Opswise Automation Center 5.1.0

User Guide

© 2012 by Stonebranch, Inc. All Rights Reserved.

Getting Started

This page provides you with a quick tour through the Opswise features and includes links to the appropriate documentation on this website.

Installation

If you are a new Opswise customer, you'll first want to consult the Installation Steps for help in getting the system installed and verified.

Getting Started

Once the system is installed, we recommend that you log in and familiarize yourself with the basic user interface features. Within Opswise, you will create all your records by entering information into forms. The records are then displayed in lists of each type, which you can sort and filter and perform a wide variety of other functions.

Tutorials

Step-by-step tutorials for many of the features described here are available on the First Look page.

Security

One of your first tasks will be to use the Opswise Security module to create users and user groups and assign them roles and permissions.

Resources

Next, you'll want to define what types of resources you'll need in your operational database. If you have installed Opswise agents on any machines, records for each of them are automatically created when they connect to the Opswise core processor. You may also need to define one or more agentless resources, such as email, database, or SNMP connections. You can also create status-based notifications for these resources. You can set up a throttling scheme for your machines using Virtual Resources and create a library of scripts that you can execute on remote machines.

Tasks

Once you have your resources in place, you can begin creating tasks. Supported task types are Workflows, Linux/Unix, Windows, z/OS, Indesca, File Transfer, Manual, Sleep, SQL, Stored Procedures, Email, Task Monitors, File Monitors, FTP File Monitors, System Monitors, SAP, and Application Control Tasks.

Triggers

To run your tasks outside of workflows, you can create task triggers, which define events, conditions, or dates/times that the tasks will run. Trigger types include Cron, Time, Temporary, Manual, File Monitor, Task Monitor, and Application Monitor. You may also need to set up one or more customized calendars that reflect your fiscal year and holiday schedules.

Monitoring Operations

You will monitor your automated operations using the Activity Display, which you can customize using filters.

Manually Running and Controlling Tasks

You may also need to manually run and control tasks, either from the user interface or from the command line.

Home Page, Dashboard, and Reporting

You can also customize the home page and Dashboard of your system for each user and create, generate, and distribute custom reports. You can also define your own set of gauges.

Monitoring and Controlling Applications

Opswise Automation Center also includes a module that allows you to monitor and control all the applications running in your entire network.

Managing Data and Audit Records

Comprehensive utilities are provided that allow you to manage your Opswise records. A complete audit history of all system activity is also available for regulatory compliance.

Command Line Interface

You can also monitor and control your operations and perform basic administrative functions from the command line.

Navigation Pane Reference

You may need to look up a description of each item on the Opswise navigation pane.

Logging In

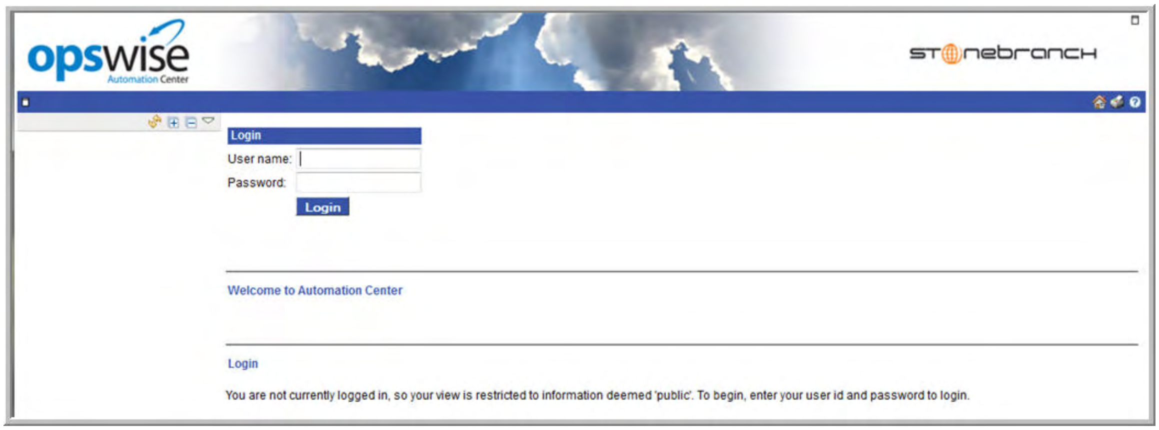
The Opswise login screen displays automatically when you bring up the system and browse to its URL. The default URL is

```
machinename:8080/opswise
```

For example, if you installed the system locally, the URL is:

```
localhost:8080/opswise
```

The login screen is shown below:



Welcome to Automation Center

Login

You are not currently logged in, so your view is restricted to information deemed 'public'. To begin, enter your user id and password to login.

The default login ID is:

```
ops.admin
```

No password is required for the initial login. When you log in for the first time on a new installation, the system prompts you to create a password.

See Security for information about setting up your own users and security restrictions.

Getting Help

Several types of help are available to Opswise Automation Center users:

- To access the Opswise user documentation on this website, click ? (the help button) in the upper right corner of any Opswise Automation Center screen. For information about this website, see HELP.
- To watch a quick video on how to perform a specific task, click **Automation Center > Support Links > Video Classroom** from the navigation pane and select the video you want to view. Additional videos for new features are available here.
- To enter a help desk ticket, select **Automation Center > Support Links > Support Portal** from the navigation pane. This page provides support phone numbers and a form where you can describe and submit your problem.

Documentation PDFs

Introduction

This page provides a choice of pre-created documents in PDF format, which you can download and save.

You can also create your own PDF documents as follows:

- Using the "Advanced" method, you can select which wiki pages you want included in your PDF document
- Use **Export to PDF** from the **Tools** menu to create a PDF document from the current wiki page.

Download Pre-Created PDFs

Stonebranch provides all Opswise Automation Center documentation in PDF format:

- Opswise Installation and Upgrade
- Opswise User Guide
- Opswise Security
- Command Line Interface Reference
- RESTful Web Services API Reference
- Browser Interface Navigation Pane Reference
- Maintenance Scripts
- Tutorials

For links to all Workload Automation 5 PDFs, click here.

Creating a PDF of Selected Pages

To create a PDF of selected Opswise Automation Center pages - which will include a front cover, bookmark, and page headers and footers - perform the following steps:

1. From the **Browse** drop-down list, click **Advanced**.
2. From the **Export** menu, click **PDF Export**.
3. Select any or all of the pages to be included in the PDF.

> ✓ **Tip**
> Deselect **Inclusions Library** and **Home** and manually select each section you want to include.

4. Click **Export**.
5. When the PDF export is complete, click the **Download here** link.

> ⚠ **Note**
> Your PDF may contain instances of the following message: **Error formatting macro: redirect: java.lang.NullPointerException**. This is not an error in the PDF; it is internally generated code that identifies where page redirection has been implemented in the online documentation in order to automatically expand the Navigation panel when a page has been selected.

Creating a PDF of the Current Page

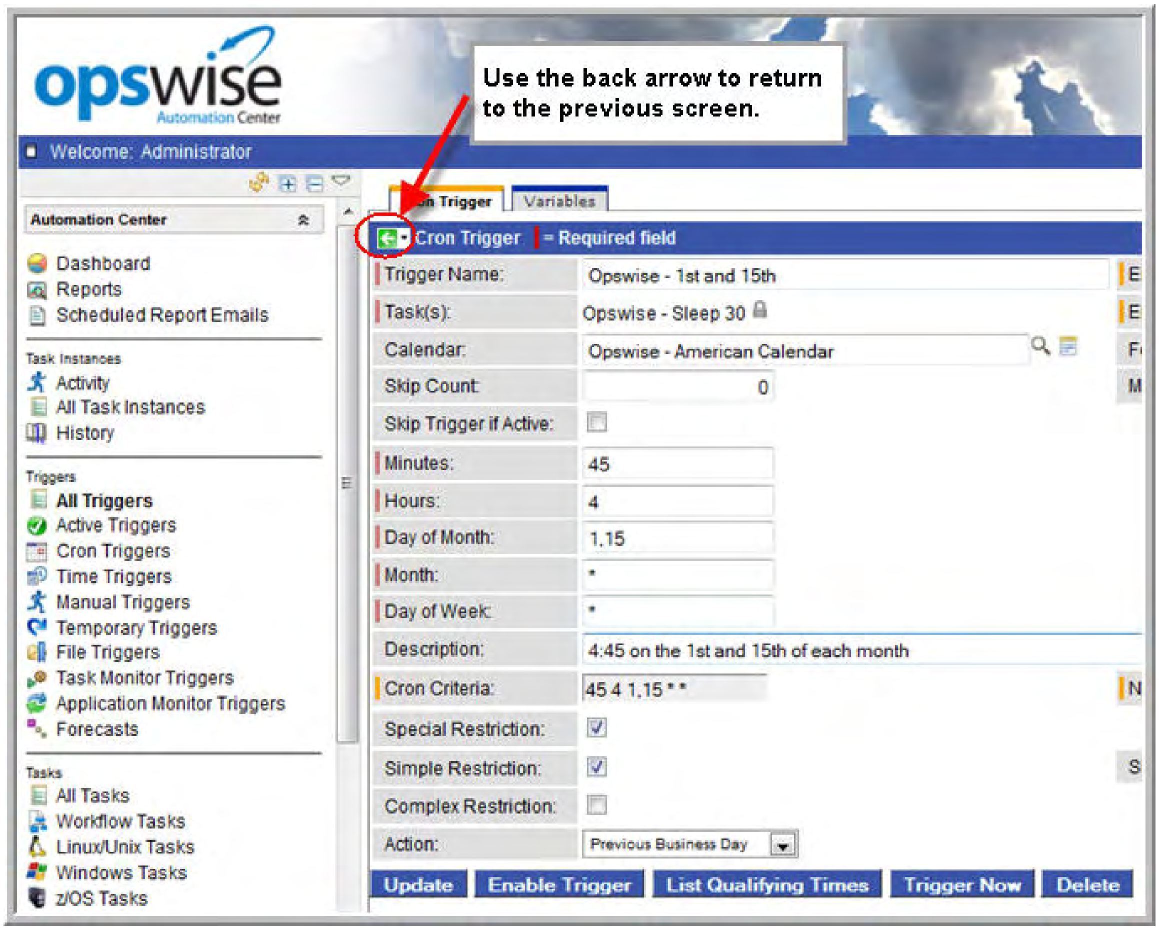
To create a PDF of the current page - without a front cover, bookmark, and page headers and footers - perform the following steps:

1. From the **Tools** drop-down list, click **Export to PDF**.
2. Open or Save the PDF.

User Interface

Back Arrow

The back arrow, shown below, allows you to back up to the previously displayed screen. If you make changes to a form and then click the back arrow, the changes are discarded.



Use the back arrow to return to the previous screen.

Welcome: Administrator

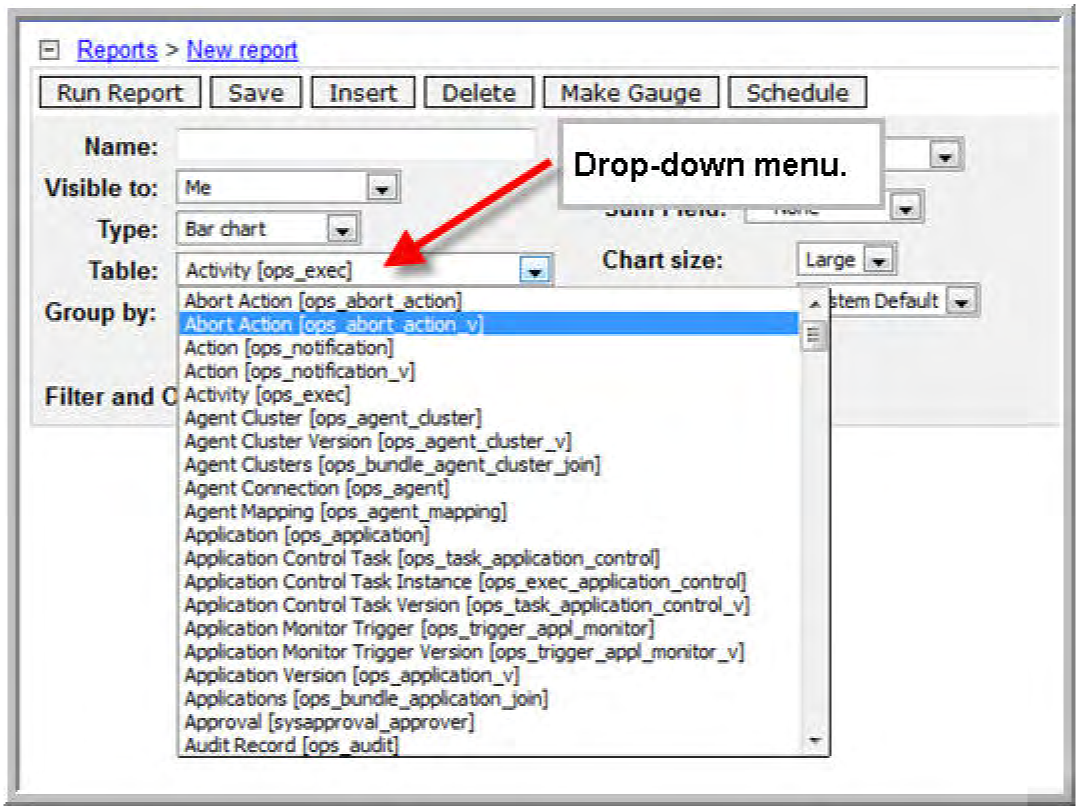
Cron Trigger | = Required field

Trigger Name:	Opswise - 1st and 15th
Task(s):	Opswise - Sleep 30
Calendar:	Opswise - American Calendar
Skip Count:	0
Skip Trigger if Active:	□
Minutes:	45
Hours:	4
Day of Month:	1,15
Month:	*
Day of Week:	*
Description:	4:45 on the 1st and 15th of each month
Cron Criteria:	45 4 1,15 * *
Special Restriction:	☑
Simple Restriction:	☑
Complex Restriction:	□
Action:	Previous Business Day

Drop Down Menus

Drop down menus appear frequently in the UI. For example, when you are creating a report, a drop down menu displays a list of tables available for your report. Drop down menus provide several methods that help you locate the record you are looking for:

- Use the up/down arrows and the scroll box to scroll through the list.
- Type a letter to jump to the first record in the list beginning with that letter. For example, type R to jump to the first record beginning with R. You then can continue pressing R to scroll through only those records in the list beginning with R.


