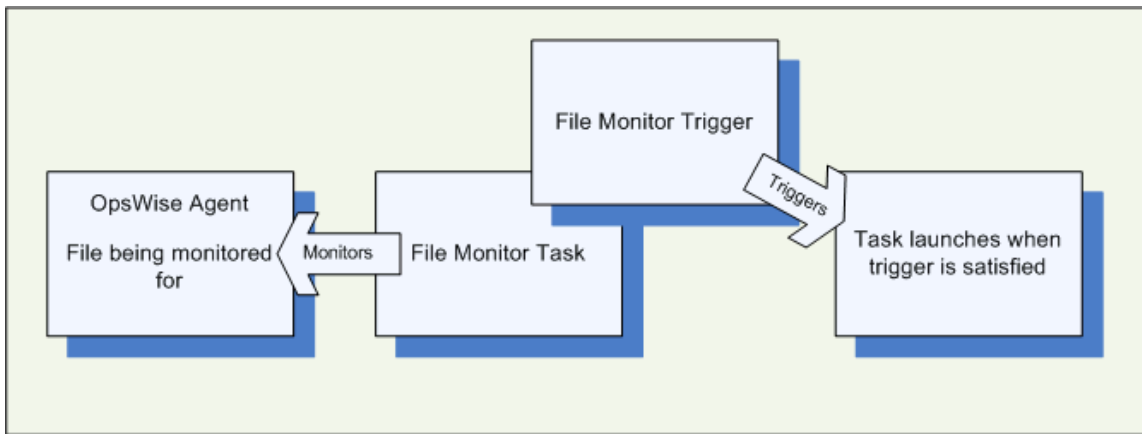
[Calendars \(user documentation\)](#)

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: An Opswise agent, a File Monitor task, a File Monitor trigger, and the Email task being launched by the trigger.



Prerequisites:

- In order to perform this exercise, you need a Windows, Unix, Linux, 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. (**Hint:** 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.

1. From the navigation pane, select **Tasks > File Monitors**.
2. Click **New**.
3. In the Task Name field, type **Demo File Monitor**.
4. In the Agent field, select the agent you are using for the exercise.
5. In the Monitor Type field, leave the default **Create**. See [File Monitor field descriptions](#) for details about the other file monitor options.
6. In the Monitor File field, type **XYZ.TXT**. Since we have not specified any directory, the system will search the root directory.
7. Enable the Recursive field. Since we are going to write our file to the OPS TUTORIAL directory, we want the system to search all subdirectories for the file.
8. In the Stable field, type **5**. This tells the system to satisfy the trigger only when the file has not changed in 5 seconds.

The screenshot shows the 'File Monitor' configuration window in the Opwise Automation Center. The window has a tabbed interface with 'File Monitor' selected. Below the tabs, there's a 'Required field' indicator and a 'Submit' button. The form is divided into several sections:

- Task Information:** Task Name (Demo File Monitor), Version (1), Agent (Opwise), Agent Variable (checkbox), Credentials (text field), Credentials Variable (checkbox), Agent Cluster (text field), Agent Cluster Variable (checkbox), and Cluster Broadcast (text field).
- Task Description:** A large text area for describing the task.
- Member of Groups:** A dropdown menu.
- User Estimated Duration:** A time picker set to 00:00:00.
- Monitor Type:** A dropdown menu set to 'Create'.
- Monitor File(s):** A text field containing 'XYZ.TXT'.
- Recursive:** A checkbox that is checked.
- Stable (seconds):** A text field containing '5'.
- File Owner:** A text field.
- Maximum Files:** A text field.
- Scan Text:** A text field.
- Late Start:** A checkbox.
- Late Finish:** A checkbox.
- Early Finish:** A checkbox.

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

9. Click **Submit**.

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**:

1. From the navigation pane, select **Tasks > Email Tasks**.
2. Click **New**.
3. In the Task Name field, type **Send Email When File Appears**.
4. In the Email Connection field, select the [Email Connection](#) you set up for the exercise.
5. In the To field, type in your email address. This is where the email will be sent.
6. In the Subject field, type **File XYZ arrived**.
7. In the body field, we are going to use an [Opwise variable](#) and an [Opwise function](#). Type the following:

```
Triggered by: ${ops_trigger_name}
Date: ${_date}
```

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.

Email Task | Task Virtual Resources | Variables | Actions | Task Instances | Triggers | Notes | Versions

Email Task | Required field | Update | Launch Task | Delete

Task Name: Send Email When File Appears | Email Template:

Version: 1 | Hold on Start: ☐

Task Description:

Member of Groups:

Email Connection: Opswise - Gmail Account |

Reply-To:

To: can@gmail.com

Cc:

Bcc:

Subject: File XYZ arrived

Body:

Triggered by: \${ops_trigger_name}

Date: \${_date}

Late Start: ☐

Late Finish: ☐

Early Finish: ☐

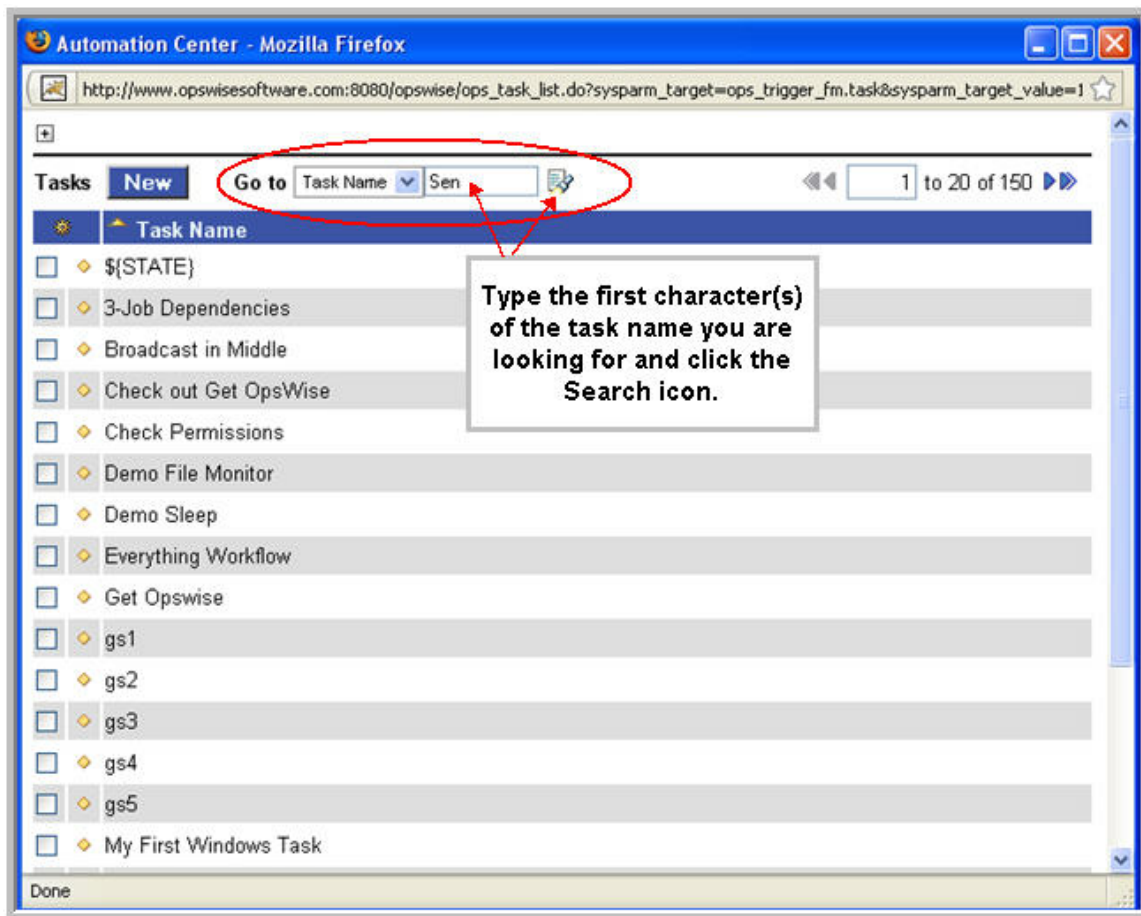
Update | Launch Task | Delete

9. Click **Submit**.

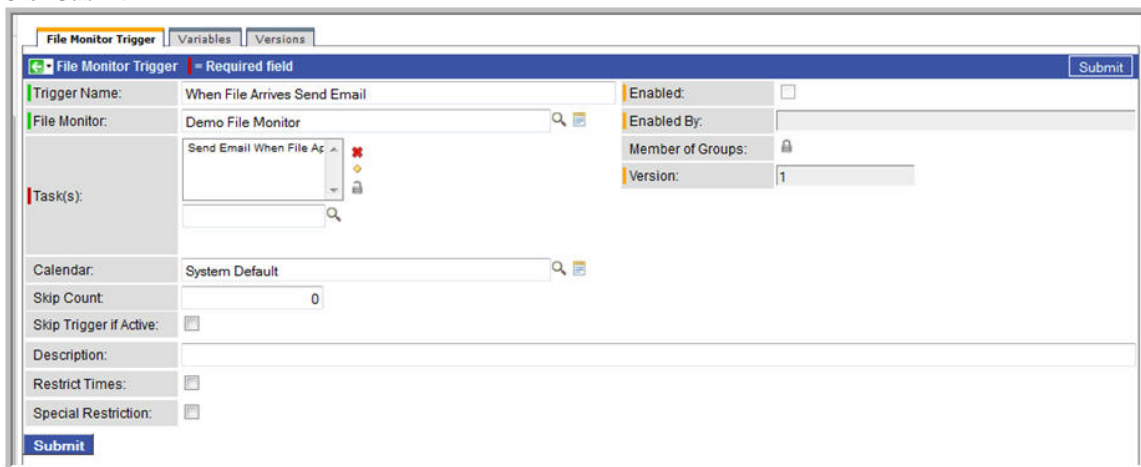
Create File Monitor Trigger

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

1. From the navigation pane, select **Triggers > File Triggers**.
2. Click **New**.
3. In the Trigger Name field, type **When File Arrives Send Email**.
4. In the File Monitor field, select the File Monitor task you defined before, **Demo File Monitor**.
5. In the Task(s) field, select the Email Task you defined before, **Send Email When File Appears**. **Hint:** If you have a lot of task definitions, can jump to the task you want by typing the first few characters into the Go to field in the popup window and clicking the search icon.



6. Click **Submit**.

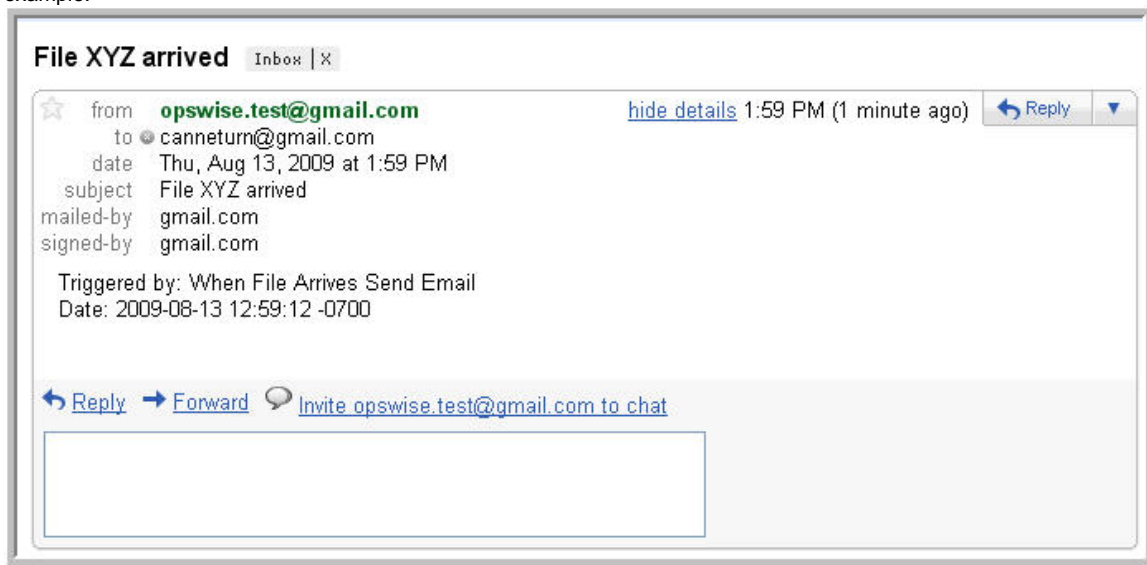


Test Your File Monitor Setup

The final step is to test the entire setup.

1. **Enable the File Monitor trigger.** This launches the File Monitor Task. It will appear in the [Activity display](#) with a status of Running.
2. Do one of the following:
 - 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.
 - 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:
 - a. Select **Triggers > File Triggers**.
 - b. Locate the When File Arrives Send Email trigger and right-click the name.
 - c. Select **Trigger Now**.

3. 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.
4. 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:



For additional information, see:

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

Tutorial - Creating a Simple Workflow

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

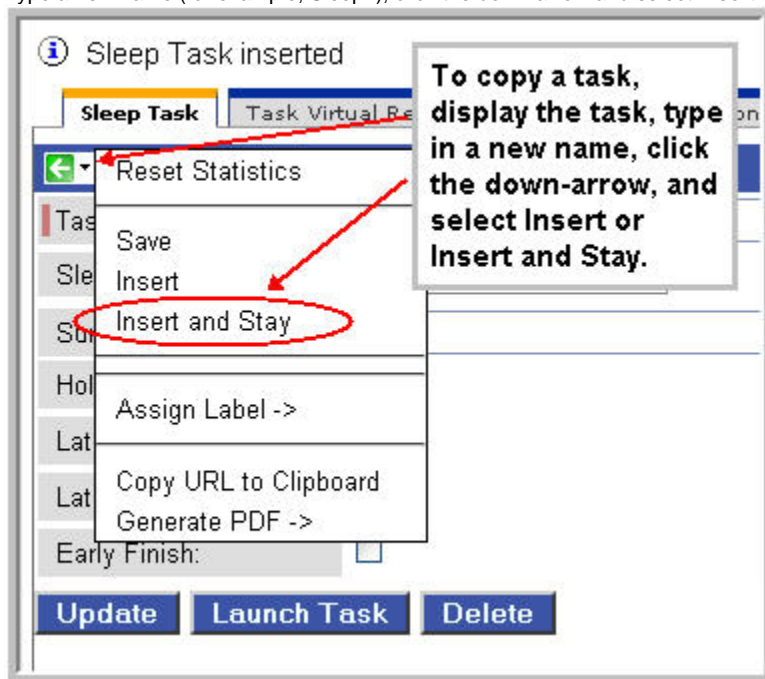
Introduction

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.

Copying Tasks

Before you begin, 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). **Hint:** You can [copy tasks](#) using a couple different methods. One method is provided below:


1. Create the first task and use the down-arrow to save it.
2. Type a new name (for example, Sleep2), click the down-arrow and select **Insert and Stay**.