Reports > New report
Run Report
Save
Insert
Delete
Make Gauge
Schedule
Abort Action [ops_abort_action]
Abort Action [ops_abort_action_v]
Action [ops_notification]
Action [ops_notification_v]
Activity [ops_exec]
Agent Cluster [ops_agent_cluster]
Agent Cluster Version [ops_agent_cluster_v]
Agent Clusters [ops_bundle_agent_cluster_join]
Agent Connection [ops_agent]
Agent Mapping [ops_agent_mapping]
Application [ops_application]
Application Control Task [ops_task_application_control]
Application Control Task Instance [ops_exec_application_control]
Application Control Task Version [ops_task_application_control_v]
Application Monitor Trigger [ops_trigger_appl_monitor]
Application Monitor Trigger Version [ops_trigger_appl_monitor_v]
Application Version [ops_application_v]
Applications [ops_bundle_application_join]
Approval [sysapproval_approver]
Audit Record [ops_audit]

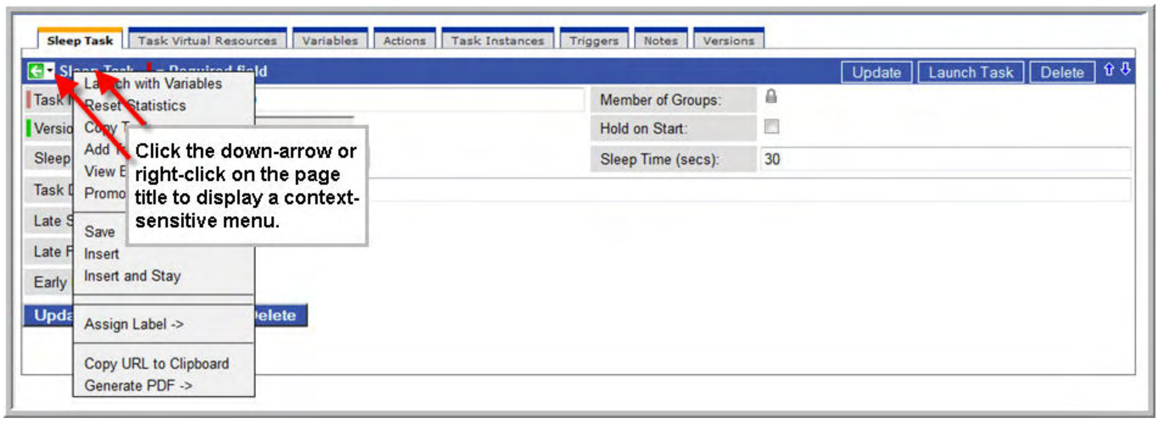
Name:
Visible to: Me
Type: Bar chart
Table: Activity [ops_exec]
Group by:
Filter and C
Chart size: Large
stem Default

Drop-down menu.

Context-Sensitive Menus

You can access context-sensitive menus from lists and forms. From a list, click on the down arrow or right-click on a column header. From a form, click on the down arrow or right-click on the title of the page on the leftmost end of the title bar. If the right-click produces the usual browser menu (back, forward, reload and so on), reposition the cursor further to the left and try again.

The contents of the menu varies, depending on the context and your user permissions. Shown below is a sample right-click menu.



Member of Groups:
Hold on Start: □
Sleep Time (secs): 30

Click the down-arrow or right-click on the page title to display a context-sensitive menu.

The list below shows basic right-click menu options that appear in most contexts:

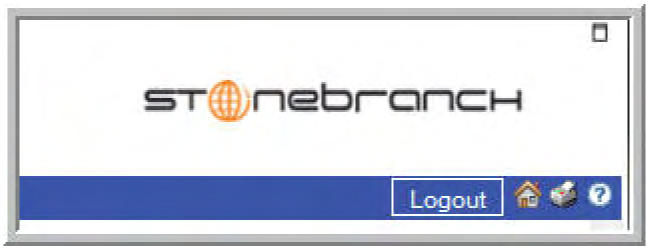
Save — See Saving, Updating, Deleting, and Copying Records

Insert — See Saving, Updating, Deleting, and Copying Records.

Insert and Stay — See Saving, Updating, Deleting, and Copying Records.

Logout/Login Button

To login and logout of Opswise, click the **Login** or **Logout** button, respectively.



Home Button

To return to your home page, click the Home 🏠 icon.

Print Button

To print the contents of the current page, click the Print 🖨 icon.

Help Button

To display help, click the Help ❓ icon.

Using Lists and Forms

A list is a display of records of the same type, such as a list of tasks, calendars, users, and so on. A form is a screen used to enter and save a record, such as the Task definition screen.

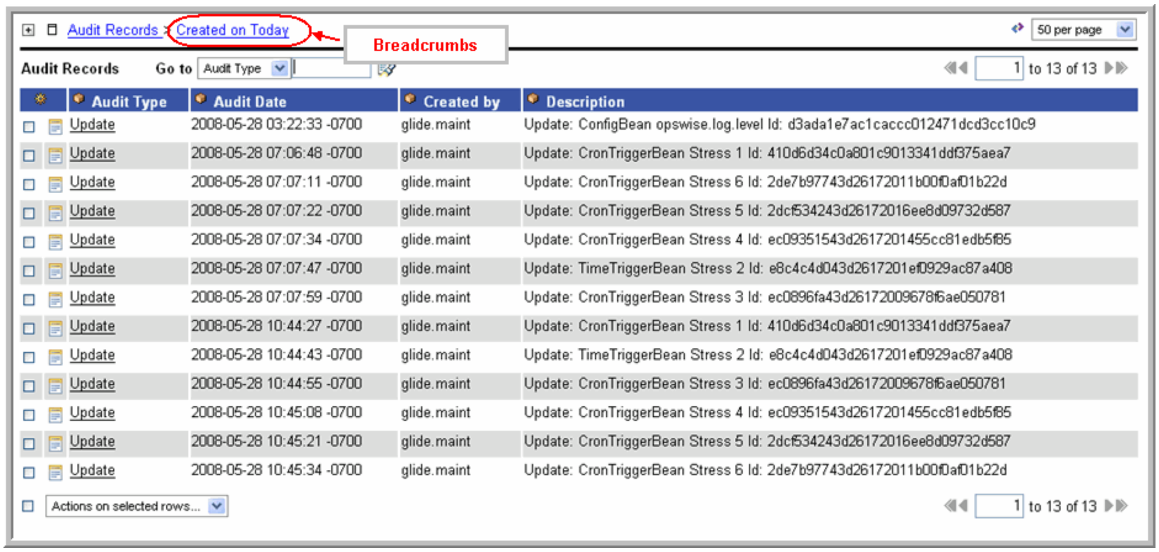
Using Lists

Introduction

A list is a display of records of the same type, such as a list of tasks, calendars, users, and so on. When you click on an application or module in the Opswise navigation pane, the system displays a list of associated records in the center pane.

Breadcrumbs

All lists display breadcrumbs. Breadcrumbs keep track of your location within the UI, and what, if any, filters you have applied. The example below shows breadcrumbs that represent the application being displayed (Audits), along with a filter (Created on Today) that has been applied to the list.

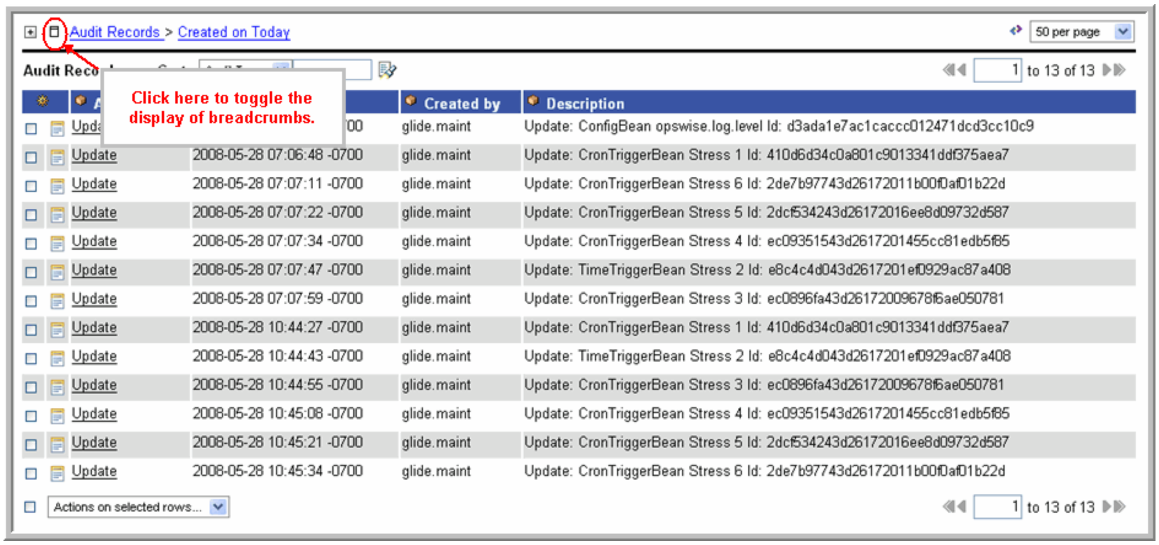


Audit Records

	Audit Type	Audit Date	Created by	Description
□	Update	2008-05-28 03:22:33 -0700	glide.maint	Update: ConfigBean opswise.log.level Id: d3ada1e7ac1caccc012471dcd3cc10c9
□	Update	2008-05-28 07:06:48 -0700	glide.maint	Update: CronTriggerBean Stress 1 Id: 410d6d34c0a801c9013341ddf375aea7
□	Update	2008-05-28 07:07:11 -0700	glide.maint	Update: CronTriggerBean Stress 6 Id: 2de7b97743d26172011b00f0af01b22d
□	Update	2008-05-28 07:07:22 -0700	glide.maint	Update: CronTriggerBean Stress 5 Id: 2dcf534243d26172016ee8d09732d587
□	Update	2008-05-28 07:07:34 -0700	glide.maint	Update: CronTriggerBean Stress 4 Id: ec09351543d2617201455cc81edb5f85
□	Update	2008-05-28 07:07:47 -0700	glide.maint	Update: TimeTriggerBean Stress 2 Id: e8c4c4d043d2617201ef0929ac87a408
□	Update	2008-05-28 07:07:59 -0700	glide.maint	Update: CronTriggerBean Stress 3 Id: ec0896fa43d26172009678f6ae050781
□	Update	2008-05-28 10:44:27 -0700	glide.maint	Update: CronTriggerBean Stress 1 Id: 410d6d34c0a801c9013341ddf375aea7
□	Update	2008-05-28 10:44:43 -0700	glide.maint	Update: TimeTriggerBean Stress 2 Id: e8c4c4d043d2617201ef0929ac87a408
□	Update	2008-05-28 10:44:55 -0700	glide.maint	Update: CronTriggerBean Stress 3 Id: ec0896fa43d26172009678f6ae050781
□	Update	2008-05-28 10:45:08 -0700	glide.maint	Update: CronTriggerBean Stress 4 Id: ec09351543d2617201455cc81edb5f85
□	Update	2008-05-28 10:45:21 -0700	glide.maint	Update: CronTriggerBean Stress 5 Id: 2dcf534243d26172016ee8d09732d587
□	Update	2008-05-28 10:45:34 -0700	glide.maint	Update: CronTriggerBean Stress 6 Id: 2de7b97743d26172011b00f0af01b22d

To toggle breadcrumbs

Click on the icon next to the topmost breadcrumb, as shown in the example below.



Audit Records > Created on Today

Audit Reco

Click here to toggle the display of breadcrumbs.

		A		Created by	Description
□		Upda		glide.maint	Update: ConfigBean opswise.log.level Id: d3ada1e7ac1caccc012471dcd3cc10c9
□		Update	2008-05-28 07:06:48 -0700	glide.maint	Update: CronTriggerBean Stress 1 Id: 410d6d34c0a801c9013341ddf375aea7
□		Update	2008-05-28 07:07:11 -0700	glide.maint	Update: CronTriggerBean Stress 6 Id: 2de7b97743d26172011b00f0af01b22d
□		Update	2008-05-28 07:07:22 -0700	glide.maint	Update: CronTriggerBean Stress 5 Id: 2dcf534243d26172016ee8d09732d587
□		Update	2008-05-28 07:07:34 -0700	glide.maint	Update: CronTriggerBean Stress 4 Id: ec09351543d2617201455cc81edb5f85
□		Update	2008-05-28 07:07:47 -0700	glide.maint	Update: TimeTriggerBean Stress 2 Id: e8c4c4d043d2617201ef0929ac87a408
□		Update	2008-05-28 07:07:59 -0700	glide.maint	Update: CronTriggerBean Stress 3 Id: ec0896fa43d26172009678f6ae050781
□		Update	2008-05-28 10:44:27 -0700	glide.maint	Update: CronTriggerBean Stress 1 Id: 410d6d34c0a801c9013341ddf375aea7
□		Update	2008-05-28 10:44:43 -0700	glide.maint	Update: TimeTriggerBean Stress 2 Id: e8c4c4d043d2617201ef0929ac87a408
□		Update	2008-05-28 10:44:55 -0700	glide.maint	Update: CronTriggerBean Stress 3 Id: ec0896fa43d26172009678f6ae050781
□		Update	2008-05-28 10:45:08 -0700	glide.maint	Update: CronTriggerBean Stress 4 Id: ec09351543d2617201455cc81edb5f85
□		Update	2008-05-28 10:45:21 -0700	glide.maint	Update: CronTriggerBean Stress 5 Id: 2dcf534243d26172016ee8d09732d587
□		Update	2008-05-28 10:45:34 -0700	glide.maint	Update: CronTriggerBean Stress 6 Id: 2de7b97743d26172011b00f0af01b22d

Sorting and Filtering

To specify record sorting

- Click on the column name you want to sort on. A small arrow appears to the left of the name to indicate the direction of sorting. An up arrow indicates ascending alphabetical or numerical order; a down arrow indicates the opposite. Click again to reverse the direction of sorting.
  or
- Position the cursor to the right of the column name you want to sort on and right click. The UI displays a menu that includes the following options:
  - Click **Sort (a to z)** to sort objects in this column in ascending alphabetical or numerical order.
  - Click **Sort (z to a)** to sort objects in this column in descending order.

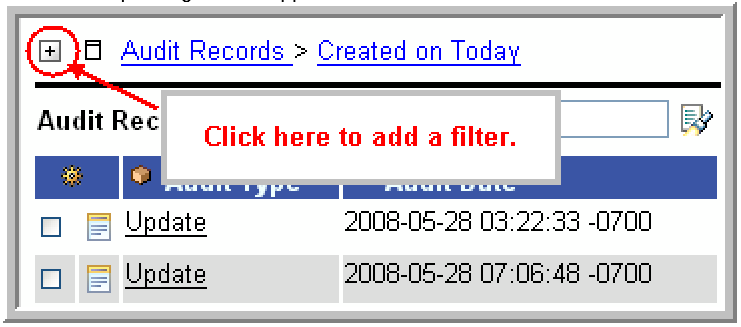
To filter out a single record from a list

1. Right-click on the record name.
2. Select **Filter Out**. This quick filter is added as a breadcrumb.
3. To filter the object back in, click on the prior breadcrumb.

To create a filter

You can create a filter and either run it immediately or save it to apply it later.

1. Click on the plus sign in the upper left corner.



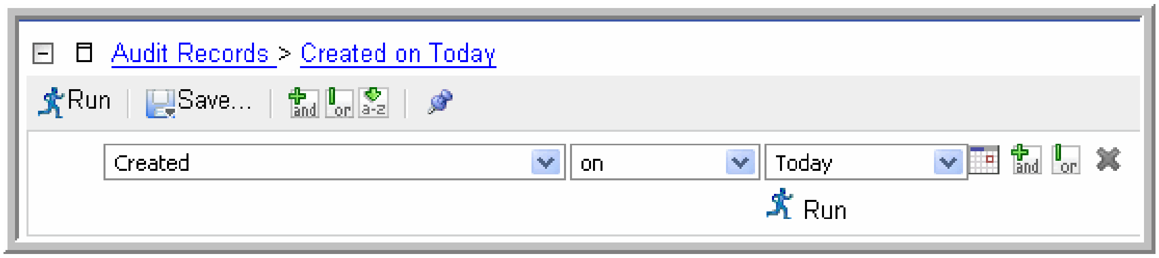
Audit Records > Created on Today

Audit Rec

Click here to add a filter.