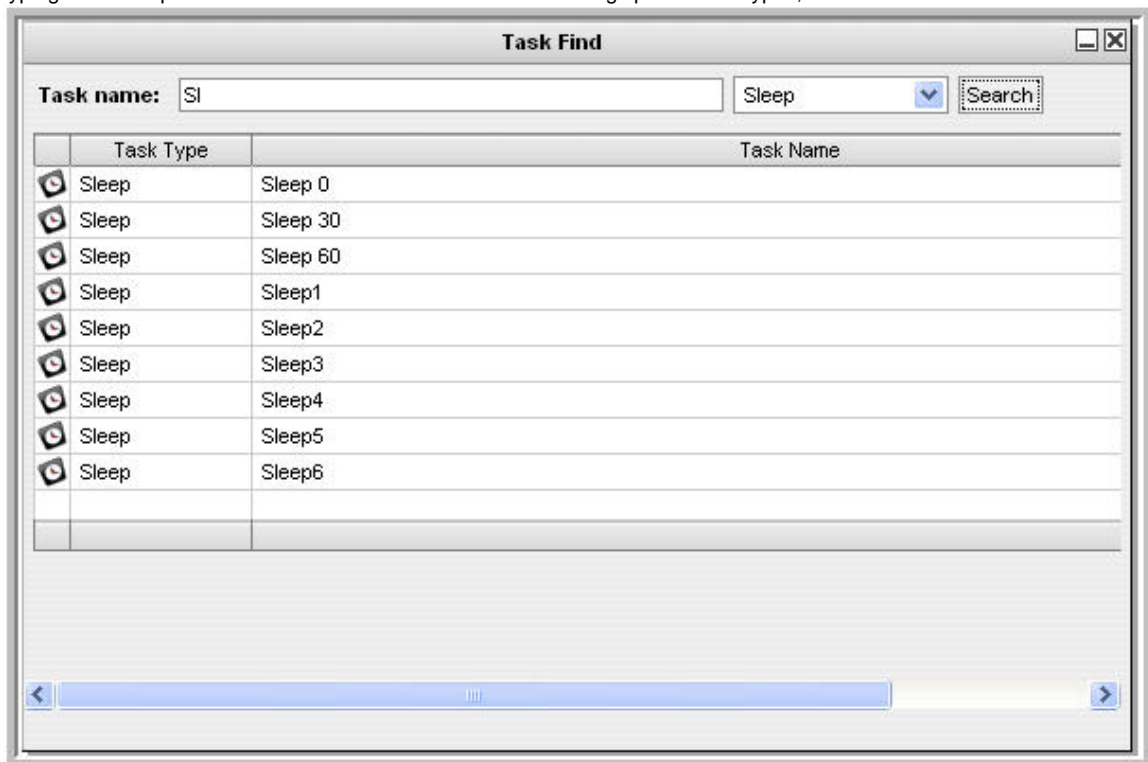
3. Repeat step 2 for the remaining Sleep tasks.

Creating a Simple Workflow

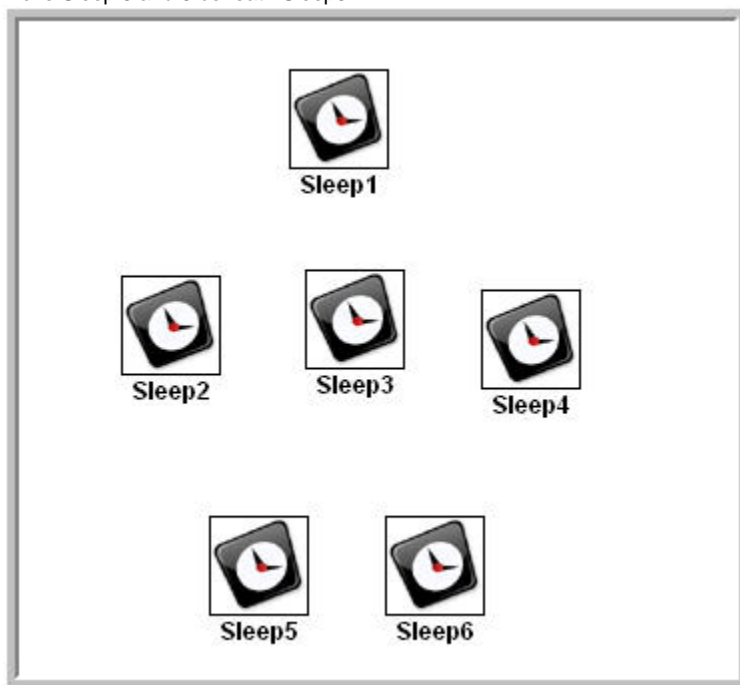
Now that we have six Sleep tasks to work with, we are ready to create a simple workflow:


1. From the navigation pane, select **Tasks > Workflow Tasks** and click **New**.
2. In the Task Name field, type **Simple Workflow**.
3. Use the down-arrow to [save the record](#) without exiting the screen. Once the record is saved, note the additional buttons that have appeared: Update, Edit Workflow, Launch Task, and Delete.
4. Click **Edit Workflow** to access the workflow editor.
5. 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. To add a new task, click the **Add Task**  icon.


- a. 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.




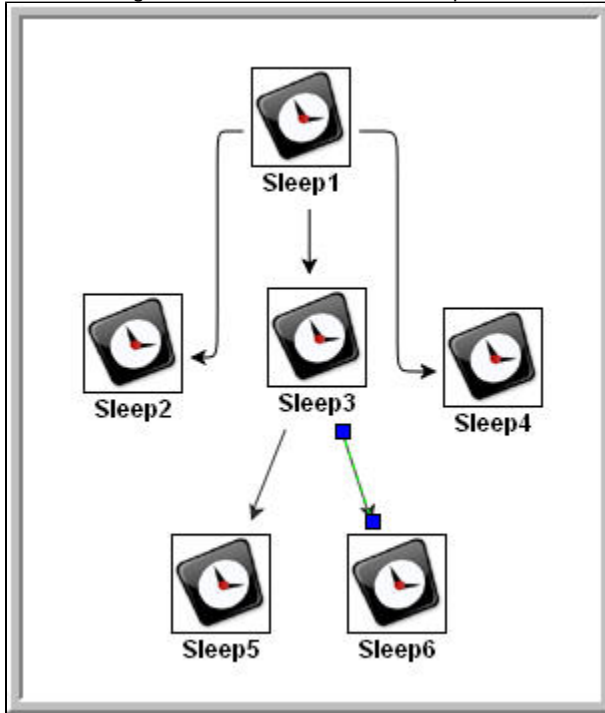
- b. Click and drag the icon onto the canvas for the Sleep1 task. Repeat the process for the remaining Sleep tasks. When you are finished, close the Task Find window.
6. Now we will arrange our workflow as follows:
- Sleep1 is the top-level task
 - Sleep 2, 3 and 4 are dependent on Sleep1.
 - Sleep 5 and 6 are dependent on Sleep 3.
- To create these connections:
- Click and drag Sleep1 to the top of the canvas and place Sleep 2, 3, and 4 in a line beneath it.
 - Move Sleep 5 and 6 beneath Sleep 3.



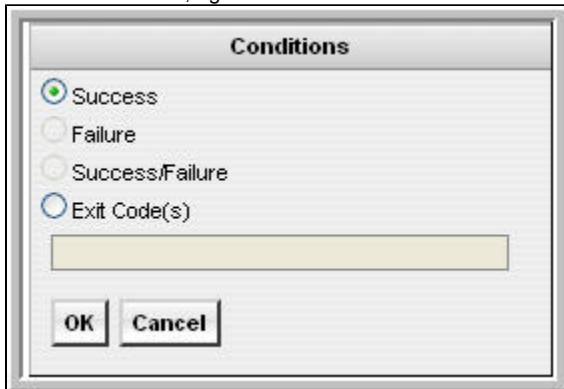
7. Now we are ready to create connections.
- Click the curved connector icon . This puts you in connect mode.
 - Click Sleep1, drag the cursor down to Sleep2, then release the mouse. A curved connection appears.
 - Repeat for Sleep3 and Sleep4.

d. For the next set of connections, we will 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:












The default conditions (or dependency) for connectors is Success. That is, the downstream task runs if the upstream task goes to success. To view the conditions, right-click on a connector and click **Conditions**. The Conditions window displays.



Note that a Sleep task cannot go to Failure so those options are greyed out.

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.

1. Click the back arrow to return to the Workflow record.
2. Click **Launch Task**.
3. From the navigation pane, select **Task Instances > Activity**. 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, 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 downstream 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 the system 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:

1. From the navigation pane, select **Tasks > SQL Tasks** and click **New**.
2. In the Task Name field, type **SQL Create Table**.
3. In the Database Connection field, select the database connection you created as a prerequisite.
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));
```


5. Click **Submit**.
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');
```

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';
```

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.

1. Select **Tasks > Manual Tasks** and click **New**.
2. Give it the name **Pause for Manual**.
3. 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 \$_date().
4. Enable the Late Finish field and select the following values:
 - Late Finish Type = Duration
 - Late Finish Duration = Hours 00 02 00 (2 minutes)
5. Click the down-arrow and select **Save**. Your task should look like this:

Now we are going to 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

In the Text field, type:

Make sure database is running.

Click **Submit**.

The screenshot shows a web form titled 'Note'. It has a 'Title' field with the text 'Probable database problem' and a 'Text' area with the text 'Make sure database is running.'. At the bottom, there are two 'Updated:' fields and a 'Submit' button.

Now we are going to 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}`

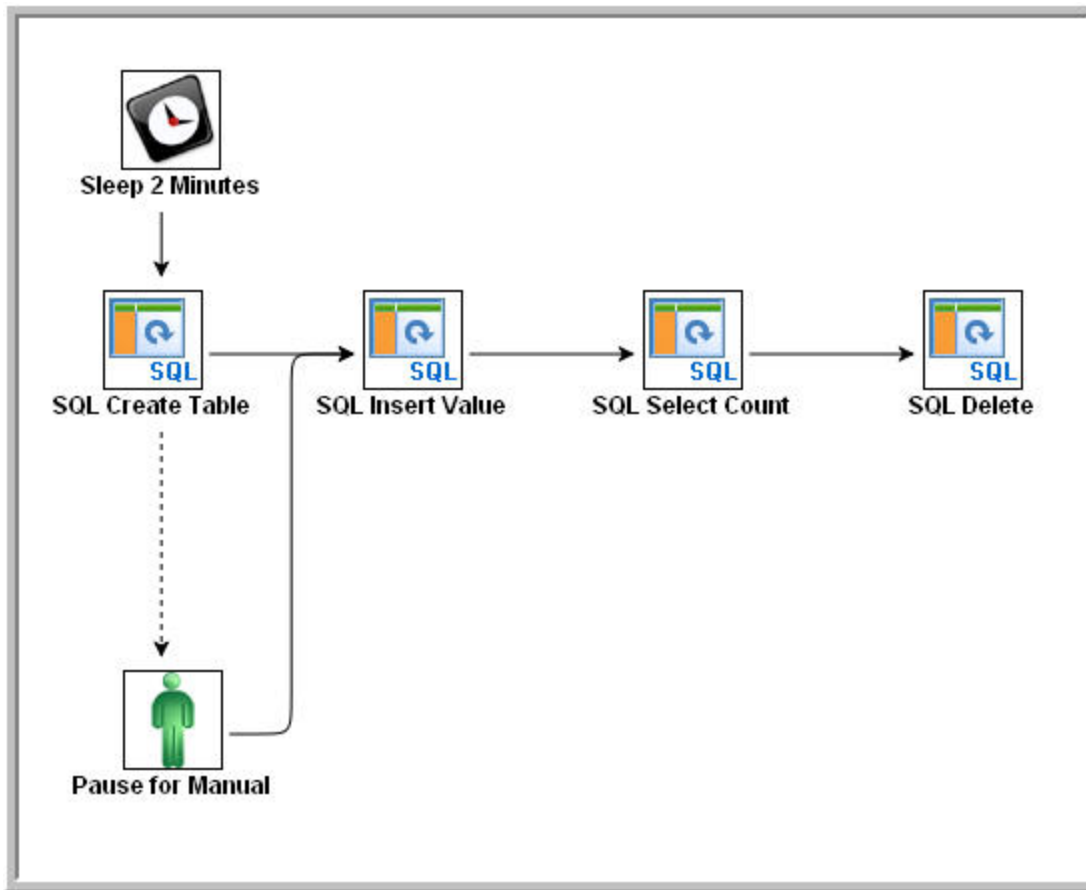
Note: For detailed descriptions, see [Creating Email Notifications](#).

8. Click **Submit**.

Creating the Workflow

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

1. Select **Tasks > Workflow Tasks** and click **New**.
2. Give it the name Bigger Workflow and click **Down-arrow > Save** to save it.
3. Click **Edit Workflow**.
4. Use the **Add Task** tool to drag the tasks you just created onto the canvas:
 - Sleep 2 Minutes
 - Pause for Manual
 - SQL Create Table
 - SQL Delete
 - SQL Insert Value
 - SQL Select Count
5. Organize the tasks as shown in the illustration below.
6. Create the connections shown in the illustration below. If you don't remember how, review the [Creating a Simple Workflow](#) exercise.
7. The Success connectors tell the system 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 system should run the Pause for Manual task:
 - a. Right-click the connector between SQL Create Table and Pause for Manual.
 - b. When the menu pops up, select **Conditions**.
 - c. Enable **Failure**.
 - d. Click **OK**. Note that the connector is a dotted line, which indicates a Failure connection.
8. On the toolbar, click **Save** .



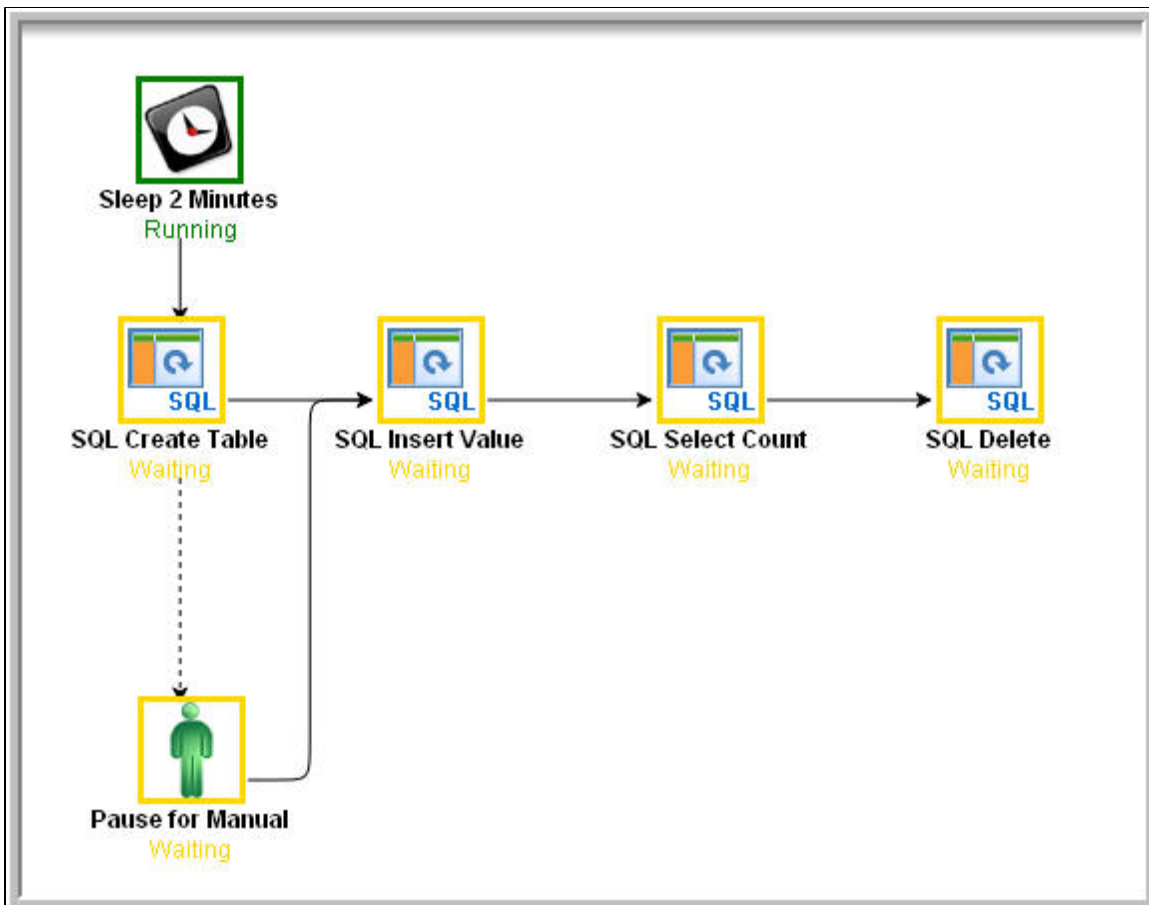
Run the Workflow to Success

We are going to launch our workflow and view it from two different perspectives: from the Activity Screen view and the workflow monitor (the canvas).

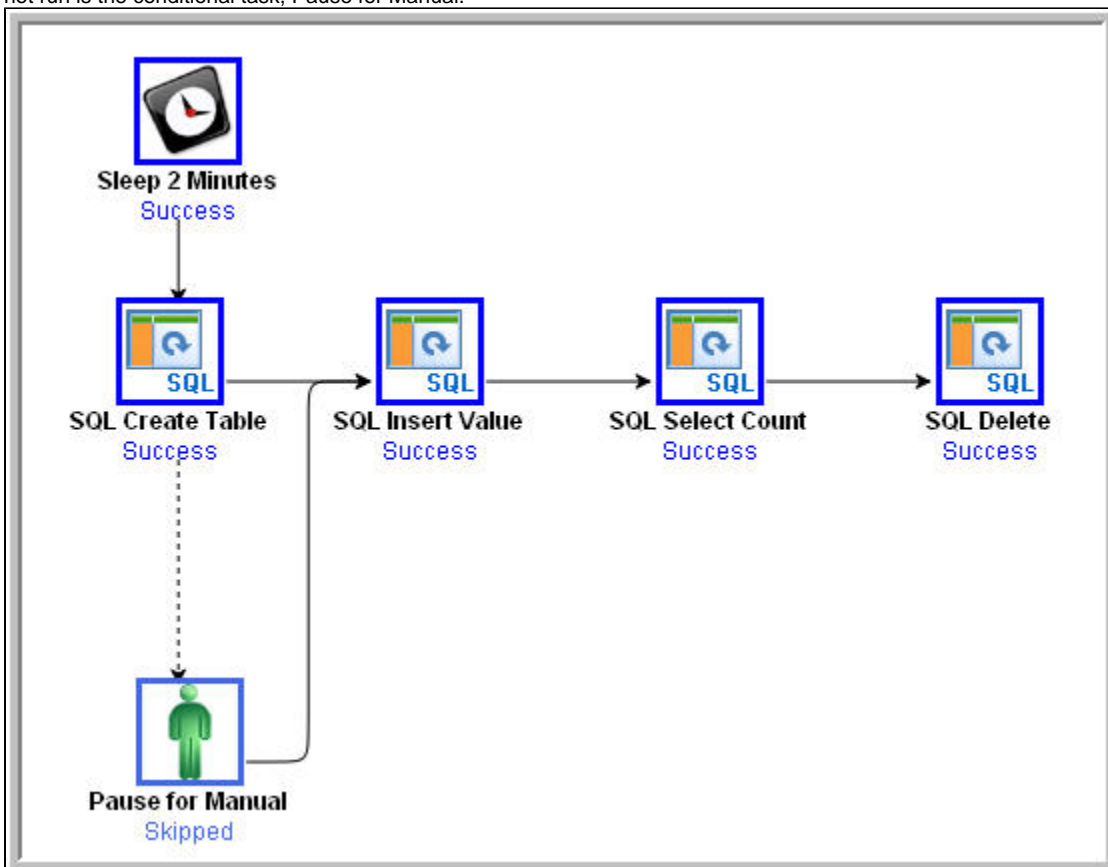
1. [Launch the workflow manually.](#)
2. Display the Activity screen. Because the Sleep 2 Minutes task is still running, your display should look something like this:

Today's Task Instances by Created Time			
Instance Name	Type	Duration	Status
Pause for Manual	Manual		Waiting
Sleep 2 Minutes	Sleep		Running
Bigger Workflow	Workflow		Running
SQL Insert Value	SQL		Waiting
SQL Select Count	SQL		Waiting
SQL Delete	SQL		Waiting
SQL Create Table	SQL		Waiting

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



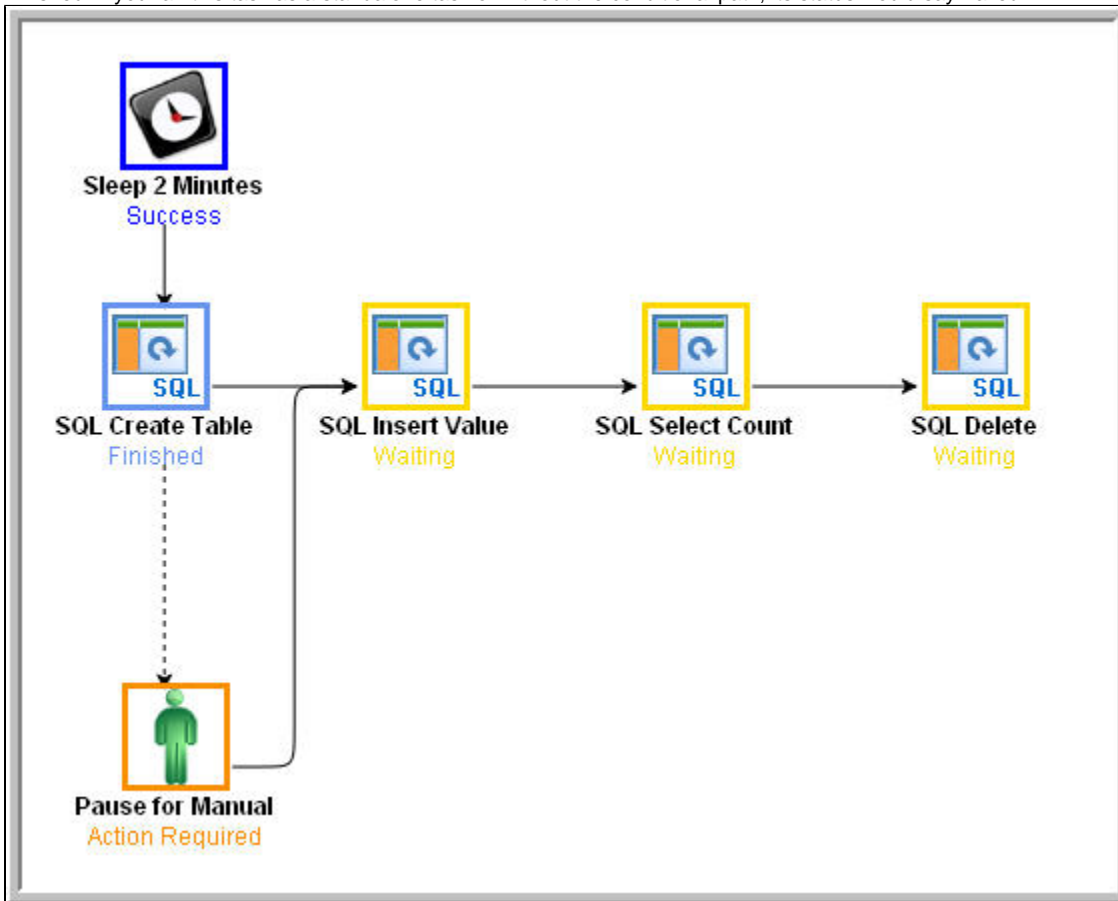
4. Once the Sleep task finishes, the SQL tasks execute. All tasks go to Success and the workflow goes to Success. The only task that did not run is the conditional task, Pause for Manual.



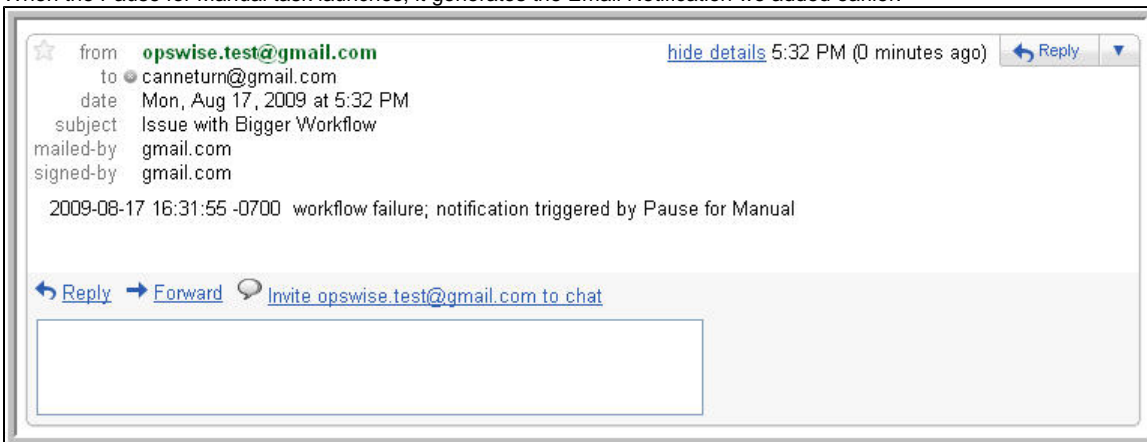
Run the Workflow Down the Conditional Path

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.

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



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

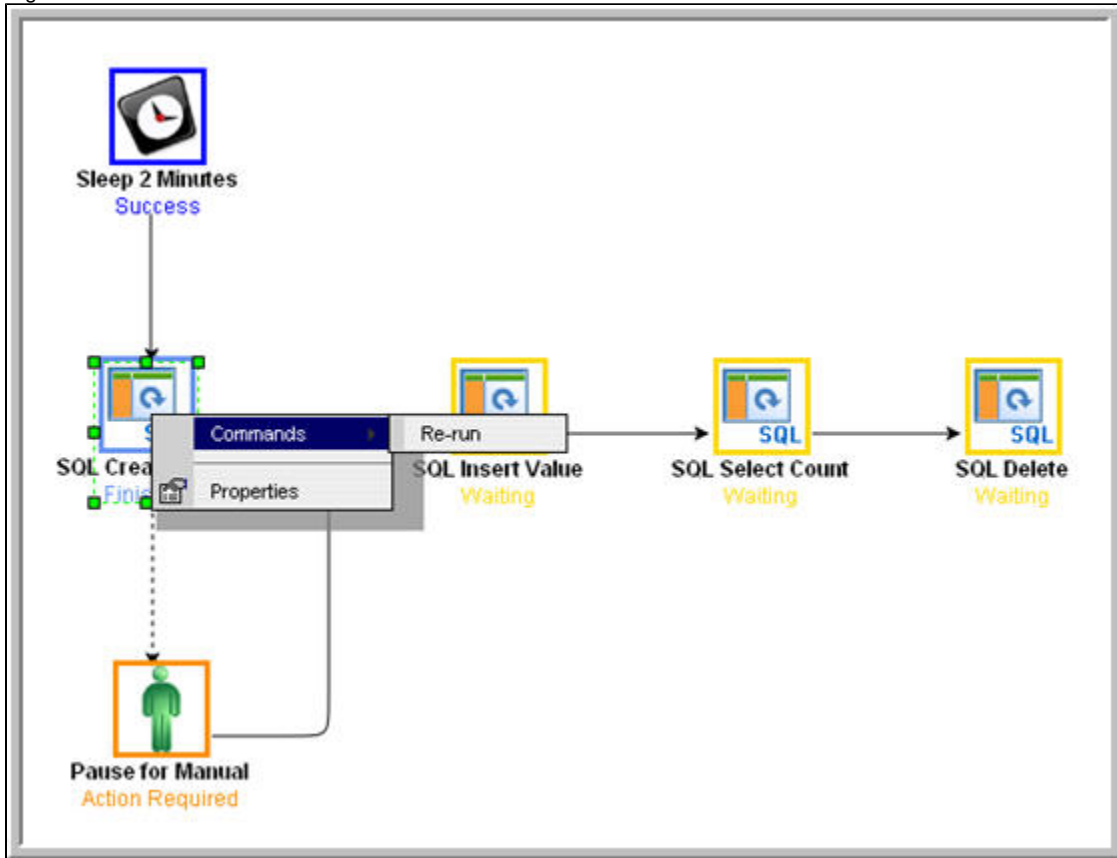


The user receives the email, which provides the name of the task that generated it (Pause for Manual in our case). The user might also be running a special Activity screen that displays only Manual tasks in the Action Required status. According to our scenario, the user opens the Pause for Manual task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

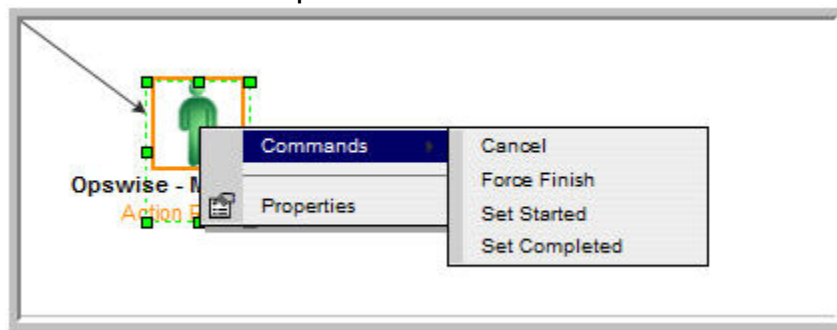
to check the database and bring it back up.

At this point, the workflow processing could continue in one of two ways: 1) Re-run the task that failed and send the workflow down the success path or 2) set the Pause for Manual task to Completed status and continue the workflow from there. We will try both methods.

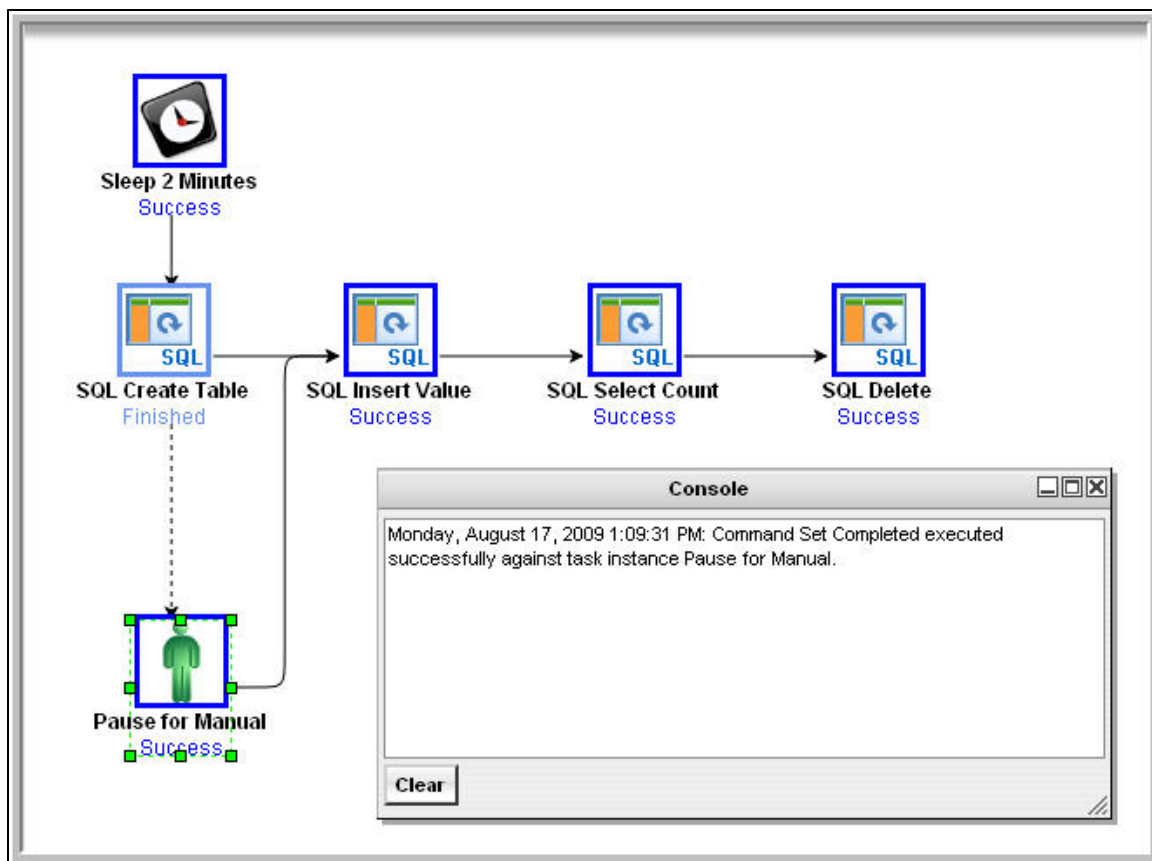
- Re-run the failed task. In a real processing situation, this is the method you would use because you need to create the table before you can continue.
 - Right-click the task that failed and select **Commands > Re-run**.



- 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.
- 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.
 - Right-click on the Manual task.
 - Select **Commands > Set Completed**.



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



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

For additional information, see:

[Activity display \(user documentation\)](#)
[Monitoring Workflows \(user documentation\)](#)