□		Update	2008-05-28 03:22:33 -0700
□		Update	2008-05-28 07:06:48 -0700

2. The UI displays the filter dialog.



3. The dialog allows you to specify a condition consisting of:
   - A field from this record type
   - A relevant comparison operator, and
   - A relevant object or field to compare the field with.
4. Make your selections from the drop down menus. Examples include:
   - Create Time before Today
   - Owner is Username
   - Enabled is False
   - Created by starts with S
5. Optionally, you can enhance your filters using any of the following:
   - To add an AND condition, click and
   - To add an OR condition, click or
   - To sort, click a-z

To run a filter

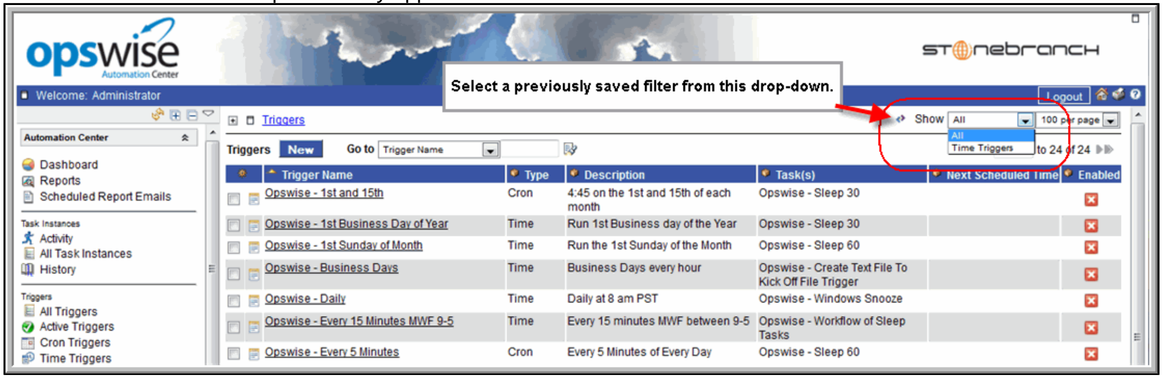
1. To apply the filter to the list, click **Run**. Note that the filter name is added to the breadcrumb.

To save a filter

1. Click **Save**.
2. Specify the filter name.
3. Specify which users are allowed to use the filter.
4. Click the **Save** button.

To run a saved filter

Select the saved filter you want to run from the **Show** drop down just below the title bar. The UI applies the filter and adds the filter name to the breadcrumbs. The Show drop-down only appears if there are saved filters associated with this list.

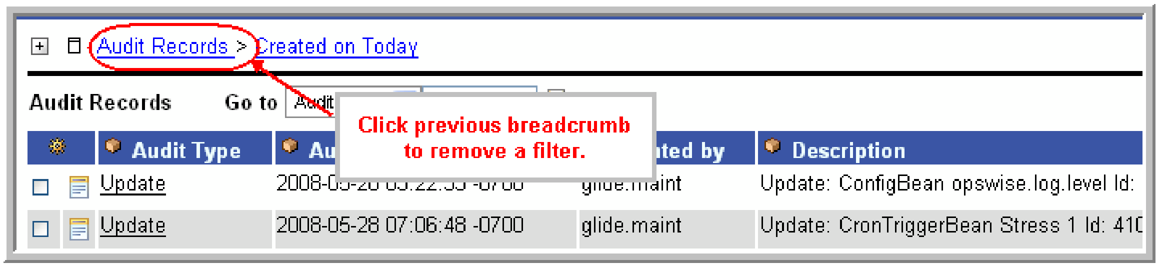


Select a previously saved filter from this drop-down.

	Trigger Name	Type	Description	Task(s)	Next Scheduled Time	Enabled
□	Opswise - 1st and 15th	Cron	4:45 on the 1st and 15th of each month	Opswise - Sleep 30		✗
□	Opswise - 1st Business Day of Year	Time	Run 1st Business day of the Year	Opswise - Sleep 30		✗
□	Opswise - 1st Sunday of Month	Time	Run the 1st Sunday of the Month	Opswise - Sleep 60		✗
□	Opswise - Business Days	Time	Business Days every hour	Opswise - Create Text File To Kick Off File Trigger		✗
□	Opswise - Daily	Time	Daily at 8 am PST	Opswise - Windows Snooze		✗
□	Opswise - Every 15 Minutes MWF 9-5	Time	Every 15 minutes MWF between 9-5	Opswise - Workflow of Sleep Tasks		✗
□	Opswise - Every 5 Minutes	Cron	Every 5 Minutes of Every Day	Opswise - Sleep 60		✗

To remove filtering from a list

Click on the previous breadcrumb, as shown on the example below.



Audit Records > Created on Today

Audit Records Go to Audit

Click previous breadcrumb to remove a filter.

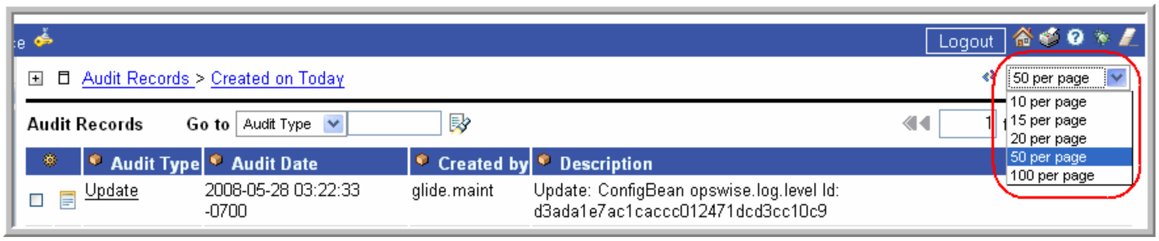
	Audit Type	Au	ted by	Description
□	Update	2008-05-28 03:22:33 -0700	glide.maint	Update: ConfigBean opswise.log.level Id:
□	Update	2008-05-28 07:06:48 -0700	glide.maint	Update: CronTriggerBean Stress 1 Id: 41C

To manage filters

From the navigation pane, select **Automation Center Administration > Filters.** The system displays all filters for which you have permission. This feature allows you to modify or delete filters. To open a filter record, click on the record name. You can make changes and click **Update** or click **Delete** to remove a filter from the system.

Specifying the Number of Records Per Page

From the dropdown in the upper right corner, select the number of records per page you want to display.



Audit Records > Created on Today

Audit Records Go to Audit Type

	Audit Type	Audit Date	Created by	Description
□	Update	2008-05-28 03:22:33 -0700	glide.maint	Update: ConfigBean opswise.log.level Id: d3ada1e7ac1caccc012471dcd3cc10c9

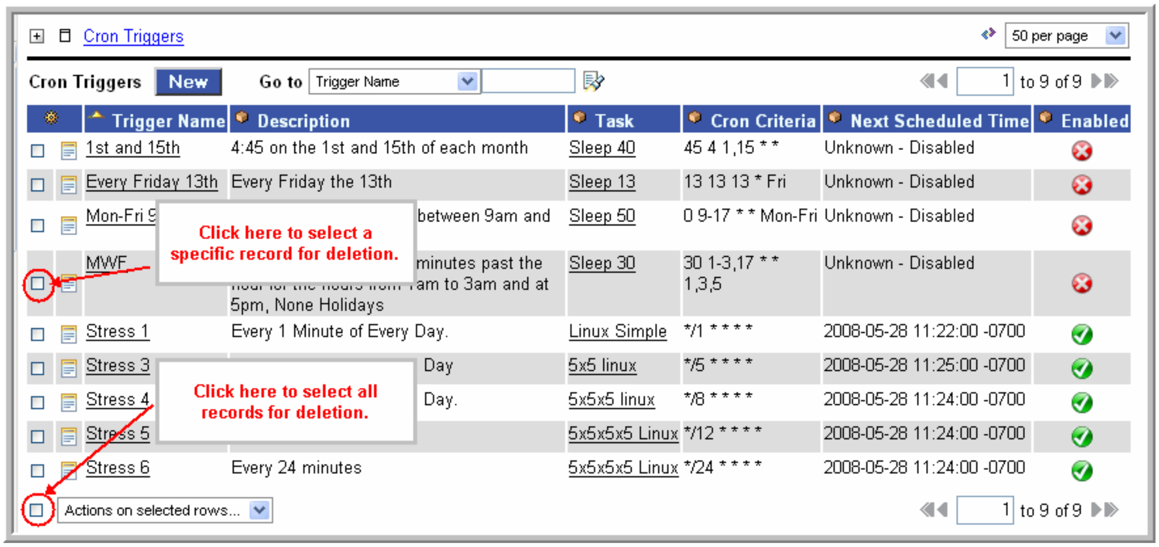
Adding Records

Adding a new record is a simple uniform procedure through the environment. From any list, click the **New** button.

Deleting Records

You can delete one or more records as follows:

- To delete one or more specific records, click on the box associated with those records and select **Delete** from the **Actions on selected rows...** menu.



Cron Triggers Go to Trigger Name 1 to 9 of 9

	Trigger Name	Description	Task	Cron Criteria	Next Scheduled Time	Enabled
□	1st and 15th	4:45 on the 1st and 15th of each month	Sleep 40	45 4 1,15 * *	Unknown - Disabled	✗
□	Every Friday 13th	Every Friday the 13th	Sleep 13	13 13 13 * Fri	Unknown - Disabled	✗
□	Mon-Fri 9	between 9am and	Sleep 50	0 9-17 * * Mon-Fri	Unknown - Disabled	✗
□	MWF	minutes past the am to 3am and at 5pm, None Holidays	Sleep 30	30 1-3,17 * * 1,3,5	Unknown - Disabled	✗
□	Stress 1	Every 1 Minute of Every Day.	Linux Simple	*/1 * * * *	2008-05-28 11:22:00 -0700	✓
□	Stress 3	Day	5x5 linux	*/5 * * * *	2008-05-28 11:25:00 -0700	✓
□	Stress 4	Day.	5x5x5 linux	*/8 * * * *	2008-05-28 11:24:00 -0700	✓
□	Stress 5		5x5x5x5 Linux	*/12 * * * *	2008-05-28 11:24:00 -0700	✓
□	Stress 6	Every 24 minutes	5x5x5x5 Linux	*/24 * * * *	2008-05-28 11:24:00 -0700	✓

□ Actions on selected rows... 1 to 9 of 9

Click here to select a specific record for deletion.

Click here to select all records for deletion.

- To delete all records currently being displayed on the list, click on the box next to the **Actions on selected rows**... menu and select **Delete**.

Updating Multiple Records

Exercise great caution when using this powerful feature.

Two menu options available from any list of records allow you to make updates to multiple records. You can make these "global" changes to the list for any field that is common to every record in the list or every selected record. For example, if your list includes multiple Cron triggers, you can make "global" changes to all fields on a Cron trigger. If your list includes all task types, you can make "global" changes to any field that is common to all task types. The more uniform your list is, the greater the number of fields you can change.

The menu options are:

- **Update Entire List** - Allows you to make updates to all records that match your selection filter (not just those on the current display). If you specify no filter, the system selects all records of that type.
- **Update Selected** - Allows you to select specific records to update by clicking the box to the left of the record name.

In order to use these features, your user ID must have the **List_Updater** role.

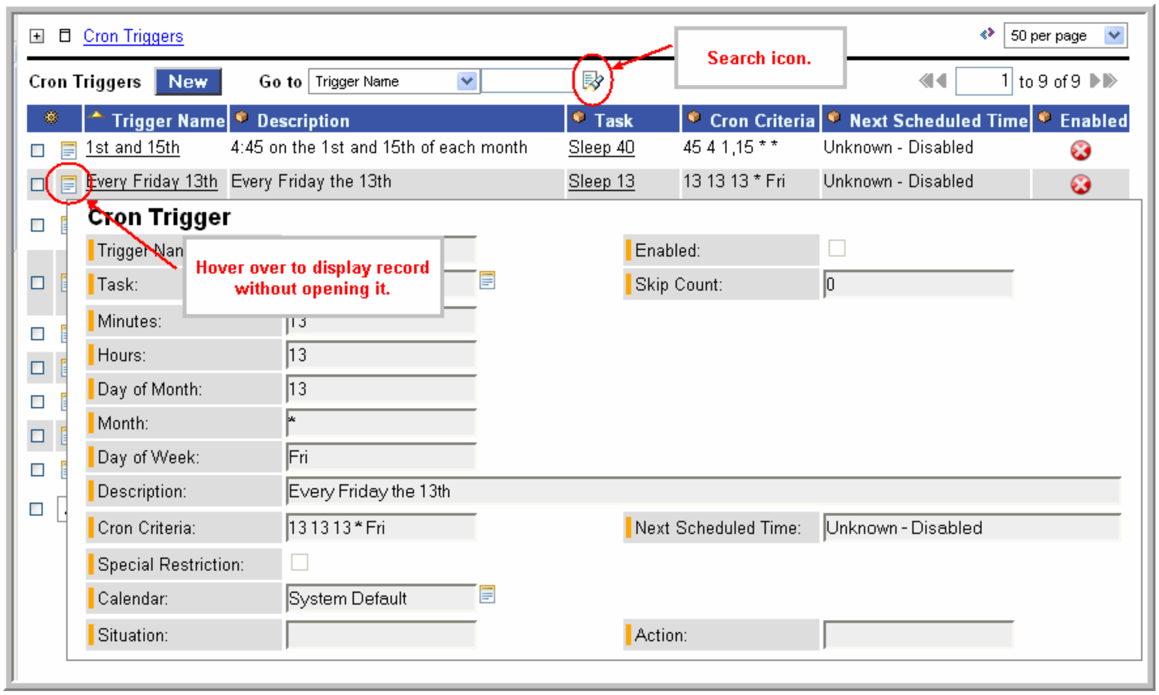
1. Optionally, filter the list so that it selects only the records you want to update.
2. If you only want to update some of the records in the displayed list, select those records by clicking the box to the left of the name.
3. Right-click on the blue title bar. A menu displays.
4. Select **Update Entire List** or **Update Selected** as appropriate. A form pops up, showing which fields you will be allowed to change. These fields are common to all the records selected by your filter or manually selected by you. If you did not use a filter or select specific records, all records of this type are considered "selected."
5. Enter the global changes you want to make. For drop-down menus and true/false fields, the default selection **None** means that the field will not be changed on any records. If you select anything else in such a field, the new value will be applied to all selected records.
6. When you are finished, click **Update**. The system displays the number of records that will be updated and asks you to confirm the operation.
7. Once you confirm, the system applies your changes to all the selected records.

Quickly Displaying a Record's Contents

You can display the contents of a record without opening it. Hover over the paper icon to the left of the record name, as shown in the illustration below.

Searching for Records

You can perform searches from any list, using the Go to boxes, shown below.



Trigger Name	Description	Task	Cron Criteria	Next Scheduled Time	Enabled
1st and 15th	4:45 on the 1st and 15th of each month	Sleep 40	45 4 1,15 * *	Unknown - Disabled	✗
Every Friday 13th	Every Friday the 13th	Sleep 13	13 13 13 * Fri	Unknown - Disabled	✗

Cron Trigger

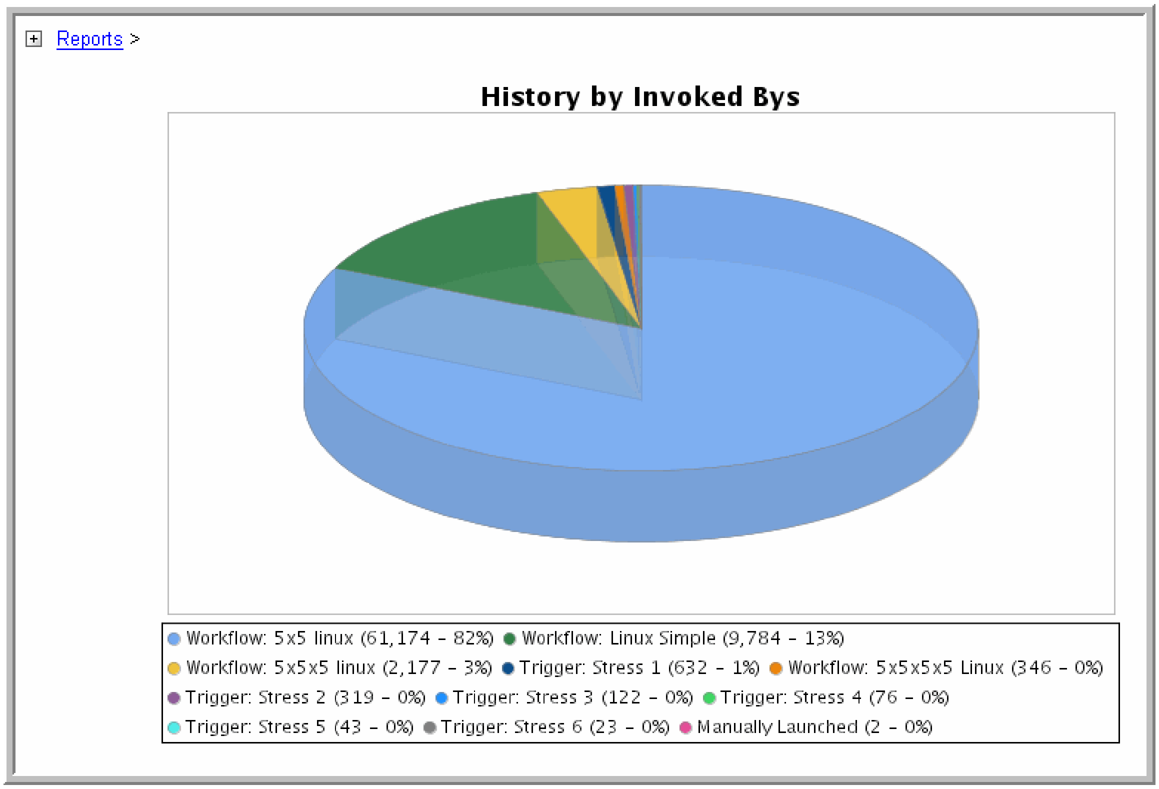
Trigger Name:		Enabled:	□
Task:		Skip Count:	0
Minutes:	13		
Hours:	13		
Day of Month:	13		
Month:	*		
Day of Week:	Fri		
Description:	Every Friday the 13th		
Cron Criteria:	13 13 13 * Fri	Next Scheduled Time:	Unknown - Disabled
Special Restriction:	□		
Calendar:	System Default		
Situation:		Action:	

The options in the Go to box vary depending on the module and the columns displaying on the list. To perform a search:

1. Select the field you want to search on. For example, Name.
2. Enter a value appropriate to that field. For text fields, you can search on a full or partial string.
3. Click the search icon. The search fetches all records that match and places them at the top of the list.

Generating Pie or Bar Charts from a List

This feature allows you to quickly generate a pie or bar chart that sorts records in the list based on a specific column. For example, you can quickly generate a pie chart that sorts tasks by type. In the example below, the pie chart below was generated by right-clicking the "Invoked By" column on the Activity History list and selecting Pie Chart.


⊞ Reports >
History by Invoked Bys
● Workflow: 5x5 linux (61,174 - 82%)
● Workflow: Linux Simple (9,784 - 13%)
● Workflow: 5x5x5 linux (2,177 - 3%)
● Trigger: Stress 1 (632 - 1%)
● Workflow: 5x5x5x5 Linux (346 - 0%)
● Trigger: Stress 2 (319 - 0%)
● Trigger: Stress 3 (122 - 0%)
● Trigger: Stress 4 (76 - 0%)
● Trigger: Stress 5 (43 - 0%)
● Trigger: Stress 6 (23 - 0%)
● Manually Launched (2 - 0%)

1. Click in the column header from which you want to generate a chart.
2. Right-click and select **Pie Chart** or **Bar Chart**. The UI generates the chart.
3. Optionally, click the Print icon in the upper right corner to print the chart. To escape the chart, select any menu option or your browser's Back key.

Exporting Lists to an Output File

This feature allows you to save basic information about the contents of list to one of the following file types: Excel, CSV (comma-separated values in an Excel file) or XML. This procedure saves all records in the list, not just those currently displayed.

1. Right-click in any column header of the list.
2. Select **Export**.
3. Select **Excel , CSV**, **XML**, or **XML (Export References)**. If you choose XML when exporting a Task record, you will also get local variables, Actions, Notes, and Virtual Resource requirements. If you choose XML (Export References) on a Task, you will get all of the previously described data, plus triggers, global variables, calendars, and any resources referred to. The "reference" records are saved to separate XML files. If you are exporting a Workflow, choosing XML will also export all the tasks in the workflow; each task is exported to a separate XML file.
4. Follow the prompts to download and save or open the output file.

You can also export the contents of a list to a PDF file. Right-click, select Generate PDF, and select the output layout.

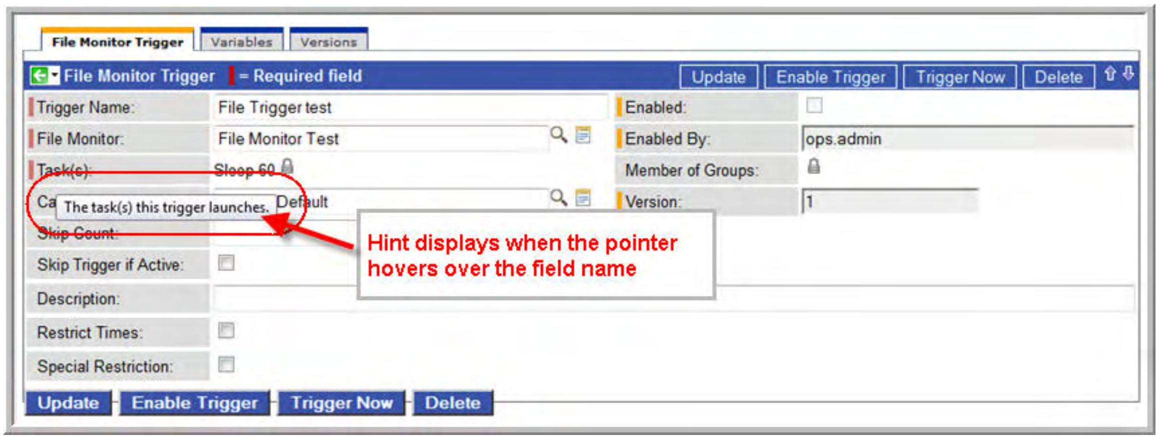
Using Forms

Introduction

A "form" is the screen used to create a record.

Field Hints

To display a brief description of a field, hover the pointer over the field name. A hint displays, as shown in the following example:



Trigger Name:	File Trigger test	Enabled:	□
File Monitor:	File Monitor Test	Enabled By:	ops.admin
Task(s):	Sleep 60	Member of Groups:	
Ca	Default	Version:	1
Skip Count:			
Skip Trigger if Active:	□		
Description:			
Restrict Times:	□		
Special Restriction:	□		

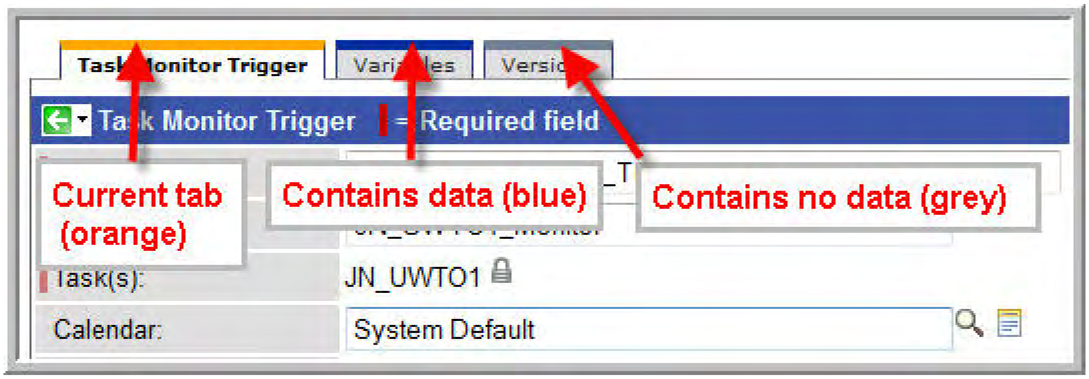
The task(s) this trigger launches.

Hint displays when the pointer hovers over the field name

Color-Coding of Tabs

Tabs on a record are color-coded as follows:

- The current tab displays an orange rule.
- Tabs that contain data display a blue rule
- Tabs that do not contain data display a grey rule



Current tab (orange)

Contains data (blue)

Contains no data (grey)

Task(s):	JN_UWTO1
Calendar:	System Default

Saving, Updating, Deleting, and Copying Records

To save a new record and return to the previous menu

1. Click **New** from the list screen, select the record type, if appropriate, and fill out the fields as required.
2. Click the **Submit** button.

To save a new record without returning to the menu

1. Click **New** from the list screen, select the record type, if appropriate, and fill out the fields as required.
2. Hover over the down arrow or right-click on the title bar and select **Save**.

To update a record and return to the previous menu

1. Display the record and make the updates you want.
2. Click the **Update** button. Note that if you changed the name of a task that's used in a workflow, the update is also reflected in the workflow.

To update a record without returning to the menu

1. Display the record and make the updates you want.
2. Hover over the down arrow or right-click on the title bar and select **Save**.

To delete a record

1. Display the record you want to delete.
2. Click the **Delete** button and click **Yes** to confirm.
3. You can also delete records from the list screen. See Deleting Records.

To copy a record

> ⚠ Generally, you should not use this method to copy a task, trigger, or calendar unless there are no associated records such as Actions, Notes, Variables, and so on. This method does not copy any records attached to the record. To copy a task, trigger, or calendar with all its associated records, use the method described in Copying Tasks, Copying Triggers, and Copying Calendars.

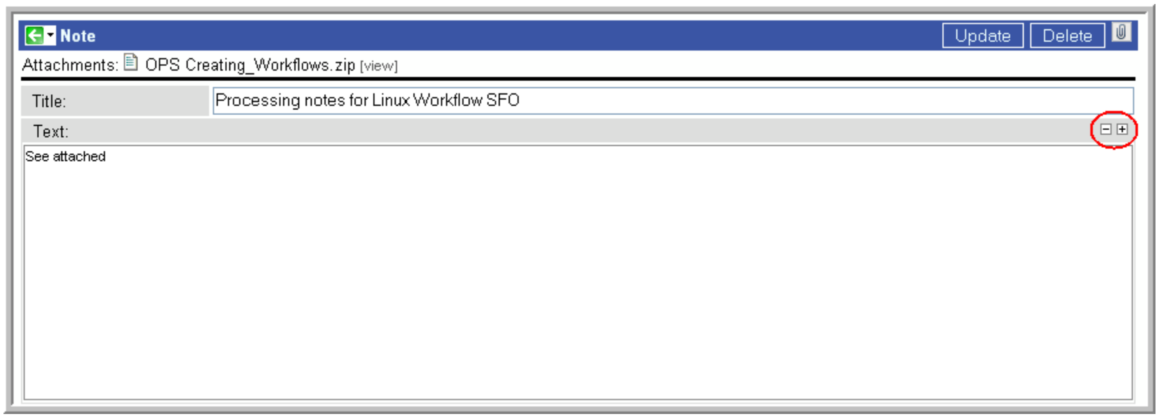
1. Display the record definition you want to copy.
2. Give the record a new name and make any other necessary changes.
3. Hover over the down arrow or right-click on the title bar and select Insert or **Insert and Stay**.
   Note • Do not use the **Update** button when you are copying a record. The **Update** button overwrites the existing record.

Exporting Record Definitions to XML

1. Open the record you want to export.
2. Hover over the down arrow or right-click on the title bar and select one of the following:
   - Export as XML to export the current record to an XML file.
   - Export All as XML*to export all records of this type to an XML file.
3. Specify whether you want to open or save the XML.

Expanding/Contracting a Text Field

Free-text fields are defaulted to a particular display size. You can expand and contract the display of the field by clicking the plus ⊕ and minus ⊖ signs at the top right corner of the field.



Attachments: OPS Creating_Workflows.zip [view]

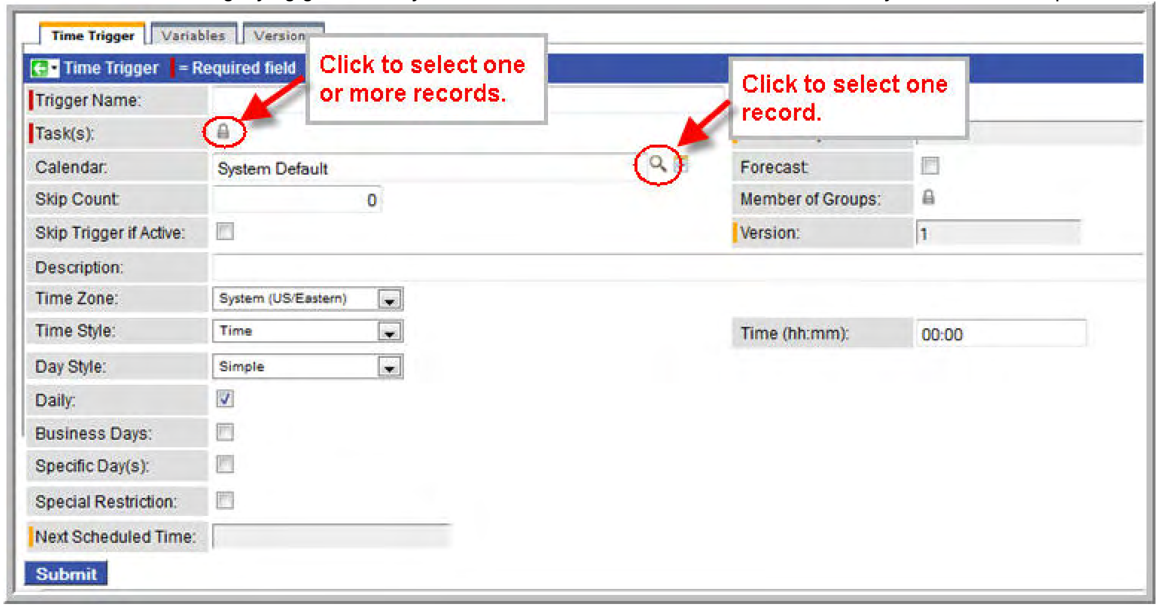
Title: Processing notes for Linux Workflow SFO

Text:

See attached

Browsing For and Selecting One or More Records

Many fields contain data from another table. This type of field, called a Reference field, has either a magnifying glass or a lock icon at the right end. Fields with the magnifying glass allow you to select one record; fields with the lock allow you to select multiple records.