[Database Connection \(user documentation\)](#)
[SQL Task \(user documentation\)](#)
[Manual Task \(user documentation\)](#)
[Creating Email Notifications \(user documentation\)](#)
[Runbook Notes \(user documentation\)](#)
[Creating Conditional Paths \(user documentation\)](#)
[Adding Skip/Run Criteria \(user documentation\)](#)

Tutorial - Running a Workflow with Skipped Criteria

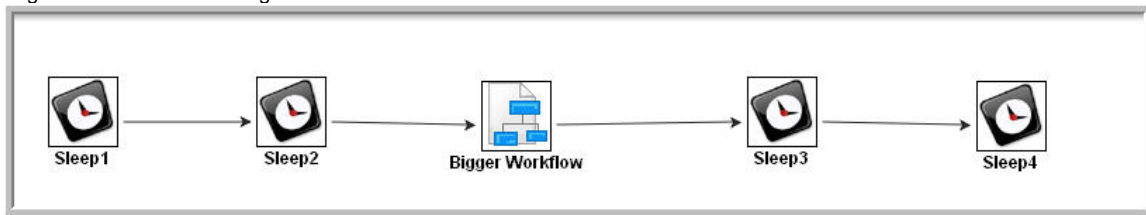
- [Introduction](#)
- [Create the Workflow](#)
- [Run the Workflow](#)
- [Check the Skipped Workflow's History](#)


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



1. Create a new workflow called Workflow with Skipped, access the Workflow Editor and add the following tasks which you created in previous exercises:
 - Sleep1, Sleep2, Sleep3, Sleep4
 - Bigger Workflow
2. Organize the workflow using all Success conditions as shown in this illustration.



3. Click Save 
4. Right-click Bigger Workflow and select **View/Edit Run Criteria**. The system navigates to the Run Criteria list.
5. Click **New**. The Task Run Criteria screen displays.
6. In the Type field, select **Skip Criteria**.
7. Select **Specific Day(s)**.
8. Select the current day (that is, if today is Friday, select Fri).

Task Run Criteria - Required field

Type:

Task Id:  

Vertex Id:

Business Day: ☐

Holiday: ☐

Specific Day(s): ☒

Custom Day: ☐

Complex: ☐

Variable: ☐

Submit

Sun: ☐

Mon: ☐

Tue: ☐

Wed: ☐

Thu: ☐

Fri: ☒

Sat: ☐

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

9. Click **Submit**.

Run the Workflow

1. Right-click anywhere in the Workflow Editor and select **Launch Workflow**.
2. Display the Activity screen and note that the child workflow (Bigger Workflow) and all its child 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.		
Pause for Manual	Manual	Skipped		2009-08-27 11:26:07 -0700			
Sleep 2 Minutes	Sleep	Skipped		2009-08-27 11:26:07 -0700			
SQL Create Table	SQL	Skipped		2009-08-27 11:26:07 -0700			
SQL Delete	SQL	Skipped		2009-08-27 11:26:07 -0700			
SQL Insert Value	SQL	Skipped		2009-08-27 11:26:07 -0700			
SQL Select Count	SQL	Skipped		2009-08-27 11:26:07 -0700			
Bigger Workflow	Workflow	Skipped		2009-08-27 11:26:07 -0700			
Sleep4	Sleep	Success	2009-08-27 11:26:37 -0700	2009-08-27 11:26:47 -0700	10 S		
Sleep3	Sleep	Success	2009-08-27 11:26:27 -0700	2009-08-27 11:26:37 -0700	10 S		
Sleep2	Sleep	Success	2009-08-27 11:26:17 -0700	2009-08-27 11:26:27 -0700	10 S		
Sleep1	Sleep	Success	2009-08-27 11:26:07 -0700	2009-08-27 11:26:17 -0700	10 S		
Workflow with Skipped	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 **All Task Instances** screen.

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

Workflow Task Instance		Update	View Parent	Show Details	View Workflow	Delete
Instance Name:	Bigger Workflow	Invoked By:	Workflow: Workflow with Skipped			
Task:	BiggerWorkflow	Member of Groups:				
Instance Reference Id:	4					
Summary:						
Status:	SKIPPED					
Status Description:	Skipped due to run/skip criteria.					
Start Time:		Duration:				
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:				
Update View Parent Show Details View Workflow Delete						

3. 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

In a previous exercise, [Launching an Email Task Based on a File Monitor](#), we included a built-in variable called `${ops_trigger_name}` and a system variable called `${_date}` 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.

Prerequisite: You need an Email Connection to perform this exercise.

Resolving Variable Using Value from Global Variable Table

1. From the navigation pane, select **Variables** and click **New**.
2. Give the variable the name **Tutorial** and a value of **Global**.
3. Click **Submit**.
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}

Save and launch the Email task.

Resolving Variable Using Value from Task

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

Resolving Variable Using Value from Trigger

1. Create a new Time trigger with the following values:

- Trigger Name=Variable Demo
 - Tasks=Email Tutorial
 - Time=a couple minutes from now
2. Save the Time trigger.
 3. Click the **Variables** tab and add the Tutorial variable with a value of **Trigger**.
 4. Enable the trigger.
 5. 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

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 the system looks within the workflow or parent workflow(s), with the global variable coming last.

Prerequisite: You need a working database connection to do this exercise.

1. Create a SQL task called **SQL With Variable** with the following SQL command.

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

2. Within the task, define a variable called **tutorial**, with a value of **task**.
3. Submit the SQL task.
4. Create a new workflow called **Variable Workflow**.
5. Add the SQL With Variable task and save the workflow.
6. 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.