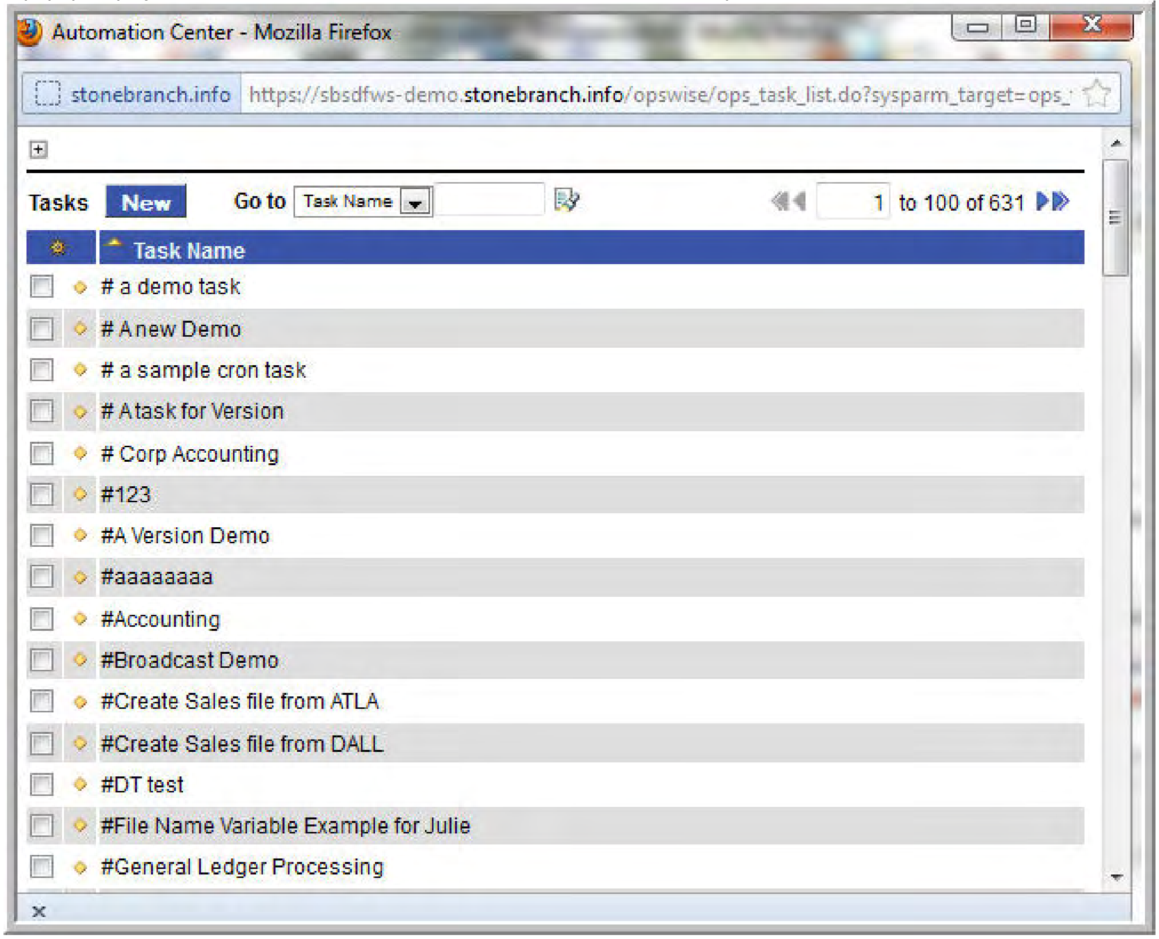


Click to select one or more records.

Click to select one record.

Trigger Name:			
Task(s):			
Calendar:	System Default	Forecast:	□
Skip Count:	0	Member of Groups:	
Skip Trigger if Active:	□	Version:	1
Description:			
Time Zone:	System (US/Eastern)		
Time Style:	Time	Time (hh:mm):	00:00
Day Style:	Simple		
Daily:	☑		
Business Days:	□		
Specific Day(s):	□		
Special Restriction:	□		
Next Scheduled Time:			

1. To select records for a multiple record field, click the lock icon. Two additional fields display – a browse field with a magnifying glass and a field above it that will contain the record names you select. You can also delete ✖ and view ◇ records. To select records from a browse field, click the magnifying glass.
2. A popup displays the contents of the associated table, as shown in the example below. To select a record, click on its name.



Automation Center - Mozilla Firefox
stonebranch.info
https://sbsdfws-demo.stonebranch.info/opswise/ops_task_list.do?sysparm_target=ops_
Tasks
New
Go to
Task Name
1 to 100 of 631

	Task Name
□ ◇	# a demo task
□ ◇	# A new Demo
□ ◇	# a sample cron task
□ ◇	# A task for Version
□ ◇	# Corp Accounting
□ ◇	#123
□ ◇	#A Version Demo
□ ◇	#aaaaaaaa
□ ◇	#Accounting
□ ◇	#Broadcast Demo
□ ◇	#Create Sales file from ATLA
□ ◇	#Create Sales file from DALL
□ ◇	#DT test
□ ◇	#File Name Variable Example for Julie
□ ◇	#General Ledger Processing

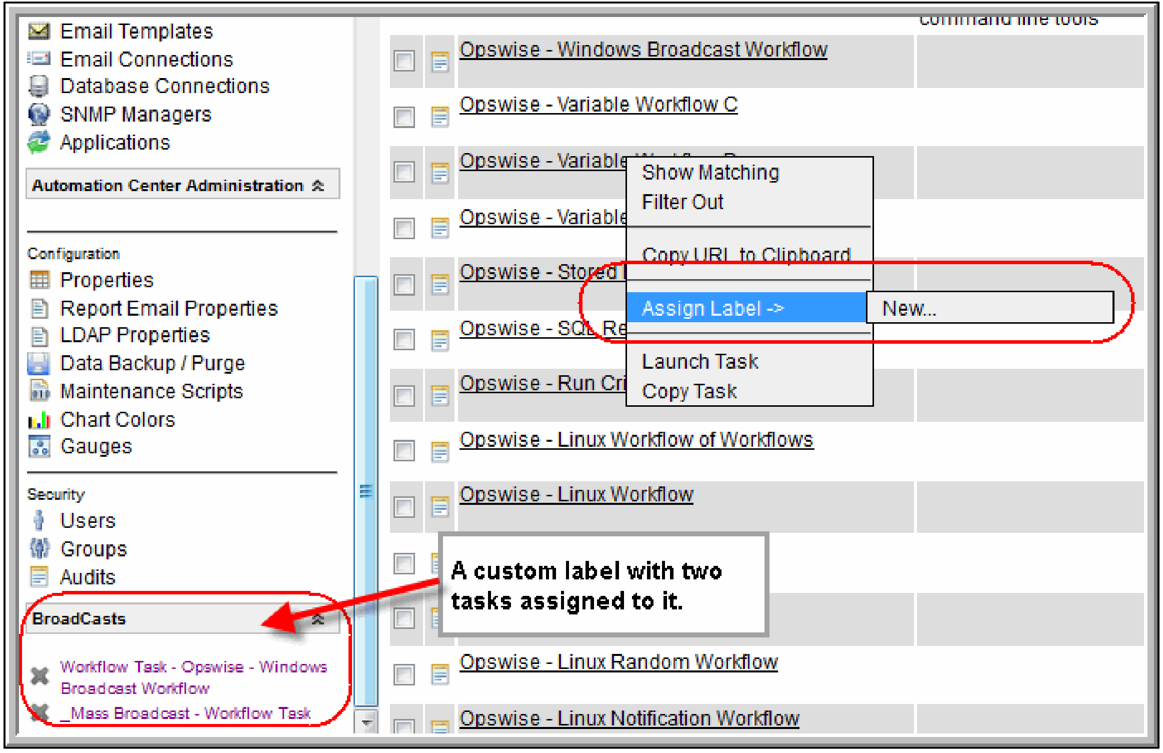
Customizing the Navigation Pane

Using Labels, you can add menu options to the navigation pane and assign records to them. This allows you to organize records into business groups for easy access from your navigation pane. For example, you might create a Label for a department and assign all related records to that Label.

When you organize records into Labels, the records are still accessible from their original menus as well.

To create a new Label and assign a record to it

1. Right-click on the record from the list screen, or open the record and hover over the down arrow or right-click on the title bar. The UI displays a menu.
2. Highlight **Assign Label** and click **New**. The UI prompts for a name.
3. Enter a name for the new Label and click **OK**. The UI adds the new Label to the bottom of the navigation pane and adds the current record to it.
4. To view the new Label and its record , click on the carets next to the folder name.



A custom label with two tasks assigned to it.

To assign a record to an existing Label

1. To assign another record to the Label, right-click on the record; or, open the record and hover over the down arrow or right-click on the title bar.
2. Highlight **Assign Label**, select New and type in the name of the existing Label. The UI adds the record to the Label menu.

To remove a record from a Label

1. Click the **X** to the left of the record name under the Label heading; or, display the record, right-click the title bar and select **Remove Label.**

To delete a Label

1. Click on the folder name. The UI displays the Label definition screen.
2. Click the **Delete** button.

Naming Tips

Many functions within Opswise are executed against one or more records. For example, you can give a user permission to change only certain tasks, issue commands against a group of task instances, or filter a trigger list to display only certain triggers.

Two methods are available to help you organize your records to facilitate the use of these functions. One method is to develop a naming scheme for records. For example, when naming tasks, you might prepend with **SF** all tasks related to San Francisco operations. Or, the name of all report-related tasks might begin with **REPT**. With such a naming scheme, you can sort and filter lists by selecting records, for example, that begin with "REPT." You can assign permissions and execute commands against records using the same method.

Another method for organizing records is using Opswise groups. Whenever you create a record, you can assign it to an Opswise group. The Opswise group is simply a method of grouping records. For example, you may have an Opswise group called "SF" or an Opswise group called "REPT." Using this method, you could then filter or sort a list based on the Opswise group. As another example, you could assign permissions to a user, giving him update permission to all records in the "REPT" Opswise group. Opswise groups allow you to create groups based on business functions and organize all your Opswise records according to categories you define.

Opswise Groups

Overview

Opswise groups allow you to organize your data into business groups. You do so by creating Opswise groups that represent your organization and assigning Opswise records, such as tasks and resources, to one or more groups. You can then sort and filter screens based on the groups, as well as generate reports. You can also take advantage of Opswise groups when you set up security by assigning roles and permission only to specific groups.

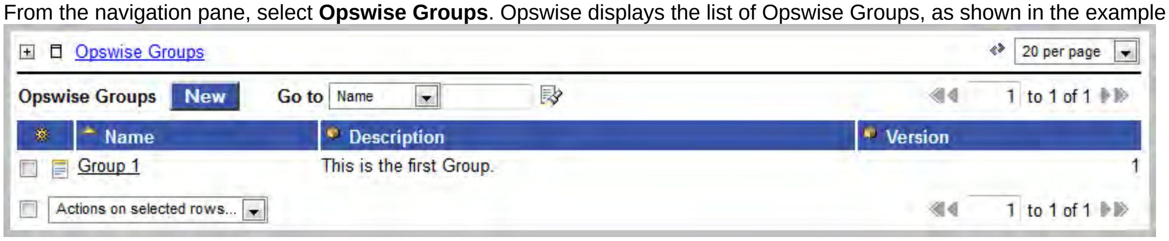
You can assign the following record types to one or more Opswise groups:

- Agents
- Calendars
- Credentials
- Tasks and Task Instances
- Triggers

Creating Opswise Groups

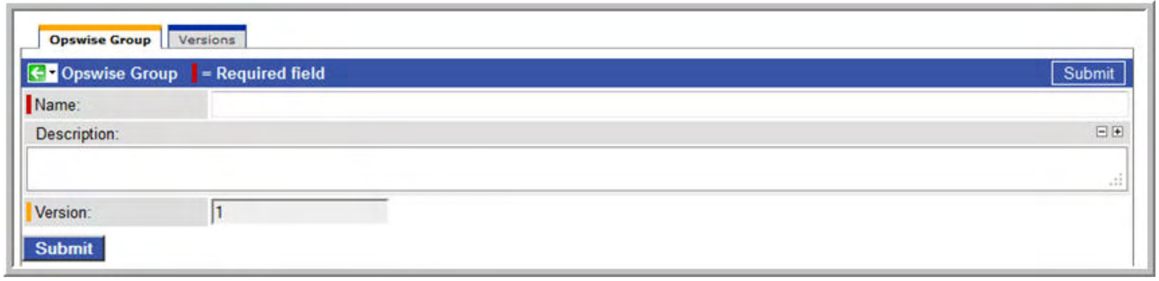
You may need administrative privileges to perform this procedure.

1. From the navigation pane, select **Opswise Groups**. Opswise displays the list of Opswise Groups, as shown in the example below.



Name	Description	Version
Group 1	This is the first Group.	1

2. To create a new group, click **New**. The Opswise Groups screen displays.



Name:

Description:

Version: 1

3. Give the group a name and a description and click **Submit**.

Assigning a Record to One or More Opswise Groups

When defining Opswise records, use the Member of Groups field to select one or more groups.

Using Wildcards

The Opswise Automation Center graphical user interface (GUI) supports two wildcards: the asterisk (*) and the question mark (?). You can use wildcards in record searches and when applying some rule or command against records. Fields that support wildcards are identified in the field description for that field.

- An asterisk (*) represents a wildcard of any number of characters. For example, a search for string "FEE*SF" returns all records whose name begins with "FEE" and ends with "SF", with any number and type of characters between the two strings.
- A question mark (?) represents a wildcard of one character in a specific position. For example, a search for string "FEE?SF" returns all records whose name begins with "FEE" and ends with "SF", with any single character between the two strings. You can also use multiple question marks. For example, a search for string "FEE??SF" returns all records whose name begins and ends with "FEE" and "SF", respectively, with any two characters between the two strings.

Resources

Opswise resources are records that you set up in advance to facilitate your operations. These include:

- Agent-Based resources, such as a Windows or Linux agent.
- Agent clusters, which are groups of agents from which Opswise uses pre-defined parameters to select the best agent.
- Agentless resources, such as an Email or Database connection.
- Virtual resources, which allow you to create a throttling scheme.
- Script library, which allows you to execute scripts stored in the Opswise database.

Agent-Based Resources

Introduction

Opswise agent-based resources refer to the agent - Universal Automation Center Agent (UAG) - running and launching programs on one or more remote machines along with the agent's connector components, which together are referred to as "outboard components." The connector components includes the message hub and transporter. Click here to view a diagram and system overview.

Displaying Information About Agents and Connector Components

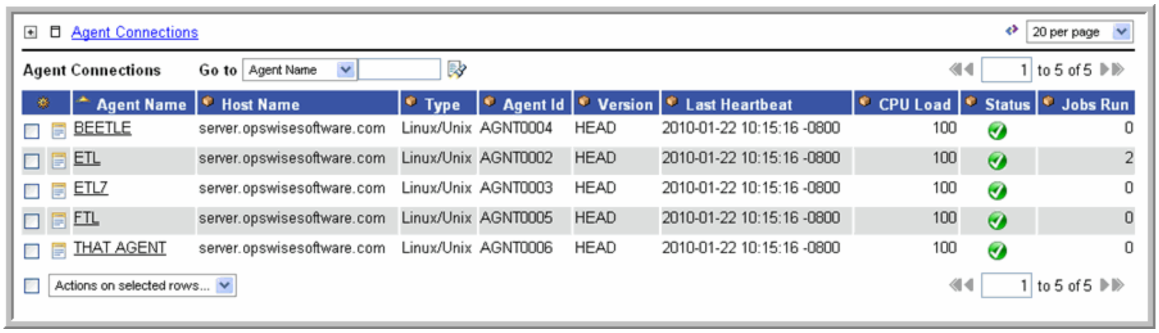
When you start an agent for the first time, Opswise automatically creates a database record for each agent and connector component. You can view these records for details and status information.

Displaying a List of Agents and Their Status

Two methods are available for displaying agent status: from the browser interface and from the command line interface. See Command Line Reference for instructions on using that interface.

Displaying Agent Status from the browser interface

1. From the navigation pane, select **Automation Center Resources > All Agents** (or select the specific type of agent). A screen similar to the one below displays:



	Agent Name	Host Name	Type	Agent Id	Version	Last Heartbeat	CPU Load	Status	Jobs Run
□	BEETLE	server.opswisesoftware.com	Linux/Unix	AGNT0004	HEAD	2010-01-22 10:15:16 -0800	100	✓	0
□	ETL	server.opswisesoftware.com	Linux/Unix	AGNT0002	HEAD	2010-01-22 10:15:16 -0800	100	✓	2
□	ETL7	server.opswisesoftware.com	Linux/Unix	AGNT0003	HEAD	2010-01-22 10:15:16 -0800	100	✓	0
□	FTL	server.opswisesoftware.com	Linux/Unix	AGNT0005	HEAD	2010-01-22 10:15:16 -0800	100	✓	0
□	THAT AGENT	server.opswisesoftware.com	Linux/Unix	AGNT0006	HEAD	2010-01-22 10:15:16 -0800	100	✓	0

2. To display more details about agents from the browser interface, click the Agent name from the agent list (see above). The full agent record displays. Most fields are display-only; however, you can make the following changes:
   - Add a Member of Groups.
   - Assign Credentials.
   - Change the default logging level.
   - Change the heartbeat interval.The heartbeat is a status message sent from the agent to the core processor.
   - Assign this agent to an agent cluster.

Field Name	Description
Agent Name-Agent Resource	Required. Defined by the user when installing the agent. This is the name used within Opswise Automation Center to identify this resource.
Host Name	Specified by the user during installation. The IP address or domain/name of the host machine where the component resides.

Type	Agent's platform: Linux/Unix, Windows, z/OS, or Indesca.
Agent ID	Unique ID for this agent, created during installation. The name of the communications server message queue. Opswise Automation Center uses the communications server for communications between the core scheduler and Opswise Automation Center agents.
Version	System-supplied. Version number of the agent program
Last Heartbeat	System-supplied. The date and time the most recent heartbeat was received from the component.
CPU Load	System-supplied. The current CPU load on the agent machine, expressed as a percentage. For example, 1 means 1% currently utilized.
Status-Agent List	System-supplied. The status of the agent. The green button with the checkmark means the agent is running. A red circle with an X means the agent is not running.
Jobs Run	System-supplied. The number of jobs run on this agent since it was installed.

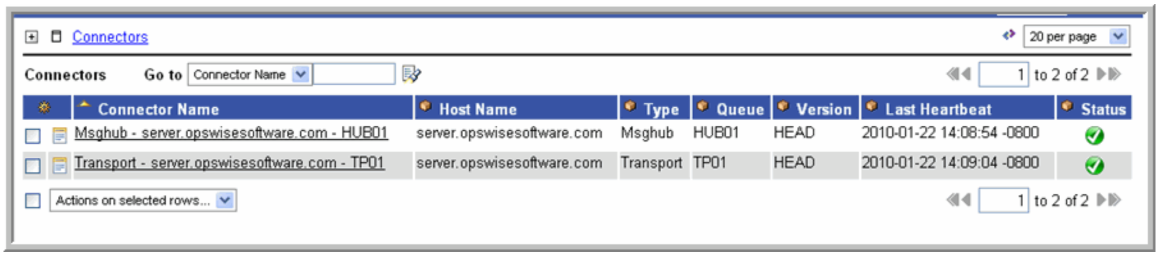
To display detailed field descriptions, click the appropriate link below:

- Linux/Unix Agent
- Windows Agent
- z/OS Agent
- Indesca Agent

Displaying Information about Connectors

A connector is either a message hub or a transporter. The connectors must be up and running in order for you to run tasks on an agent. They start automatically when you start an agent. Normally you only need to consult this display if you are experiencing connection problems with your agent. You can view information about message hubs and transporters from the browser interface as follows:

From the navigation pane, select **Automation Center Resources > Connectors**. The system displays a screen similar to the following:



	Connector Name	Host Name	Type	Queue	Version	Last Heartbeat	Status
□	Msghub - server.opswisesoftware.com - HUB01	server.opswisesoftware.com	Msghub	HUB01	HEAD	2010-01-22 14:08:54 -0800	✓
□	Transport - server.opswisesoftware.com - TP01	server.opswisesoftware.com	Transport	TP01	HEAD	2010-01-22 14:09:04 -0800	✓

Field Name	Description
Connector Name	System-supplied name for this connector.
Host Name	Specified by the user during installation. The IP address or domain/name of the host machine where the component resides.
Type	Either **msghub** or **transport**.

Queue	System-supplied. Internal use only.
Version	System-supplied. Version number of the agent program
Last Heartbeat	System-supplied. The date and time the most recent heartbeat was received from the component.
Status	System-supplied. The status of the component. The green button with the checkmark means the component is active. A red circle with an X means the component is not active.

Starting/Stopping Agents and Connector Components

For instructions on starting and stopping outboard components, click here and select your platform.

Sending Status Notifications on Opswise Components

You can configure any or all of your Opswise components (Opswise Core or Cluster Node, Agents, and Connectors) to send a notification via email or SNMP when the component goes down (Offline) or comes back up (Active).

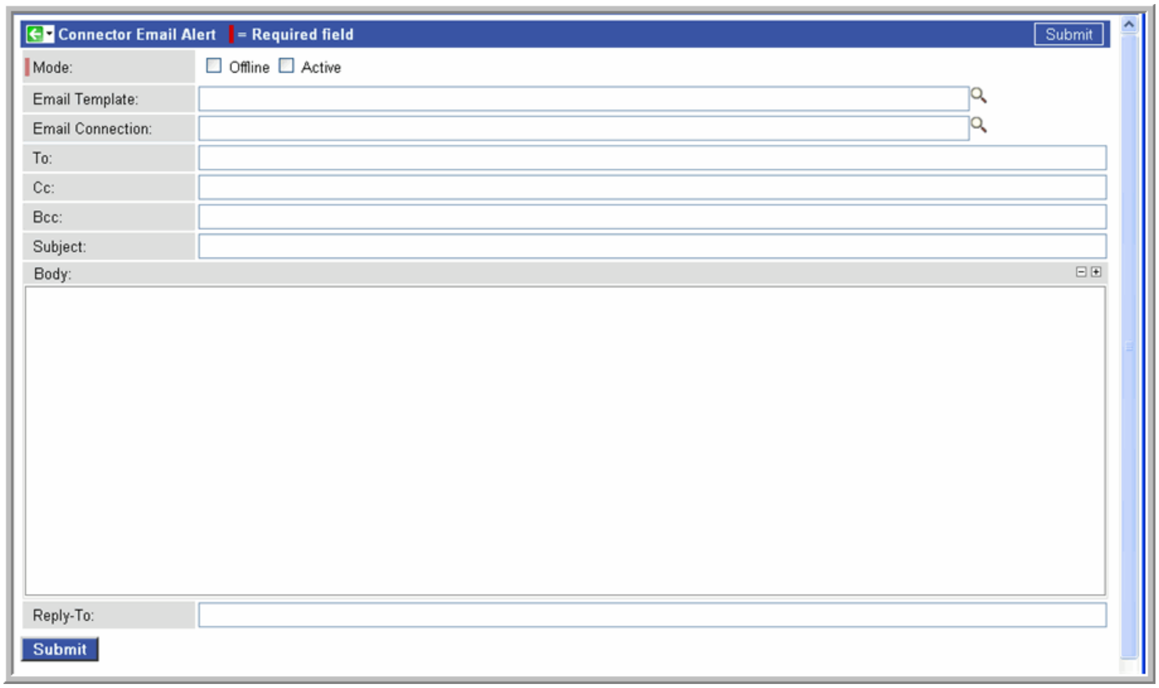
1. From the **Automation Center Resources** section of the navigation pane, select the Opswise component for which you want to configure a notification – UAG, Connector, or Cluster Node (core processor).
2. Open the component record by clicking on it.
3. Click the Notifications tab. This tab displays a list of all notifications configured for this component.
4. Click **New**.
5. Select the type of notification you want to configure, either **Email Alert** or **SNMP Alert**.
6. Using the field descriptions provided below for guidance, complete the fields as needed.

> Built-in variables are available to pass data about the Agent, Connector or Core Processor into the notification. See Agent Variables, Connector Variables, or Core Processor Variables as appropriate.

7. Click the **Submit** button to save the record.
8. If appropriate, repeat these steps for any additional notifications you want to add.

Email Notifications

Shown below is the Email Alert screen for a Connector. The screen is the same for all components.



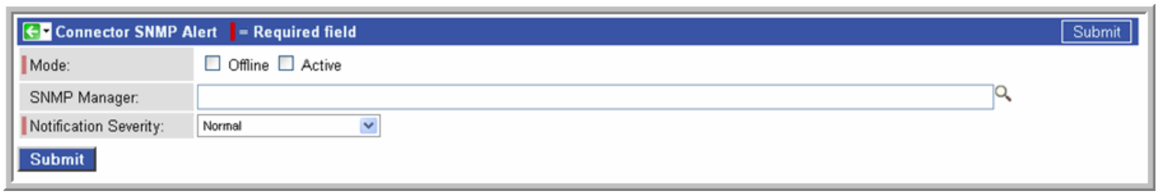
The table below describes the fields and buttons on the Email Alert screen.

Field Name	Description
Mode	- Offline = Trigger the notification when the component goes offline. - Active = Trigger the notification when the component comes up.
Email Template	Optional. The name of the Email template defined using the Email template screen. The Email template allows you to specify standard recipients and text for outgoing emails. Type in a name, or click the magnifying glass to browse to an existing Email template or create a new one. You must specify either an Email template or Email connection, or both. If you specify both, the Email server specified in the Email Connection record overrides the server in the template.
Email Connection	Required. The name of the Email connection defined using the Email connection screen. The email connection specifies information about the email server. You can also specify the Email connection in the Email template (see above). You must specify either an Email template or Email connection, or both. If you specify an Email template and an Email connection, the server selected in the Email connection overrides the server selected in the Email template. Type in a name, or click the magnifying glass to browse to an existing Email server definition or create a new one.
To	Required. Specifies the email address of the recipient. Use commas to separate multiple recipients. Variables supported.
CC	Optional. Specifies the email address of the party being sent a copy of the email, if any. Use commas to separate multiple recipients. Variables supported.
BCC	Optional. Specifies the email address of the party being sent a blind (hidden) copy of the email, if any. Use commas to separate multiple recipients. Variables supported.
Subject	Optional. Specifies the subject line of the email. Variables supported.

Body	Optional. Contains the text of the email message. Variables supported. If both the email template and the email task contain text in the body, the text is appended. Specific built-in variables are available for passing information about the agent, cluster node (server), or connector. You must use the appropriate variables for each component type; that is, use ops_agent variables for agent notifications, use ops_cluster variables for cluster notifications, and ops_connector variables for connector notifications. See Built-In Variables.
Reply-To	Required. Specifies the email address of the sender. Use commas to separate multiple recipients. Variables supported.
Submit button	Submits the new record to the database.
Update button	Saves updates to the record.
Delete button	Deletes the current record.

SNMP Notifications

Shown below is the SNMP Alert screen. The screen is the same for all components.



Connector SNMP Alert = Required field

Mode: ☐ Offline ☐ Active

SNMP Manager:

Notification Severity: Normal

The table below describes the fields and buttons on the SNMP Alert screen.

Field Name	Description
Mode	- Offline = Trigger the notification when the component goes offline. - Active = Trigger the notification when the component comes up.
SNMP Manager	The SNMP Manager that will be used to generate the notification.
Notification Severity	Optional. Informational only. Indicates the severity of this notification. Options: Normal, Warning, Minor, Major Critical.
Submit button	Submits the new record to the database.
Update button	Saves updates to the record.
Delete button	Deletes the current record.

zOS Resource

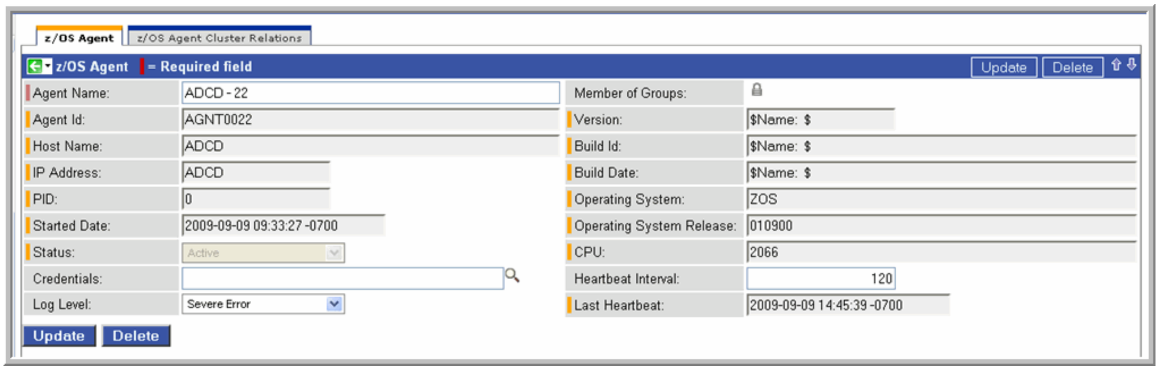
Overview

The z/OS agent resource provides information about a z/OS agent running on a z/OS platform. To run a z/OS task, you need an Opswise z/OS agent installed and running on the target machine.

z/OS Agent Definition

The resource definition provides the information necessary for the scheduler to locate and communicate with the machine where the Opswise agent resides. Opswise creates this record automatically when the agent connects with the core processor.