```
CREATE TABLE task20090913 (name varchar(128), value varchar(128));
```

7. Open the task and delete the task variable.
8. Go back to Variable Workflow and add the following variable:

```
tutorial/workflow
```

9. Open the task instance. The SQL command used the variable from the workflow because the task no longer had a variable.

```
CREATE TABLE workflow20090913 (name varchar(128), value varchar(128));
```

For additional information, see:

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

Tutorial - Launching Tasks at a Future Date and Time

In this exercise, we will launch several tasks at a single future date and time (a couple minutes from now). For this exercise, we will use the SQL tasks we created before.

Create a Temporary Trigger

1. From the navigation pane, select **Triggers > Temporary Triggers** and click **New**.
2. In the Trigger Name field, type **Launch SQL Tasks**.
3. 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
4. In the Date field, leave the default set to the current date.
5. In the Time field, enter a time a couple of minutes in the future, using 24-hour time.
6. In the Time Zone field, select your timezone.

7. Right-click the down-arrow and select **Save**.
8. Enable the trigger.

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

In this exercise, we will use the Cron trigger to launch the same tasks we launched in the previous exercise, Launching Tasks Using Temporary Triggers.

Create a Cron Trigger

1. From the navigation pane, select **Triggers > Cron Triggers** and click **New**.
2. In the Trigger Name field, type **Launch SQL Tasks Using Cron**.
3. 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
4. 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.
5. 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. The system uses the timezone of the Opwise server.
6. Leave asterisks in the remaining fields.

The screenshot shows the 'Cron Trigger' configuration window. The 'Trigger Name' is 'Launch SQL Tasks Using Cron'. The 'Task(s)' field contains a search for 'SQL' and lists four tasks: 'SQL Create Table', 'SQL Delete', 'SQL Insert Value', and 'SQL Select Count'. The 'Calendar' is set to 'System Default'. The 'Skip Count' is 0. The 'Skip Trigger if Active' checkbox is unchecked. The 'Description' field is empty. The 'Minutes' field is 16, 'Hours' is 15, 'Day of Month' is *, 'Month' is *, and 'Day of Week' is *. The 'Cron Criteria' field shows '16 15 ***' and the 'Next Scheduled Time' is displayed. The 'Special Restriction' checkbox is unchecked. The interface includes tabs for 'Cron Trigger', 'Variables', and 'Versions', and buttons for 'Update', 'List Qualifying Times', 'Enable Trigger', 'Trigger Now', and 'Delete'.

7. Right-click the down-arrow and select **Save**.
8. Enable the trigger.

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](#)

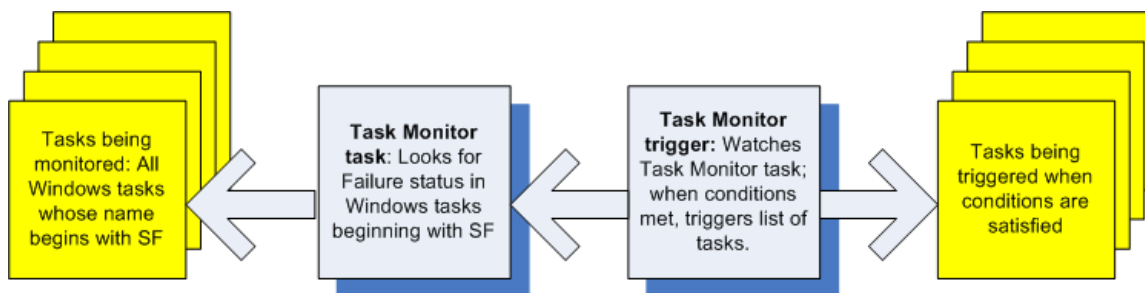
Tutorial - Launching an Email Task Based on a Task Monitor

- 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, the system 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.

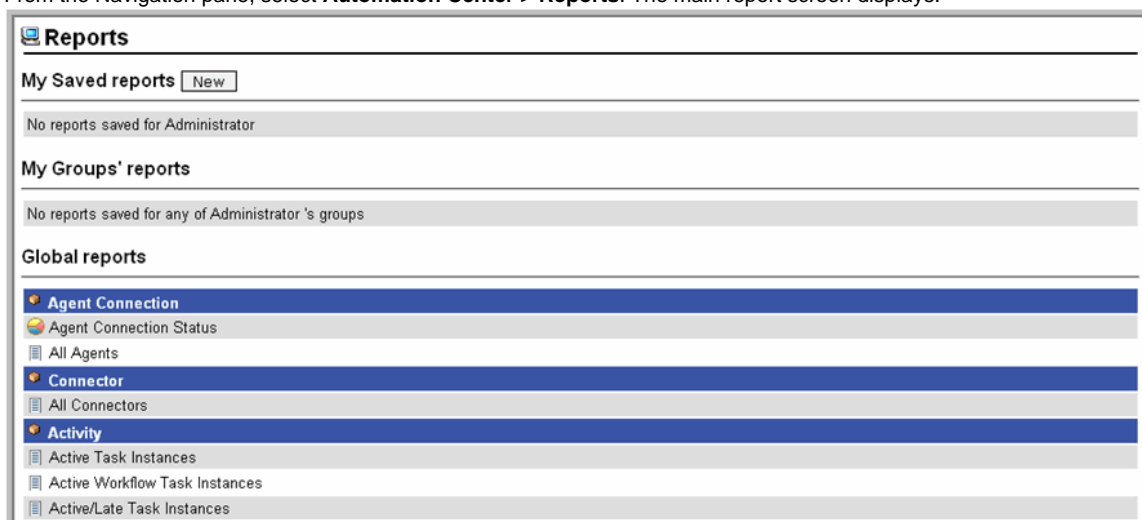
The following illustration shows the various components used to trigger tasks based on the status of other tasks.



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 Opwise, this filter is referred to as a report. For more detailed explanations of each field, see [Reports](#).

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.

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

3. In the Name field, enter **Problem Task Instances**.
 4. In the Visible to field, select **Everyone**.
 5. In the Type field, select **List**.
 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.
 7. Skip the **Group by** field.
 8. In the Filter and Order fields, click the plus (+) sign and select the following:
 - a. In choose field, select **Status**.
 - b. In the operator field, select **is one of**.
 - c. In the last field, use CTRL+Click to select all of: **Undeliverable, Running/Problems, In Doubt, Start Failure, Cancelled, Failed**.
 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
- Hint: 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.
10. To save the report, click **Save**.

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

1. Open a new browser tab.
2. From the first tab, click and drag the Activity option to the new tab.
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

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

Today's Task Instances by Created Time	
Active Task Instances	
Active Workflow Task Instances	
Active/Late Task Instances	
All Task Instances by Status	
Cancelled Task Instances	
Finance Tasks	
Held Task Instances	
In Doubt Task Instances	
PeopleSoft Jobs	
Problem Task Instances	
Queued Task Instances	
Running Task Instances	
Today's Failed Task Instances by Status	
Today's Successful Task Instances	
Today's Task Instances by Created Time	
Triggered by Task Status	
My First Windows Task	
Copy file from Point A to Point B	
Triggered by Task Status	
Pause for Manual	
Monitors SQL Create Table	
SQL Create Table	
SQL Create Table	
Triggered by Task Status	

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

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).

Problem Task Instances			
Status	Type	Instance Name	Invoked By
Cancelled	Manual	Pause for Manual	Workflow: Bigger Workflow
Cancelled	Manual	Pause for Manual	Workflow: Bigger Workflow
Failed	SQL	SQL Create Table	Manually Launched
Failed	SQL	SQL Create Table	Manually Launched
Failed	SQL	SQL Create Table	Manually Launched
Failed	SQL	SQL Create Table	Manually Launched
Failed	SQL	SQL Create Table	Manually Launched
Failed	SQL	SQL Create Table	Manually Launched
Failed	SQL	SQL Delete	Workflow: Bigger Workflow2
Failed	SQL	Opwise - SQL Create	Workflow: Bigger Workflow
Failed	FTP	Opwise - FTP Download	Manually Launched
Failed	Linux/Unix	Opwise - FTP Download	Workflow: A Workflow with UNIX and Stored Procedures
Failed	Linux/Unix	Linux Failure Exit Code 10	Workflow: Opwise - Conditional
Failed	Linux/Unix	Linux Failure Exit Code 10	Workflow: Opwise - Conditional
Failed	Linux/Unix	Linux Failure Exit Code 10	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.

1. From the navigation pane, select **Resources > Email Templates**.
2. For Template Name, type **Notification based on status**.
3. For Email Connection, select the connection you defined for earlier exercises.

- For the To field, enter a usable email account that you can use for the tutorial.
- For Subject, type:

Task failure alert

For Body, type:

Task failure, see Problem Report Activity screen

Click **Submit**. Your Email Template should resemble the sample shown below.

Email Template	
Template Name:	Notification based on status
Email Connection:	Opswise - Gmail Account
Reply-To:	
To:	cannetum@gmail.com
Cc:	
Bcc:	
Subject:	Task failure alert
Body:	Task failure, see Problem Report Activity screen

Creating an Email Task Using an Email Template

- From the navigation pane, select **Tasks > Email Tasks** and click **New**.
- In the Task Name field, type **Triggered by Task Status**.
- 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.
- 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.

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

Task Monitor | Task Virtual Resources | Variables | Actions | Task Instances | Task Monitor Triggers | Notes | Versions

Task Monitor | Required field

Task Name: Monitors for Problems | Member of Groups:

Version: 1

Task Description:

Status:

Monitoring Type: General Task(s)

Task Name To Monitor Condition: All Tasks

Task Type To Monitor:

<input type="checkbox"/> DEFINED	<input type="checkbox"/> WAITING
<input type="checkbox"/> HELD	<input type="checkbox"/> RESOURCE REQUESTED
<input type="checkbox"/> RESOURCE WAIT	<input checked="" type="checkbox"/> UNDELIVERABLE
<input type="checkbox"/> QUEUED	<input type="checkbox"/> SUBMITTED
<input type="checkbox"/> ACTION REQUIRED	<input type="checkbox"/> STARTED
<input type="checkbox"/> RUNNING	<input checked="" type="checkbox"/> RUNNING/PROBLEMS
<input checked="" type="checkbox"/> IN DOUBT	<input checked="" type="checkbox"/> START FAILURE
<input type="checkbox"/> CONFIRMATION REQUIRED	<input checked="" type="checkbox"/> CANCELLED
<input checked="" type="checkbox"/> FAILED	<input type="checkbox"/> SKIPPED
<input type="checkbox"/> FINISHED	<input type="checkbox"/> SUCCESS

Workflow Condition: -- None --

Time Scope: -- None --

Late Start: ☐

Late Finish: ☐

Early Finish: ☐

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.

1. From the navigation pane, select **Triggers > Task Monitor Triggers** and click **New**.
2. In Trigger Name, type **Controls Monitors for Problems**.
3. In the Task Monitor field, browse to the Task Monitor task you just created (Monitors for Problems).
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".
Hint: Click [here](#) for more information about selecting records.

5. Select **Triggered by Task Status**, which you created earlier.
6. Click the down-arrow and select **Save**.
7. Click **Enable Trigger**.

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.

1. From the navigation pane, select **Tasks > Manual Tasks**.
2. Right-click on the Pause for Manual task and select **Launch Task**.
3. Display the Activity screen with the Today's Task Instances by Created Times filter.
4. First note that the Monitor for Problems task is running. It was launched when you enabled the trigger.

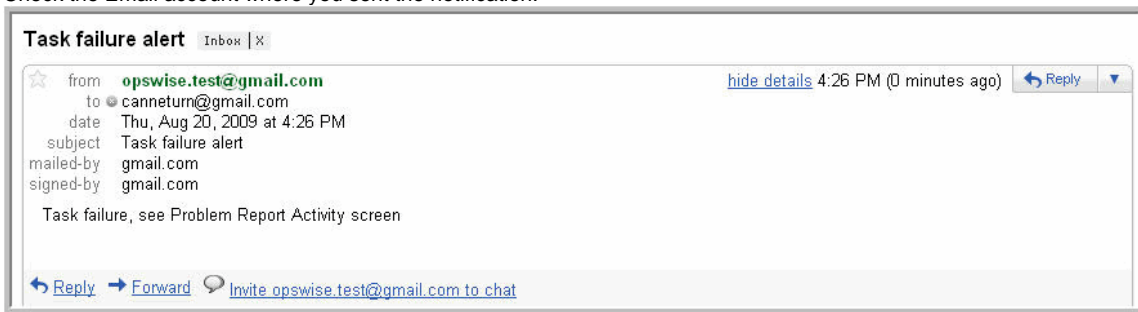
Today's Task Instances by Created Time					New Report	Edit R
Instance Name	Type	Duration	Status	Start Time		
Monitors for Problems	Task Monitor		Running	2009-08-18 18:32:41 -C		

5. Locate the **Pause for Manual** task instance, which will be in Action Required status.
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
Pause for Manual	Manual	Action Required
Triggered by task status	Sleep	Success
Triggered by task status	Sleep	Success
Triggered by task status	Sleep	Success
Triggered by task status	Sleep	Success
Triggered by task status	Sleep	Success
cc.test.ucm#003	z/OS	Finished
ABR daily	Workflow	Running
ABR TIA	Workflow	Running
ABR jobset 1	Workflow	Running
ABR job 2	Windows	Waiting
ABR File Monitor Task	File Monitor	Queued
ABR job 3	Windows	Waiting
ABR job 1	Windows	Skipped

7. Select **Cancel**. Pause for Manual goes to Cancelled status and an Email task is launched by the Task Monitor trigger.
8. Check the Email account where you sent the notification.



9. Once the user receives the email, he or she can quickly check for more information by looking at the Activity screen using the Problem 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 Groups:

Instance Reference Id: 1

Task Description: A Manual task run at 2012-02-08 12:37:06 -0500

Status: CANCELLED

Status Description:
State was cancelled from ACTION REQUIRED to CANCELLED

Start Time: 2012-02-08 12:37:06 -0500 Duration:

End Time: 2012-02-08 12:39:35 -0500

User Estimated End Time: Shortest Estimated End Time:

Average Estimated End Time: Longest Estimated End Time:

Late Finish: [x] Late Finish Type: Duration

Finished Late: [x] Late Finish Duration: 00:02:00 hh:mm:ss

[Update] [Show Details] [Force Finish] [Re-run] [Delete]

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

For additional information, see:

- Email Task (user documentation)
- Email Resource (user documentation)
- Email Template (user documentation)
- Reports (user documentation)

- [Activity Table \(user documentation\)](#)
- [Activity display \(user documentation\)](#)
- [Task Monitor Task \(user documentation\)](#)
- [Task Monitor Trigger\(user documentation\)](#)
- [Command Quick Reference \(user documentation\)](#)
- [Cancelling a Task Run \(user documentation\)](#)
- [Modifying Activity Screen Reports \(2 minute movie\)](#)

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 Windows agent.

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:

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

Agent Name	Host Name	Type	Agent Id	Version	Last Heartbeat	CPU Load	Status	Jobs Run
ADCD - 22	ADCD	z/OS	AGNT0022	\$Name: \$	2009-09-01 13:33:38 -0700	0	✓	3
centos-build.opslab - 4	centos-build.opslab	Linux/Unix	AGNT0004	development	2009-09-01 13:33:18 -0700	1	✓	1457
centos47.opslab - 2	centos47.opslab	Linux/Unix	AGNT0002	development	2009-09-01 13:32:08 -0700	0	✓	0
win2k3-32 - 1	win2k3-32	Windows	AGNT0001	1.4.0	2009-09-01 13:32:08 -0700	0	✓	3

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.

Required field	
Agent Name:	win2k3-32 - 1
Agent Id:	AGNT0001
Host Name:	win2k3-32
IP Address:	192.168.1.63
PID:	2056
Started Date:	2009-08-30 03:09:25 -0700
Status:	Active
Credentials:	
Log Level:	Severe Error
CPU:	3790
Member of Groups:	
Version:	1.4.0
Build Id:	build.3
Build Date:	20090817210719
Operating System:	Microsoft Windows
Operating System Release:	Server 2003 family, Standard Edition Service Pack
Heartbeat Interval:	120
Last Heartbeat:	2009-09-01 13:34:08 -0700
CPU Load:	0

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

1. Select **Tasks > Windows Tasks** and click **New**.
2. In Task Name, type **Making Dir "Tutorial"**.
3. In the Agent field, select your Windows agent.
4. In the Command field, type:

```
md c:\tutorial
```

5. Click **Submit**.
6. From the Agent list, right-click on **Making Dir "Tutorial"** and select **Launch Task**.
7. Check the Activity screen for the task instance.

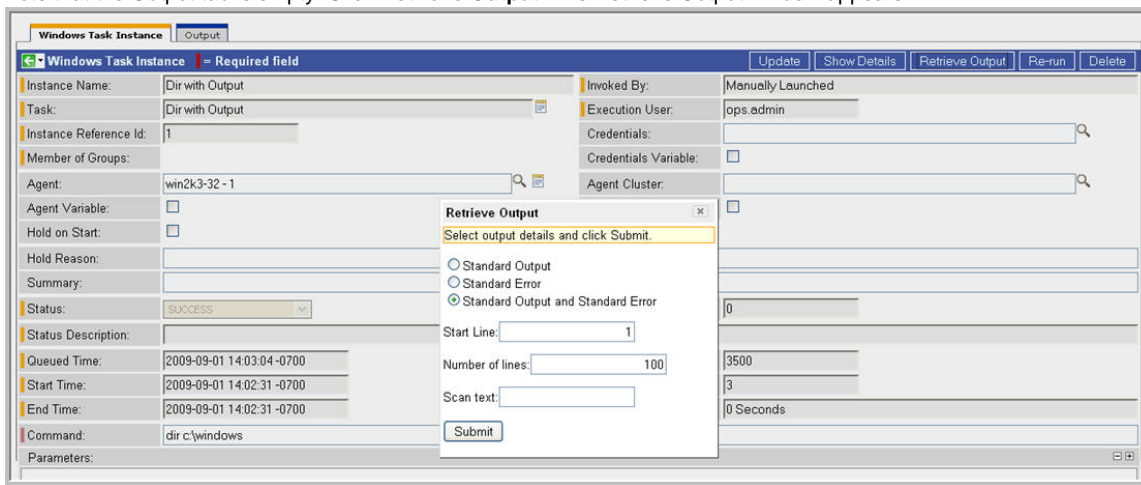
Manually Retrieving Output from a Windows Task

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.

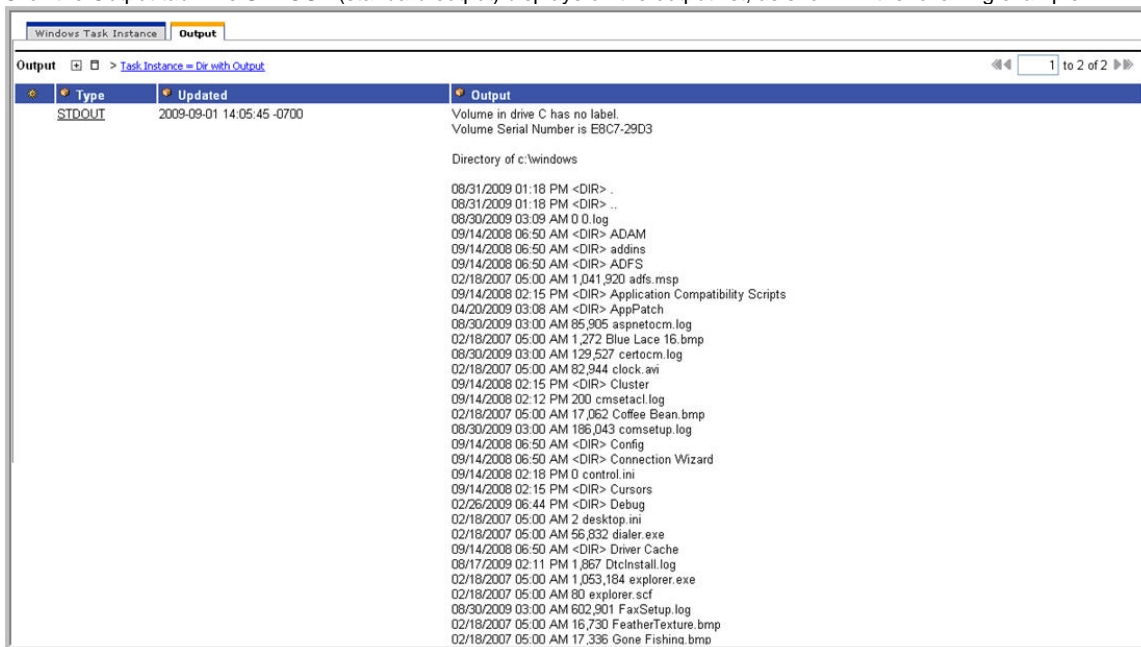
1. Use the same steps to create another Windows task called **Dir with Output**. Do not specify Automatic Output Retrieval, and use the following command:

```
dir c:\windows
```

2. Save and launch the task.
3. View the task on the Activity screen, and click on the name.
4. Note that the Output tab is empty. Click **Retrieve Output**. The Retrieve Output window appears:



5. Change the number of lines to 300 and click **Submit**.
6. Click the Output tab. The STDOUT (standard output) displays on the output list, as shown in the following example:

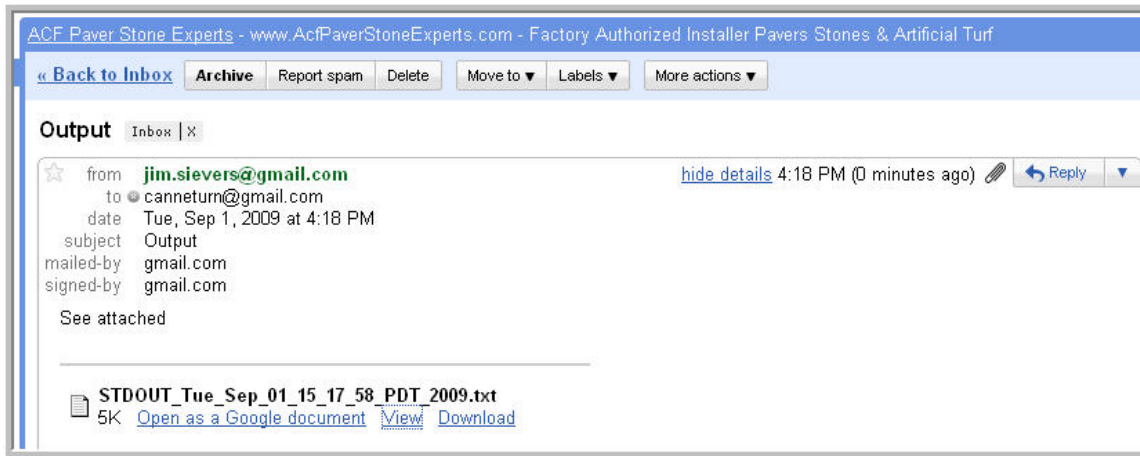


You can also click the STDOUT to view the record.

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.

1. Open the **Dir with Output** task you just created.
2. Click the Actions tab and click **New**.
3. Select **Email Notification**.
4. Specify the following:
 - Status=Success
 - Email Connection=Your email connection
 - To=Your email address
 - Subject=Output
 - Body=See attached.
 - Attach Standard Output=enabled
 - Start Line=1
 - Number of lines=300
5. Click **Submit**.
6. Launch the task.
7. Once the Windows task goes to success, check your inbox for the email. The output is attached, as shown in the following example.



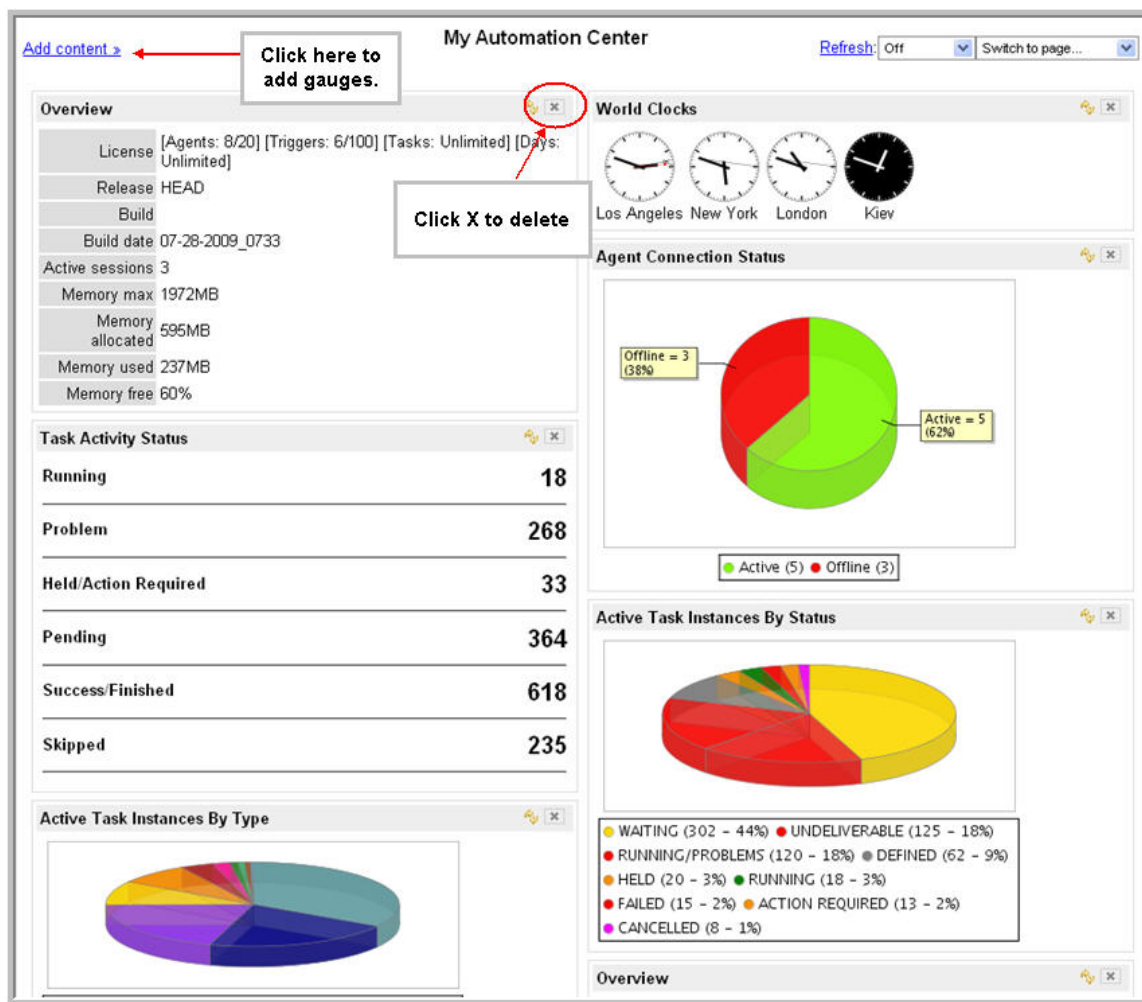
Tutorial - Selecting Gauges for the Home Page

- Introduction
- Deleting Gauges
- Adding Gauges
- Setting the Refresh Rate

Introduction

The Homepage is associated with a login ID. It is the first page you see when you log in to Opswise.

Shown below is a sample home page. Each window showing on the page contains a different piece of information, referred to as a gauge. For this exercise, we are going to delete all the windows and add a new set. Your home page may differ from the sample shown.



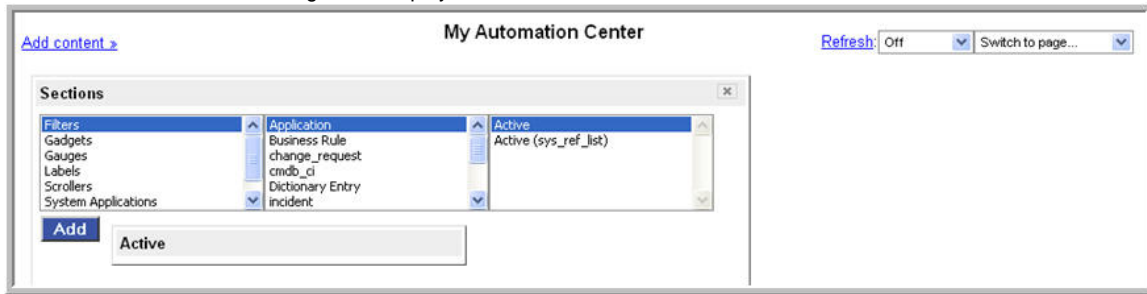
Deleting Gauges

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

Adding Gauges

Now we are going to add the gauges shown in the sample.

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



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

2. To add the Overview gauge, select **Gadgets > System Information > Overview** and click **Add**. The Overview panel appears on the home page.
3. To add the World Clocks, select **World Clocks > World Clocks** and click **Add**.
4. To add Task Activity Status, click **Gauges > Activity > Task Instances by Status** and click **Add**.
5. To add Active Task Instances By Type, click **Gauges > Activity > Active Task Instances By Type** and click **Add**.
6. To add Agent Connection Status, click **Gauges > Agent > Agent Connection Status** and click **Add**.
7. We are finished adding windows, so click the **X** in the upper corner of the **Add Content > Sections** menu.
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**.

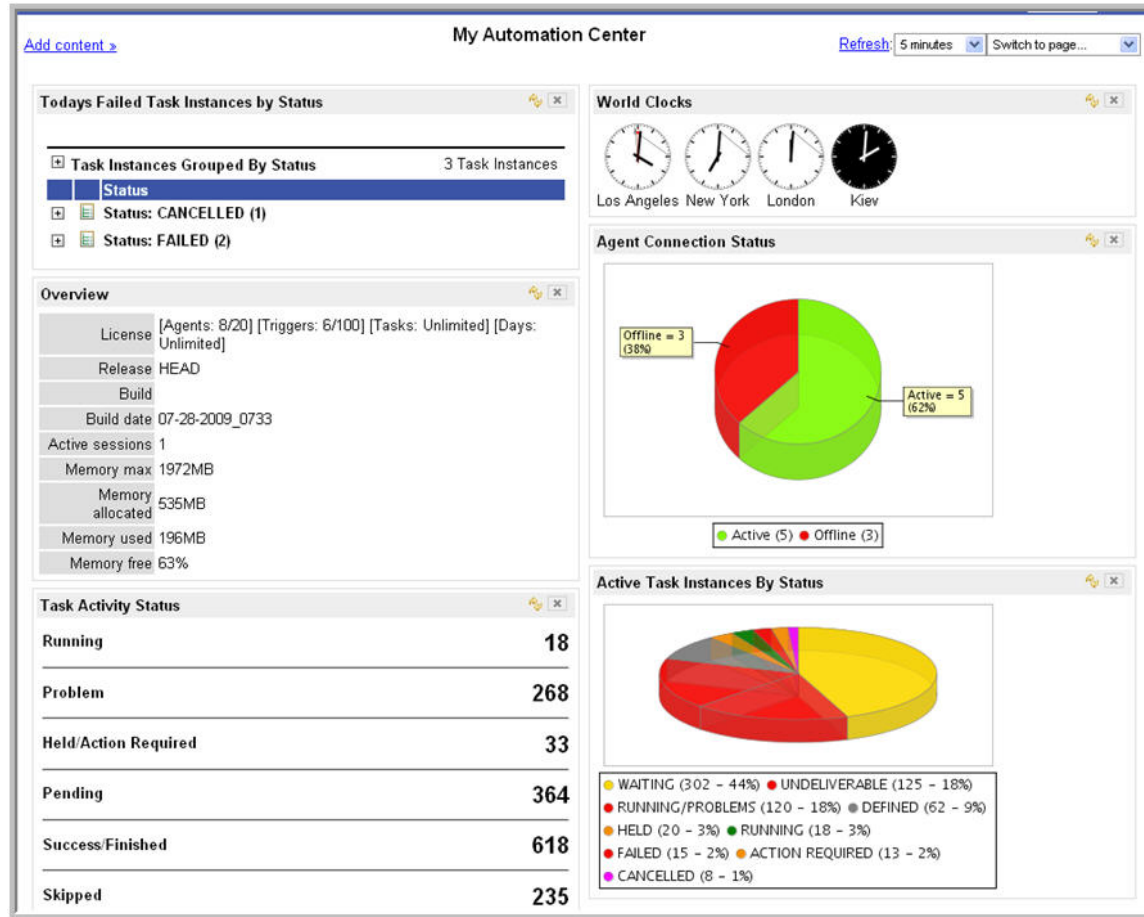
For additional information, see:

- [Home Page, Widgets, Dashboard \(3 minute movie\)](#)
- [Customizing Your Home Page](#)

Tutorial - Making a Gauge

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

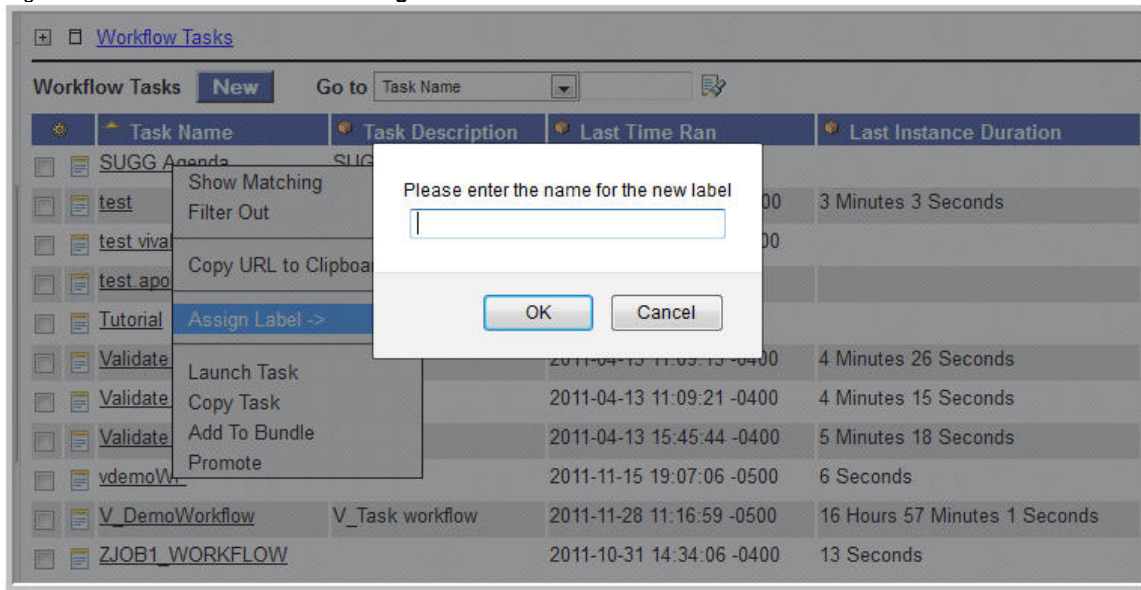
1. Select **Automation Center > Reports**.
2. Scroll down to the Activity section, locate **Today's Failed Task Instances by Status**, and click it to open it.
3. Click **Make Gauge**. Note that Type has reverted back to List. A gauge has been created from the report but the report remains as it was.
4. Click the home  icon (upper right) to return to the home page.
5. Click **Add content**.
6. Select **Gauges > Activity > Today's Failed Task Instances by Status** and click **Add**. The gauge is added to the home page.



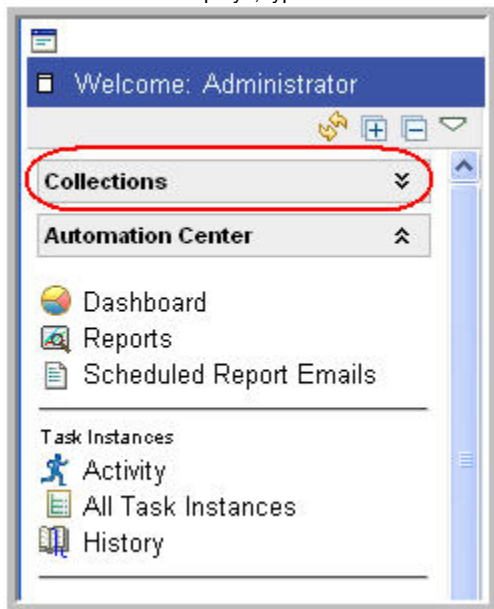
Tutorial - Adding Options to the Navigation Pane

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.

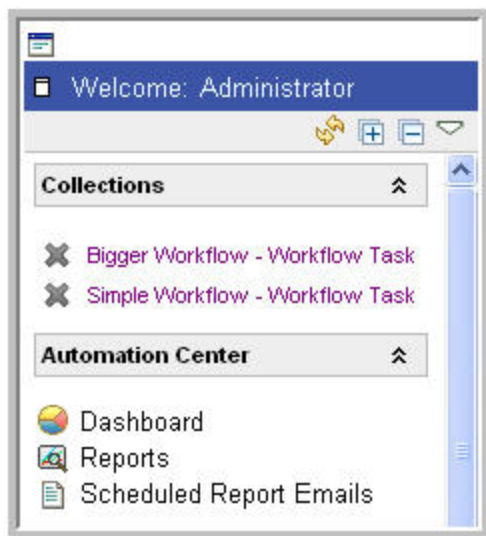
1. Display the Tasks list and locate the workflow you created in an earlier exercise called **Simple Workflow**.
2. Right-click the task name and select **Assign Label > New**.



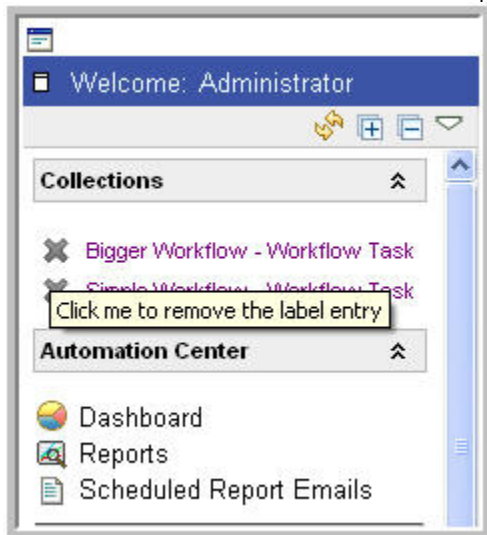
3. When the window displays, type **Collections** and click **OK**. The new label has been added to the navigation pane.



4. Locate the other workflow you created called **Bigger Workflow**.
5. Right-click the name and select **Assign Label > Collections**.
6. In the navigation pane, click the down arrows next to the Collections entry.



7. Click one of the entries under Collections. The system displays the record.
8. Now we will remove both of the records, along with the new menu option.
 - a. Click the **X** to the left of each of the records. It disappears from the list.



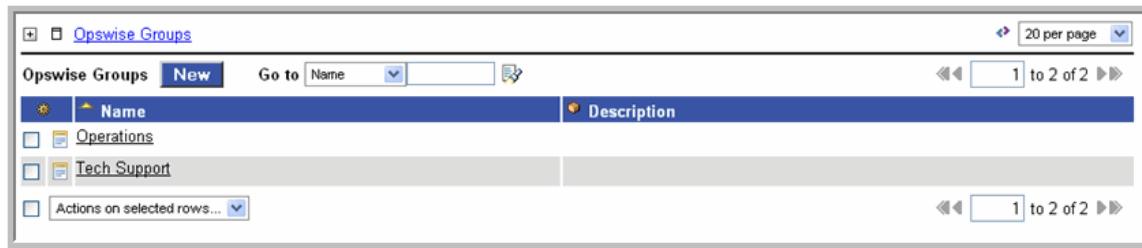
- b. Click the name, **Collections**. The label record appears.

- c. Click the **Delete** button. The label disappears from the navigation pane.

Tutorial - Opswise Groups

In this exercise, we will create two hypothetical Opswise groups: Tech Support and Operations. **Hint:** Do not confuse Opswise groups with Security Groups. Opswise groups are used to group records into business functions; Security groups are used to assign permissions to groups of users.

1. Scroll down the navigation pane and select **Opswise Groups**.
2. Click **New**.
3. In the Name field, type **Tech Support**.
4. Click **Submit**.
5. Repeat the above steps for a group called **Operations**.





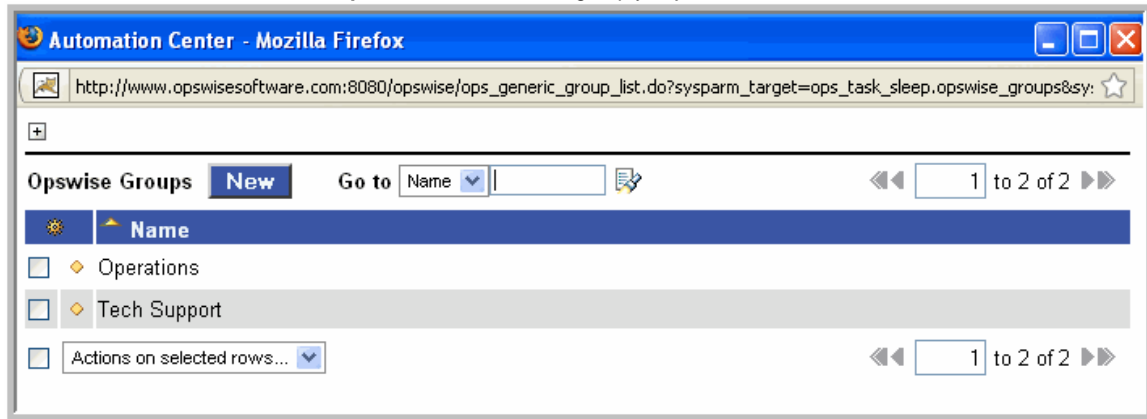
For additional information, see:

- [Opswise Groups](#)

Tutorial - Assigning Records to Opswise Groups

In this exercise, we are going to assign our Sleep tasks and the Simple Workflow task to the Operations group and the SQL tasks and Bigger Workflow task to the Tech Support group.

1. Open the task called Sleep1, which you created for an earlier exercise.
2. Click the lock  icon next to **Member of Groups**.
3. Click the search icon  and click **Operations**, which is the group you just added.



4. Click **Update** to save your changes.
5. Repeat the above steps for the other Sleep tasks you created earlier, Sleep2 up to Sleep6, along with the Workflow we put them in, [Simple Workflow](#).
6. Open the SQL Create Table task you created for an earlier exercise.
7. Click the Member of Groups lock, assign the task to the Tech Support group, and click **Update**.
8. Repeat the previous step for the other three SQL tasks you created earlier, along with the workflow we put them in, [Bigger Workflow](#).

For additional information, see:

- [Opswise Groups](#)



Tutorial - Taking Advantage of Opswise Groups

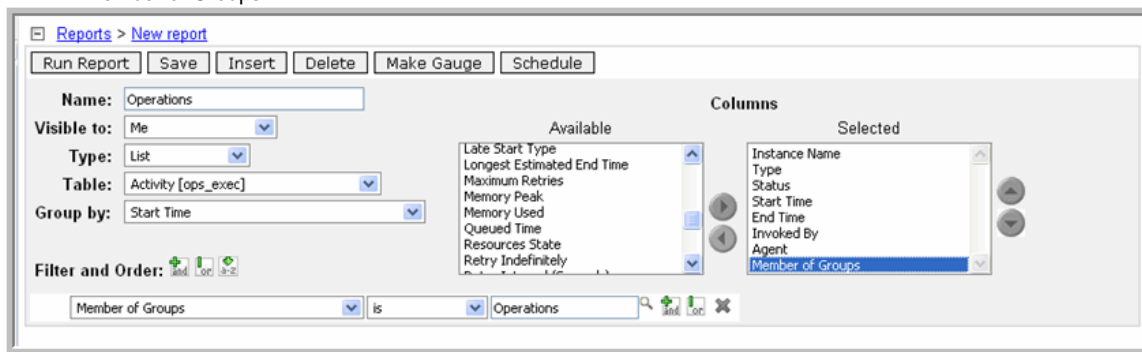
The advantage to Opswise groups is demonstrated in the following additional tutorials:

- [Create a Report Based on Opswise Groups](#)
- [Viewing Activity by Group](#)
- [Creating Users and Assigning Permissions](#) based on Opswise Groups

Tutorial - Create a Report Based on Opswise Groups

In this exercise, we will create two Activity reports so that users from our hypothetical Operations and Tech Support departments can view activity related to their organizations.

1. Select **Automation Center > Reports** and click **New**.
2. In the Name field, type **Operations**.
3. For now, leave the Visible to field with the default value.
4. For type, select **List**.
5. For Table, leave the default Activity(ops_exec), which is the table that contains all Task Instances.
6. In Group by, select **Start Time**.
7. In Filter and Order, click the plus sign. The choose field options appear.
8. Select:
 - Member of Groups
 - is
 - Operations (Hint : Type an **O** into the field and "Operations" pops up.)
9. Modify the Available and Selected Columns to display the following fields. Hint: Use  and  to reorder the columns.
 - Instance Name
 - Type
 - Status
 - Start Time
 - End Time
 - Invoked by
 - Agent
 - Member of Groups



Reports > New report

Run Report Save Insert Delete Make Gauge Schedule




Name: Operations

Visible to: Me

Type: List

Table: Activity [ops_exec]

Group by: Start Time

Filter and Order:   

Member of Groups is Operations

Columns

Available

- Late Start Type
- Longest Estimated End Time
- Maximum Retries
- Memory Peak
- Memory Used
- Queued Time
- Resources State
- Retry Indefinitely

Selected

- Instance Name
- Type
- Status
- Start Time
- End Time
- Invoked By
- Agent
- Member of Groups

10. Click **Save**. 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."