To view an agent resource, select the type you want to view from the Resources section of the navigation pane. A list displays showing all the registered agents of the type you selected. Shown below is a z/OS agent definition.



Agent Name:	ADCD - 22	Member of Groups:	
Agent Id:	AGNT0022	Version:	$Name: $
Host Name:	ADCD	Build Id:	$Name: $
IP Address:	ADCD	Build Date:	$Name: $
PID:	0	Operating System:	ZOS
Started Date:	2009-09-09 09:33:27 -0700	Operating System Release:	010900
Status:	Active	CPU:	2066
Credentials:		Heartbeat Interval:	120
Log Level:	Severe Error	Last Heartbeat:	2009-09-09 14:45:39 -0700

You can change any of the following details on the record by entering the change and clicking **Update**:

- Heartbeat interval
- Log level
- Credentials

See the field descriptions below for details about those fields.

z/OS Resource Field Descriptions

The table below describes the fields, buttons, and tabs on the z/OS Agent resource screen.

Field Name	Description
Agent Name-Agent Resource	Required. Defined by the user when installing the agent. This is the name used within Opswise Automation Center to identify this resource.
Member of Groups-Agent	Linux/Unix only. Must be specified at installation. Assigns this agent to a Unix group ID.
Agent ID	Unique ID for this agent, created during installation. The name of the communications server message queue. Opswise Automation Center uses the communications server for communications between the core scheduler and Opswise Automation Center agents.

Version	System-supplied. Version number of the agent program
Host Name	Specified by the user during installation. The IP address or domain/name of the host machine where the component resides.
Build ID	System-supplied, provided by the agent. The build ID of the Opswise Automation Center agent. Internal use only.
IP Address	Provided by the user during installation. The TCP/IP address of the machine where the agent is running.
Build Date	System-supplied, provided by the agent. The date the agent program was last built.
PID	System-supplied, provided by the agent. Process ID of the agent.
Operating System	System-supplied. The operating system where the agent is running.
Started Date	System-supplied. The date/timestamp when the agent was last started.
Operating System Release	System-supplied. Release information for the operating system where the agent is running.
Status-Agent	System-supplied. The status of the agent.
CPU	System-supplied. Information about the CPU on the agent machine.
Credentials	Login credentials Opswise Automation Center will use to access the remote machine. They must be provided either in the resource definition or task definition, or both. Credentials provided in the task definition override any credentials provided in the agent resource definition. See Credentials.
Heartbeat Interval	User-modifiable. The heartbeat interval in seconds. The heartbeat is a status message sent from the agent to the core processor.
Log Level	User-modifiable. The level of logging the agent should perform. Options: • Severe Error • Errors • Warning • Informational • Debug • Trace
Last Heartbeat	System-supplied. The date and time the most recent heartbeat was received from the component.
Update button	Saves updates to the record.
Delete button	Deletes the current record.

Connector Notifications tab	System-supplied. Displays a list of connector notifications that have been defined for this component.
Task Instances tab	System-supplied. Displays a list of all instances that have run or are ready to run on this agent since it last started.

Indesca Resource

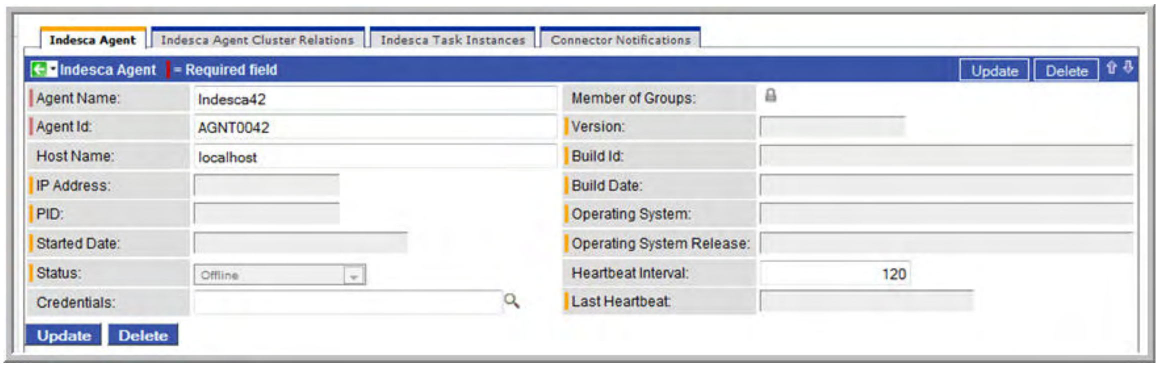
Overview

The Indesca agent resource provides information about an Indesca agent. To run an Indesca task, you need an Indesca agent installed and running on the target machine.

Indesca Agent Definition

The resource definition provides the information necessary for the scheduler to locate and communicate with the machine where the agent resides. Opswise creates this record automatically when the agent connects with the core processor.

To view an agent resource, select the type you want to view from the Resources section of the navigation pane. A list displays showing all the registered agents of the type you selected. Shown below is an Indesca agent definition.



Agent Name:	Indesca42	Member of Groups:	
Agent Id:	AGNT0042	Version:	
Host Name:	localhost	Build Id:	
IP Address:		Build Date:	
PID:		Operating System:	
Started Date:		Operating System Release:	
Status:	Offline	Heartbeat Interval:	120
Credentials:		Last Heartbeat:	

You can change the following details on the record by entering the change and clicking **Update**:

- Heartbeat Interval
- Credentials

See the field descriptions below for details about those fields.

Indesca Resource Field Descriptions

The table below describes the fields, buttons, and tabs on the Indesca Agent resource screen.

Field Name	Description
Agent Name	Required. Defined by the user when installing the agent. This is the name used within Opswise Automation Center to identify this resource.
Member of Groups	Linux/Unix only. Must be specified at installation. Assigns this agent to a Unix group ID.
Agent ID	Unique ID for this agent, created during installation. The name of the communications server message queue. Opswise Automation Center uses the communications server for communications between the core scheduler and Opswise Automation Center agents.
Version	System-supplied. Version number of the agent program

Host Name	Specified by the user during installation. The IP address or domain/name of the host machine where the component resides.
Build ID	System-supplied, provided by the agent. The build ID of the Opswise Automation Center agent. Internal use only.
IP Address	Provided by the user during installation. The TCP/IP address of the machine where the agent is running.
Build Date	System-supplied, provided by the agent. The date the agent program was last built.
PID	System-supplied, provided by the agent. Process ID of the agent.
Operating System	System-supplied. The operating system where the agent is running.
Started Date	System-supplied. The date/timestamp when the agent was last started.
Operating System Release	System-supplied. Release information for the operating system where the agent is running.
Status-Agent	System-supplied. The status of the agent.
Heartbeat Interval	User-modifiable. The heartbeat interval in seconds. The heartbeat is a status message sent from the agent to the core processor.
Credentials	Login credentials Opswise Automation Center will use to access the remote machine. They must be provided either in the resource definition or task definition, or both. Credentials provided in the task definition override any credentials provided in the agent resource definition. See Credentials.
Last Heartbeat	System-supplied. The date and time the most recent heartbeat was received from the component.
Update button	Saves updates to the record.
Delete button	Deletes the current record.
Agent Cluster Relations tab	Provides a list of agent clusters that this agent belongs to, if any. See Configuring Agent Clusters.
Task Instances tab	System-supplied. Displays a list of all instances that have run or are ready to run on this agent since it last started.
Connector Notifications tab	System-supplied. Displays a list of connector notifications that have been defined for this component.

Windows Resource

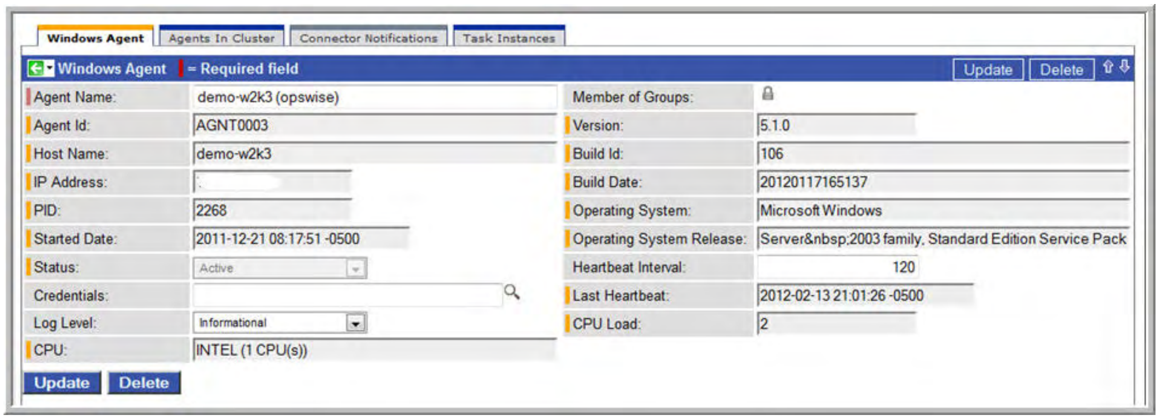
Overview

The Windows agent resource provides information about a Windows agent running on a Windows platform. To run a Windows task, you need a Windows agent installed and running on the target machine.

Windows Agent Definition

The resource definition provides the information necessary for the scheduler to locate and communicate with the machine where the agent resides. Opswise creates this record automatically when the agent connects with the core processor.

To view an agent resource, select the type you want to view from the Resources section of the navigation pane. A list displays showing all the registered agents of the type you selected. Shown below is a Windows agent definition.



Agent Name:	demo-w2k3 (opswise)	Member of Groups:	
Agent Id:	AGNT0003	Version:	5.1.0
Host Name:	demo-w2k3	Build Id:	106
IP Address:	[redacted]	Build Date:	20120117165137
PID:	2268	Operating System:	Microsoft Windows
Started Date:	2011-12-21 08:17:51 -0500	Operating System Release:	Server 2003 family, Standard Edition Service Pack
Status:	Active	Heartbeat Interval:	120
Credentials:		Last Heartbeat:	2012-02-13 21:01:26 -0500
Log Level:	Informational	CPU Load:	2
CPU:	INTEL (1 CPU(s))		

You can change any of the following details on the record by entering the change and clicking **Update**:

- Heartbeat interval
- Log level
- Credentials

See the field descriptions below for details about those fields.

Windows Resource Field Descriptions

The table below describes the fields, buttons, and tabs on the Windows Agent resource screen.

Field Name	Description
Agent Name	Required. Defined by the user when installing the agent. This is the name used within Opswise Automation Center to identify this resource.
Member of Groups	Optional. User-defined at installation. Allows you to specify one or more Opswise groups that this resource definition belongs to.
Agent ID	Unique ID for this agent, created during installation. The name of the communications server message queue. Opswise Automation Center uses the communications server for communications between the core scheduler and Opswise Automation Center agents.

Version	System-supplied. Version number of the agent program
Host Name	Specified by the user during installation. The IP address or domain/name of the host machine where the component resides.
Build ID	System-supplied, provided by the agent. The build ID of the Opswise Automation Center agent. Internal use only.
IP Address	Provided by the user during installation. The TCP/IP address of the machine where the agent is running.
Build Date	System-supplied, provided by the agent. The date the agent program was last built.
PID	System-supplied, provided by the agent. Process ID of the agent.
Operating System	System-supplied. The operating system where the agent is running.
Started Date	System-supplied. The date/timestamp when the agent was last started.
Operating System Release	System-supplied. Release information for the operating system where the agent is running.
Status	System-supplied. The status of the agent.
Heartbeat Interval	User-modifiable. The heartbeat interval in seconds. The heartbeat is a status message sent from the agent to the core processor.
Credentials	Login credentials Opswise Automation Center will use to access the remote machine. They must be provided either in the resource definition or task definition, or both. Credentials provided in the task definition override any credentials provided in the agent resource definition. See Credentials.
Last Heartbeat	System-supplied. The date and time the most recent heartbeat was received from the component.
Log Level	User-modifiable. The level of logging the agent should perform. Options: • Severe Error • Errors • Warning • Informational • Debug • Trace
CPU Load	System-supplied. The current CPU load on the agent machine, expressed as a percentage. For example, 1 means 1% currently utilized.
CPU	System-supplied. Information about the CPU on the agent machine.

Update button	Saves updates to the record.
Delete button	Deletes the current record.
Agents in Cluster tab	Provides a list of agent clusters that this agent belongs to, if any. See Configuring Agent Clusters.
Task Instances tab	System-supplied. Displays a list of all instances that have run or are ready to run on this agent since it last started.
Connector Notifications tab	System-supplied. Displays a list of connector notifications that have been defined for this component.

Linux Unix Resource

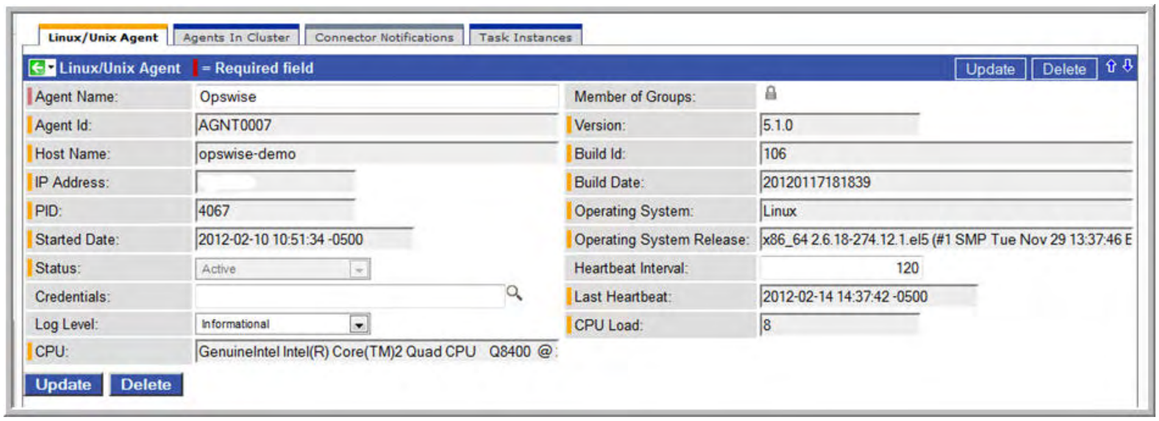
Overview

The Linux/Unix agent resource provides information about a Linux or Unix agent running on a Linux or Unix platform. To run a Linux/Unix task, you need an Opswise Linux/Unix agent installed and running on the target machine.

Linux/Unix Agent Definition

The resource definition provides the information necessary for the scheduler to locate and communicate with the machine where the Opswise agent resides. Opswise creates this record automatically when the agent connects with the core processor.

To view an agent resource, select the type you want to view from the Resources section of the navigation pane. A list displays showing all the registered agents of the type you selected. Shown below is a Linux/Unix agent definition.