11. Repeat the above sets for a report called Tech Support, filtered by "Member of Groups is Tech Support". Hint: Keep the Operations report displayed, change the name and filter specifications, then click **Insert**.

For additional information, see:

- [Reports](#)

Tutorial - Viewing Activity by Group

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

Introduction

In this exercise, we will open two new browser tabs, drag the Activity function into each one of them, and apply each of our two new Activity reports. First, we will launch two of our workflows from previous exercises so we have some activity to view.

Create Some Activity

If you do not have any of the activity available from our previous exercises, create some by launching the following two workflows:

- Simple Workflow
- Bigger Workflow

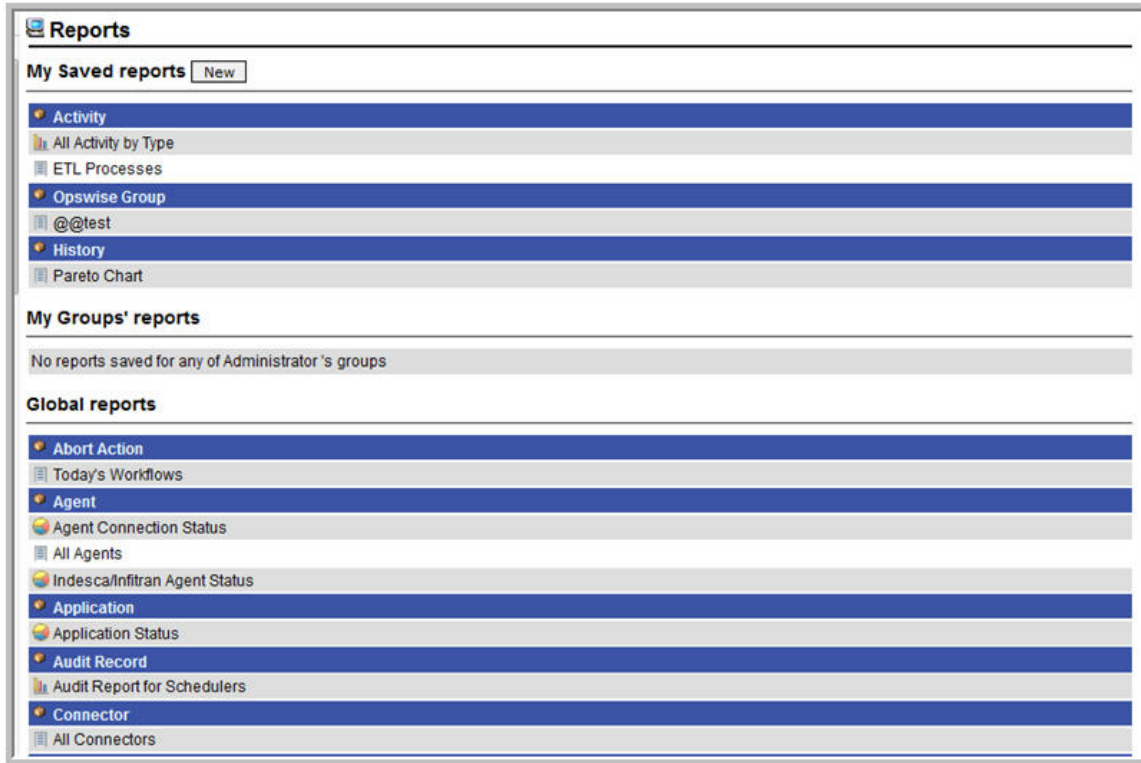
Create Activity Screens Based on Groups

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

Tutorial - Creating a Report

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

1. From the Navigation pane, select **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.

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

3. In the Name field, enter "Tasks Sorted by Resource".
4. In the **Visible to** field, select **Me**. This identifies who will have access to the report as follows:
 - Me — Only you can view/run the report.
 - Everyone — All users can view/run the report.
 - All other options — The remainder of the dropdown list consists of the groups that have been entered by selecting **Security > Groups** from the Navigation pane.
5. In the **Type** field, select **List**.
6. In the **Table** field, select the [Opwise report table|Reports#Opwise Report Tables] you want to use. For this tutorial, select **Activity** (ops_exec), which is the table used for all Activity reports and contains information about all task activity.
7. In **Group by**, select **Agent**. This field specifies that the report should be displayed in sections by agent.
8. 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:
 - 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.
9. To run the report, click **Run Report**. Opwise creates a list of Task Types, as specified in the report.
10. To expand all sections of the report, click the plus sign.
11. Now you will filter the report to select only tasks that ran today:
 - a. Under Filter and Order, click on the box with a plus sign and the word "and".

- b. You are prompted to select filtering criteria.
 - c. For **choose field**, select Created.
 - d. Select 'on'.
 - e. Keep the default 'Today'.
 - f. Click **Run Report** again. The report is re-run with the filter applied, removing those task instances that ran before today.
 12. To save the report, click **Save**.
 13. The report is now listed in your section (My Saved Reports) of the Reports list. To view it, click **Reports** from the Navigation Pane. It has also been added to the list of Activity reports on the Activity screen.
If you want to print the report, save it first and then print it (see [Viewing, Running, Printing an Existing Report](#)).
-

For additional information, see:

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

Tutorial - Scheduling and Distributing Reports

Before you can distribute reports, [configure the email server](#) used for this purpose.

1. Open the Operations report you just created.
2. Click **Schedule**.
3. Complete the fields as follows:
 - Name = Operations
 - Report = Operations
 - Every = Day
 - Time = (specify a few minutes from now)
 - Email addresses = (your email address)
 - Subject = Daily Operations Report
 - Message = Attached please find the Operations Activity Report for \${_date}.

The screenshot shows the 'Scheduled Email of Report' configuration window. It includes fields for Name, Report, Users, Groups, Email addresses, Subject, and Time. The 'Introductory Message' field contains the text 'Attached please find the Operations Activity Report for \${_date}.' The window also features a rich text editor toolbar and a 'Path' field set to 'body > span'. At the bottom, there are buttons for 'Update', 'Execute Now', and 'Delete'.

Scheduled Email of Report		Update	Execute Now	Delete	↑	↓
Name:	Operations	Active:	<input checked="" type="checkbox"/>			
Report:	Operations	Every:	Day			
Users:		Time:	Hours 18 : 00 : 00			
Groups:						
Email addresses:	cannetum@gmail.com					
Subject:	Daily Operations Report					
Introductory Message:						
<div> <div>Arial</div> <div>1 (8 pt)</div> <div>Heading 1</div> <div>B</div> <div><i>I</i></div> <div><u>U</u></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> </div> <div>Attached please find the Operations Activity Report for \${_date}.</div>						
Path: body > span						
Type:	PDF-landscape					
Include detail:	<input type="checkbox"/>					
Update		Execute Now		Delete		

4. Click **Submit**.
5. At the time you indicated, check your inbox for a message containing the report in PDF format.

For additional information, see:

- [Reports](#)

Tutorial - Setting Up a Virtual Resource

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

1. Select **Resources > Virtual** and click **New**.
2. Name the Virtual Resource "Sofa" and change the Resource Limit to 5 and **Submit**.
3. Set up the Sleep tasks as follows:
 - a. Add the following Sleep task:
 - Task Name=BigGuy1
 - Sleep Time=60
 - b. Save the task and click the **Task Virtual Resources** tab.
 - c. Click **Edit**.
 - d. Select Sofa.
 - e. Sofa is added to the list with a default resource amount of 1. Click on the 1 to open the record.

- f. Change the Amount to 4, which means that this task will consume most of the resource.
 - g. 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.
 - h. 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.
4. Display the Sofa Virtual Resource and click the **Task Virtual Resources** tab. All tasks assigned to this virtual resource are listed, along with the resource amount that each one uses.
5. We want all three of our tasks to launch at once, so use a simple [time trigger](#) to launch all three in a minute or two. Hint: Don't forget to enable the trigger.
6. 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.

Today's Task Instances by Created Time						New Report	Edit Report
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					

7. On the Virtual Resource Sofa, click the **Currently In Use By** tab and observe which tasks are running on this virtual resource. Note that this display does not automatically refresh.

For additional information, see:

[Creating Virtual Resources](#)

Tutorial - Getting the Most from the Activity Screen

1. Navigate to the Activity screen.
2. Click on the Instance Name of any task. Note that the bottom of the screen shows additional details that are not there when you display the original task record. These details provide information about the run of this task instance, along with information about 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 displays the 'SQL Task Instance' details page. At the top, there are tabs for 'SQL Task Instance', 'SQL Result Set', and 'SQL Warning Set'. Below the tabs, there are buttons for 'Update', 'Show Details', 'Re-run', and 'Delete'. The main form is divided into several sections:

- Task Information:** Instance Name (Payroll DB Query), Task (Payroll DB Query), Instance Reference Id (1), Member of Groups (empty), Database Connection (opswisedb), Credentials (root/sbs), Credentials Variable (checkbox), Maximum Rows (empty), Task Description (empty).
- Status:** Status (SUCCESS), Exit Code (0), Status Description (empty).
- Timing:** Start Time (2012-02-13 16:00:48 -0500), End Time (2012-02-13 16:00:49 -0500), Duration (1 Seconds), Rows Retrieved (100).
- SQL Details:** SQL State (empty), SQL Error Message (empty), SQL Command (SELECT * FROM ops_task_windows;).
- Configuration:** Result Processing (Success Exitcode Range), Exit Codes (0), Auto Cleanup (checkbox checked), Maximum Retries (0), Retry Interval (Seconds) (60), Current Retry Count (0), Retry Indefinitely (checkbox unchecked).
- Estimated End Times:** User Estimated End Time (empty), Shortest Estimated End Time (empty), Average Estimated End Time (empty), Longest Estimated End Time (empty).

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

In [Create a Report Based on Opwise Groups](#), you learned how to use the Reports function to create a new filter for the Activity display. Remember that all reports created using the [Opwise Activity table](#) table (ops_exec) will appear in the drop-down menu on the Activity display. You can view all task instances details stored in the Activity table as follows:

1. From the Activity screen, click on the Instance Name of any task.
2. Click the **Show Details** button. A new browser tab opens, displaying all details from the task. The column on the left shows each field name from the [Opwise Activity table](#) table (ops_exec). The column on the right shows the current value in each field for the task instance you are displaying.

Task Instance SQL Create Table

attempt_count	1
auto_cleanup	true
avg_estimated_end	2009-08-31 20:00:30
calendar	77171434c0a801c9016d5b2b5d17ddee
column_op	EQUAL
connection_id	613c1b12c0a801ca01dc0b999e2795b9
cpu_time	0
credentials	6b84c28ad861e5e400fe9ac400ca87b8
credentials_var_check	false
duration	431
duration_seconds	0
early_finish	false
ef_enabled	false
ef_time	00:00
ef_type	TIME
end_time	2009-08-31 20:00:30
execution_user	ops.admin
exit_code	0
forced	false
high_estimated_end	2009-08-31 20:00:30
invoked_by	Workflow: Bigger Workflow
io_other	0
io_reads	0
io_writes	0
late_finish	false
late_start	false
lf_enabled	false
lf_time	00:00
lf_type	TIME
low_estimated_end	2009-08-31 20:00:30
ls_enabled	false
ls_time	00:00

ls_type	TIME
memory_peak	0
memory_used	0
name	SQL Create Table
opswise_groups	61f627d5d861e5e401c2092d7fa11d9a
priority	MEDIUM
res_state	Initial
result_processing	SKIP
retry_counter	0
retry_indefinitely	false
retry_interval	60
retry_maximum	0
run_called	true
run_criteria_rt	false
run_criteria_tt	false
security_name	SQL Create Table
sql_command	CREATE TABLE opswise_tut20090905 (name varchar(128), value varchar(128));
sql_rows	1
start_held	false
start_time	2009-08-31 20:00:30
status_code	SUCCESS
sys_class_name	ops_exec_sql
sys_created_by	ops.admin
sys_created_on	2009-08-31 20:00:05
sys_id	720afc8cd861e5e401c1a84875066ccb
sys_mod_count	4
sys_updated_by	ops.admin
sys_updated_on	2009-08-31 20:00:30
task_id	5cd9d11ed861e5e401e8de6203ec9f3e
task_ref_count	12
type	SQL
vertex_id	3
workflow_id	720afc6ed861e5e4017362078c8de40a

Tutorial - Aborting a Process Launched by a Task

You can use the Abort Actions tab to instruct the system 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 if the task runs longer than 45 seconds.

1. Create a new Sleep task that will run for 60 seconds. Enable Late Finish, specify a Late Finish Type of Duration, and enter 45 seconds.
2. This means that the task will be considered late finishing if it takes longer than 45 seconds.
3. Save the task and click the **Actions** tab.
4. Click **New** and select **Abort Action**.
5. Enable On Late Finish and click **Submit**.

5. Launch the Sleep task.
6. Navigate to the Activity screen. After 45 seconds, the task goes to Finished status.
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 we created in the last exercise. We will also assign access and management rights via user roles and Opwise permissions:

- Permissions control who can add, change, delete and control Opwise 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. See [Description of Roles](#) for details.

Create New Users

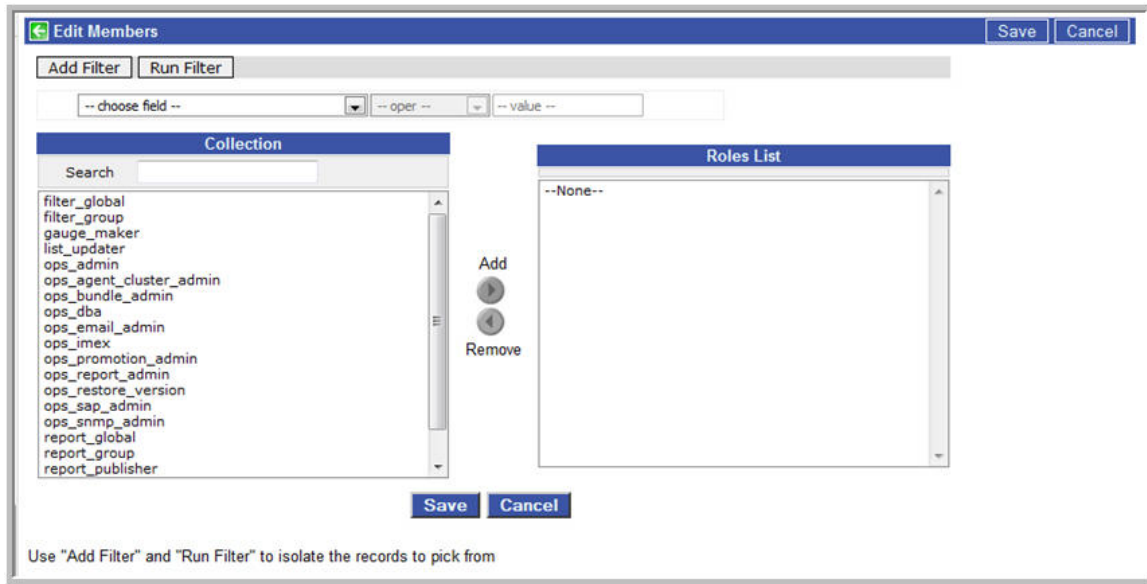
In this exercise, we will create three new users.

1. From the navigation pane, select **Security > Users**. You will see the default user, ops.admin.
2. Click **New**.
3. In the Name field, type OpMan (short for Operations Manager).
4. Fill in the remaining fields as follows:
 - First name=Ben
 - Lastname=Bernanke
 - Password=abc
5. Click **Submit**.
6. Add two more users:
 - TSMan, name=Tim Geithner, password=abc
 - OpClerk, name=Jimmy Smith, password=abc
7. To test one of the new users:
 - a. Click the **Logout** button in the upper right corner of the display.
 - b. Log back in using the OpMan/abc ID and password.
 - c. Click on several options in the navigation pane. Since we have not assigned any permissions to OpMan, this user is prohibited from the following:
 - Viewing tasks, tasks instances, or triggers
 - Changing or deleting the dashboard or home page items, email or database connections, calendars or custom days, Opwise groups.
 - The Security and administrative options do not display on the navigation pane.
 - d. When you are finished, click **Logout** and log back in using ops.admin.

Assign Permissions Via User Roles

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

1. Open the OpMan user and click the **User Roles** tab.
2. Click **Edit**.
3. This user is a manager so we will give him the [admin roles](#). Ctrl-Click ops_admin and ops_report_admin, and click the right-arrow .



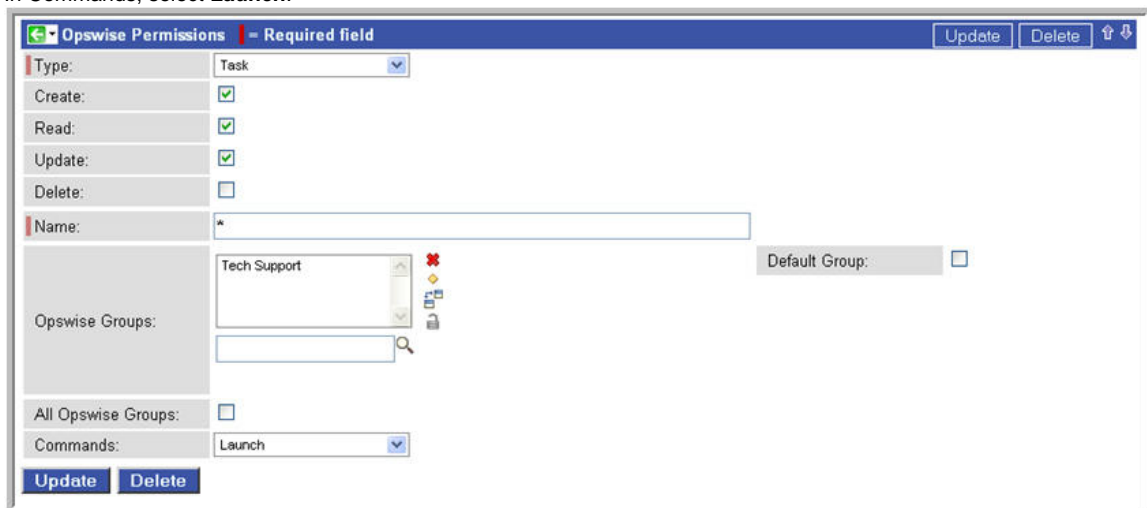
Click **Save**.

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 that we added in the previous exercise [Creating Opwise Groups](#).

1. Open the User, TSMAN.
2. Click the **Opwise Permissions** tab and click **New**.
3. First we will give permission to view, add, update, and manually launch tasks.
4. In Type, select **Task**.
5. Select Create, Read, and Update.
6. In Opwise Groups, select **Tech Support**.
7. Deselect Default Group. Hint: See [Permissions Field Descriptions](#) for more details.
8. In Commands, select **Launch**.



9. Click **Submit**.
10. 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, he cannot see them on the Activity screen because he was not given permissions on Task Instances.

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.

1. Open the user record OpClerk.
2. Click the **Group Members** tab.
3. Click **New** to add a new group.
4. Name the group Operations Clerks.
5. Right-click the down arrow and click **Save**.
6. While still displaying the Operations Clerks group, click the **Opswise Permissions** tab and click **New**.
7. **Assign the following permissions:**
 - Type = Task Instance
 - Read, Update, Delete = Enabled
 - Opswise Groups = Operations
 - Default Group = Disabled
 - Commands = All

These permissions give the Operations Clerks full permissions on all activity (task instances) related to the Operations Opswise group. In future, any users you assign to this user group will inherit these permissions.

8. Click **Submit**.
9. **Log out and log back in as OpClerk and check the permissions.**

For additional information, see:

- [Security](#)

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.

- Create the following three users:
 - AmyH/Amy Hempel/abc
 - TobyW/Tobias Wolff/abc
 - RayC/Ray Carver/abc
- Select **Users > Groups** and add a group called **Data Center** and save it.
 - Click the **Opswise Permissions** tab and add the following permission:
 - Type=Task Instance
 - Read
 - All Opswise Groups
 - Click the **Group Members** tab and click **Edit**.
 - Add Ray Carver to the group.
- Add another group called **Banking Operations** and save it.
 - Add the following two permissions:
 - Type=Task Instance
 - Read
 - Update
 - Opswise Groups=Collections
 - Default Group=deselect
 - Commands=None
 and
 - Type=Task Instance
 - Read
 - Update
 - Opswise Groups=Letters of Credit
 - Default Group=deselect
 - Commands=All

Type	Operations	Commands	Name	Opswise Groups	Default Group
Task Instance	Read, Update	*	*	Collections	false
Task Instance	Read, Update	ALL	*	Letters of Credit	false

- Use the Group Members tab and **Edit** button to add Amy Hempel to the group.
- Add a third group called **IT Personnel** and save it.
 - Add the following two permissions: Type=Trigger
 - Create
 - Read
 - Update
 - Delete
 - All Opswise Groups
 - Commands=None
 - Repeat the above permissions for Type=Task.

Type	Operations	Commands	Name	Opswise Groups	Default Group
Trigger	Create, Read, Update, Delete	*	*	*	true
Task	Create, Read, Update, Delete	*	*	*	true

- Add Tobias Wolff to the group.
- 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:

- [Security](#)