Agent Name:	Opswise	Member of Groups:	
Agent Id:	AGNT0007	Version:	5.1.0
Host Name:	opswise-demo	Build Id:	106
IP Address:		Build Date:	20120117181839
PID:	4067	Operating System:	Linux
Started Date:	2012-02-10 10:51:34 -0500	Operating System Release:	x86_64 2.6.18-274.12.1.el5 (#1 SMP Tue Nov 29 13:37:46 E
Status:	Active	Heartbeat Interval:	120
Credentials:		Last Heartbeat:	2012-02-14 14:37:42 -0500
Log Level:	Informational	CPU Load:	8
CPU:	GenuineIntel Intel(R) Core(TM)2 Quad CPU Q8400 @		

You can change any of the following details on the record by entering the change and clicking **Update**:

- Heartbeat interval
- Log level
- Credentials

See the field descriptions below for details about those fields.

Linux/Unix Resource Field Descriptions

The table below describes the fields, buttons, and tabs on the Linux/Unix Agent resource screen.

Field Name	Description
Agent Name	Required. Defined by the user when installing the agent. This is the name used within Opswise Automation Center to identify this resource.
Member of Groups	Linux/Unix only. Must be specified at installation. Assigns this agent to a Unix group ID.
Agent ID	Unique ID for this agent, created during installation. The name of the communications server message queue. Opswise Automation Center uses the communications server for communications between the core scheduler and Opswise Automation Center agents.

Version	System-supplied. Version number of the agent program
Host Name	Specified by the user during installation. The IP address or domain/name of the host machine where the component resides.
Build ID	System-supplied, provided by the agent. The build ID of the Opswise Automation Center agent. Internal use only.
IP Address	Provided by the user during installation. The TCP/IP address of the machine where the agent is running.
Build Date	System-supplied, provided by the agent. The date the agent program was last built.
PID	System-supplied, provided by the agent. Process ID of the agent.
Operating System	System-supplied. The operating system where the agent is running.
Started Date	System-supplied. The date/timestamp when the agent was last started.
Operating System Release	System-supplied. Release information for the operating system where the agent is running.
Status	System-supplied. The status of the agent.
Heartbeat Interval	User-modifiable. The heartbeat interval in seconds. The heartbeat is a status message sent from the agent to the core processor.
Credentials	Login credentials Opswise Automation Center will use to access the remote machine. They must be provided either in the resource definition or task definition, or both. Credentials provided in the task definition override any credentials provided in the agent resource definition. See Credentials.
Last Heartbeat	System-supplied. The date and time the most recent heartbeat was received from the component.
Log Level	User-modifiable. The level of logging the agent should perform. Options: • Severe Error • Errors • Warning • Informational • Debug • Trace
CPU Load	System-supplied. The current CPU load on the agent machine, expressed as a percentage. For example, 1 means 1% currently utilized.
CPU	System-supplied. Information about the CPU on the agent machine.

Update button	Saves updates to the record.
Delete button	Deletes the current record.
Agents in Cluster tab	Provides a list of agent clusters that this agent belongs to, if any. See Configuring Agent Clusters.
Connector Notifications tab	System-supplied. Displays a list of connector notifications that have been defined for this component.
Task Instances tab	System-supplied. Displays a list of all instances that have run or are ready to run on this agent since it last started.

Configuring Agentless Resources

Agentless resources include database, email, SNMP, and SAP connections.

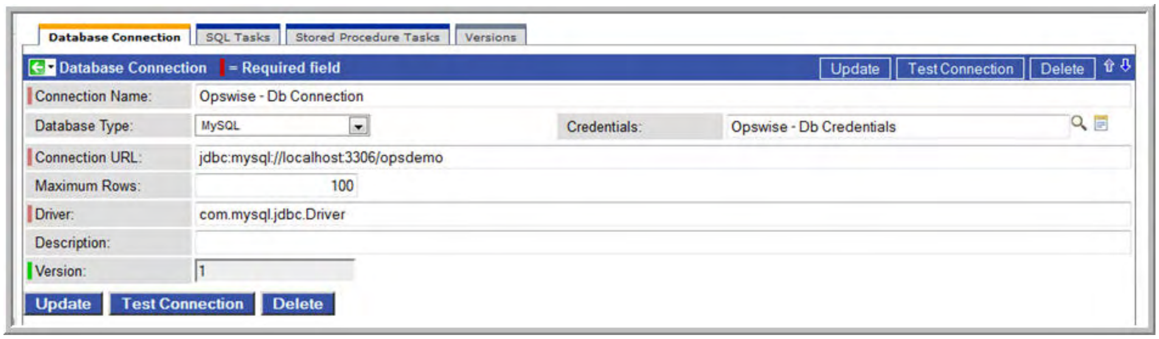
Database Connection

Overview

The Database Connection provides all the database server information necessary for Opswise to execute an SQL task or a Stored Procedure Task.

Creating a New Database Connection

1. From the Resources section of the Navigation Pane, select **Database Connections**.
2. From the Database Connections wizard, select **New**. Opswise displays the Database connection screen, shown in the example below:



Connection Name: Opswise - Db Connection
Database Type: MySQL
Credentials: Opswise - Db Credentials
Connection URL: jdbc:mysql://localhost:3306/opsdemo
Maximum Rows: 100
Driver: com.mysql.jdbc.Driver
Description:
Version: 1

3. Using the field descriptions provided below as a guide, complete the fields as needed.
4. Right-click and select **Save** to save the record and remain on the current display.
5. To test the connection, click **Test Connection**.
6. Click the **Submit** button to save the record.
7. If appropriate, repeat these steps for any additional Database Connections you want to add.

Database Connection Field Descriptions

The table below describes the fields, buttons, and tabs on the Database connection screen.

Field Name	Description
Connection Name	Required. Name used within Opswise to identify this resource. Up to 40 alphanumerics.It is the user's responsibility to develop a workable naming scheme for resources.
Database Type	Required. The type of database. Options: • MySQL • MS SQL Server • Oracle • DB2 • Sybase SQL Anywhere • Other
Credentials	Login credentials Opswise Automation Center will use to access the remote machine. They must be provided either in the resource definition or task definition, or both. Credentials provided in the task definition override any credentials provided in the agent resource definition. See Credentials.

Connection URL	The URL of the database.
	Note: If you are using a MySQL database and want the ability to issue multiple SQL commands from a single task, you need to enable this by appending the following string to the end of the connection string:
	`?allowMultiQueries=true`
	For example:
	`jdbc:mysql://localhost:3306/opswise?allowMultiQueries=true`

Maximum Rows	Optional. If necessary, specify a limit to the number of rows you want returned by the SQL statement. This value overrides any value you specify in the database connection.
Driver	The name of the JDBC driver.
Description	Optional. Description of the database.
Version	Task definition only; system-supplied. The version number of the current record, which is incremented by the system every time a user updates a record. Click on the Versions tab to view previous versions. For details, see Record Versioning.
Submit button	Submits the new record to the database.
Update button	Saves updates to the record.
Test Connection button	After saving the record to the database, click **Test Connection** to run a connection test.
Delete button	Deletes the current record.
SQL and Stored Procedure Tasks tabs	A separate tab is provided for each task type that uses a database (SQL and Stored Procedures). Each tab displays a list of tasks that are using this database connection.
Versions tab	Stores copies of all previous versions of the current record (see Record Versioning).

Email Connection

Overview

Email connections are used two ways within Opswise:

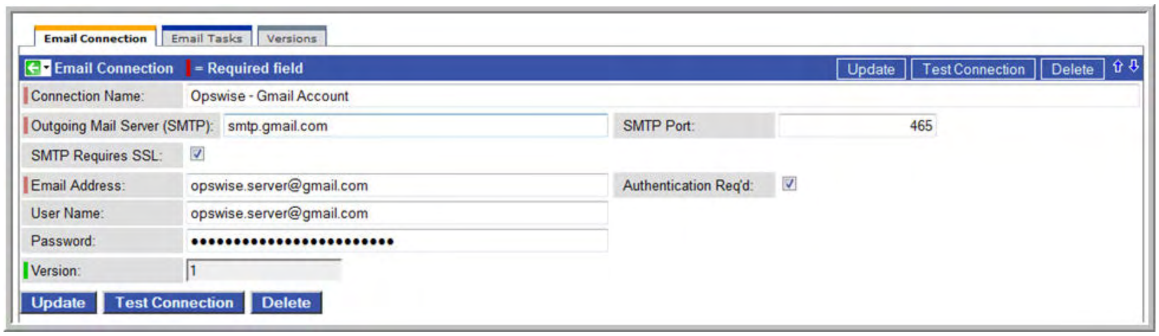
- The Email Task uses the Email connection to generate emails independent of tasks. See Email Task.
- The Email Notification uses the Email connection to generate notifications related to tasks. See Email Notifications.

> ⚠ **Note**
> The Email connections described here are not used for emailing reports. See Reports - Setting Up Email Properties.

Creating a New Email Connection

The Email connection provides all the email server information necessary for the system to send an email using an Email task or notification.

1. From the navigation pane, select **Automation Center Resources > Email Connections**. The Email Connection List screen displays.
2. Click **New**. The Email Connection Definition screen displays:



Connection Name:	Opswise - Gmail Account		
Outgoing Mail Server (SMTP):	smtp.gmail.com	SMTP Port:	465
SMTP Requires SSL:	☑		
Email Address:	opswise.server@gmail.com	Authentication Req'd:	☑
User Name:	opswise.server@gmail.com		
Password:	●●●●●●●●●●●●●●●●●●●●●●●●		
Version:	1		

3. Using the field descriptions provided below as a guide, complete the fields as needed.
4. Right-click and select **Save** to save the record and remain on the current display.
5. To test the connection, click **Test Connection**.
6. Click the **Submit** button to save the record.
7. If appropriate, repeat these steps for any additional Email connections you want to add.

Email Connection Field Descriptions

The table below describes the fields, buttons, and tabs on the Email connection screen.

Field Name	Description
Resource Name	Required. Name used within Opswise to identify this resource. Up to 40 alphanumerics.It is the user's responsibility to develop a workable naming scheme for resources.
Outgoing Mail Server (SMTP)	Required. The name or IP address of the outgoing email server.
SMTP Port	Port number on the machine where the email server resides. Any port number between 1 and 65535.

SMTP Requires SSL	Specifies whether your SMTP server requires SSL.
Email Address	Required. The email address of the sender.
Authentication Req'd	If enabled, the user name and password (below) are required.
User Name	Name Opswise will use to connect to the server.
Password	Password Opswise will use to connect to the server.
Version	Task definition only; system-supplied. The version number of the current record, which is incremented by the system every time a user updates a record. Click on the Versions tab to view previous versions. For details, see Record Versioning.
Submit button	Submits the new record to the database.
Update button	Saves updates to the record.
Test Connection button	After saving the record to the database, click **Test Connection** to run a connection test.
Delete button	Deletes the current record.
Email Tasks tab	Provides a list of tasks that use this email server.
Versions tab	Stores copies of all previous versions of the current record (see Record Versioning).

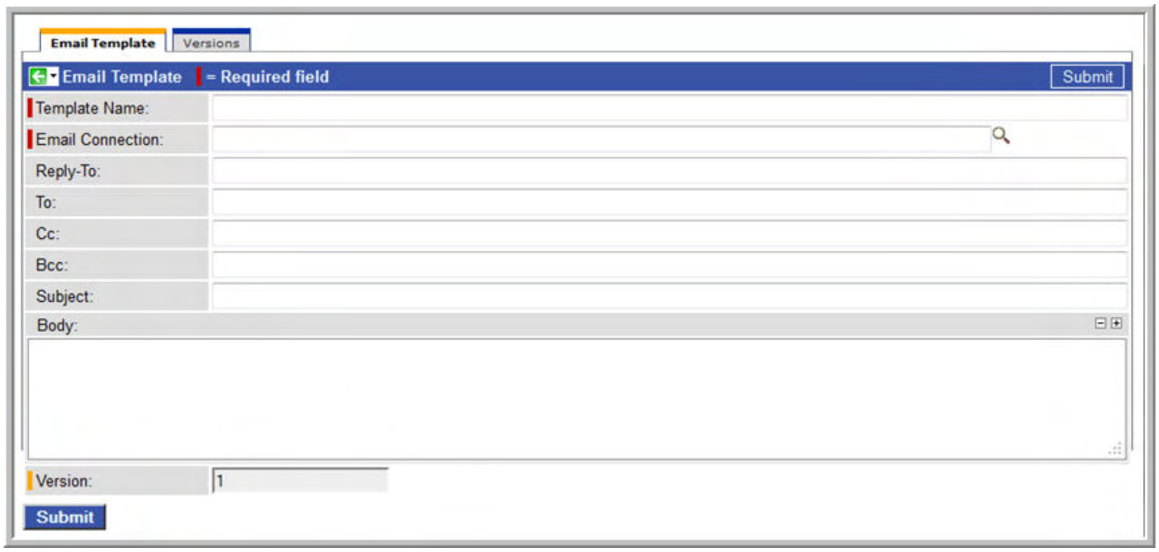
Email Template

Overview

The Email template allows you to construct commonly-used information that can be copied to create Email tasks. If an Email task specifies a template, the system uses the information in the template to construct and execute the Email task. Any information specified in the task overrides what is specified in the template.

Creating a New Email Template

1. From the navigation pane, select **Automation Center Resources > Email Templates**. The Email Templates List screen displays.
2. Click **New**. The Email Template Definition screen displays.



Version: 1

3. Using the field descriptions provided below as a guide, complete the fields as needed.
4. Click the **Submit** button to save the record and return to the menu, or, right-click on the title bar and select **Save** to save the record and remain on the current display.
5. If appropriate, repeat these steps for any additional templates you want to add.

Email Template Field Descriptions

The table below describes the fields, buttons, and tabs on the Email template screen.

Field Name	Description
Task Name	Required. Name used within Opswise to identify this task. Up to 40 alphanumerics; variables supported. It is the user's responsibility to develop a workable naming scheme for tasks.
Email Connection	Required. The name of the Email connection defined using the Email connection screen. The email connection specifies information about the email server. You can also specify the Email connection in the Email template (see above). You must specify either an Email template or Email connection, or both. If you specify an Email template and an Email connection, the server selected in the Email connection overrides the server selected in the Email template. Type in a name, or click the magnifying glass to browse to an existing Email server definition or create a new one.

Reply-To	Required. Specifies the email address of the sender. Use commas to separate multiple recipients. Variables supported.
To	Required. Specifies the email address of the recipient. Use commas to separate multiple recipients. Variables supported.
CC	Optional. Specifies the email address of the party being sent a copy of the email, if any. Use commas to separate multiple recipients. Variables supported.
BCC	Optional. Specifies the email address of the party being sent a blind (hidden) copy of the email, if any. Use commas to separate multiple recipients. Variables supported.
Subject	Optional. Specifies the subject line of the email. Variables supported.
Body	Optional. Contains the text of the email message. Variables supported. If both the email template and the email task contain text in the body, the text is appended.
Version	Task definition only; system-supplied. The version number of the current record, which is incremented by the system every time a user updates a record. Click on the Versions tab to view previous versions. For details, see Record Versioning.
Submit button	Submits the new record to the database.
Update button	Saves updates to the record.
Delete button	Deletes the current record.
Versions tab	Stores copies of all previous versions of the current record (see Record Versioning).

SNMP Manager

Overview

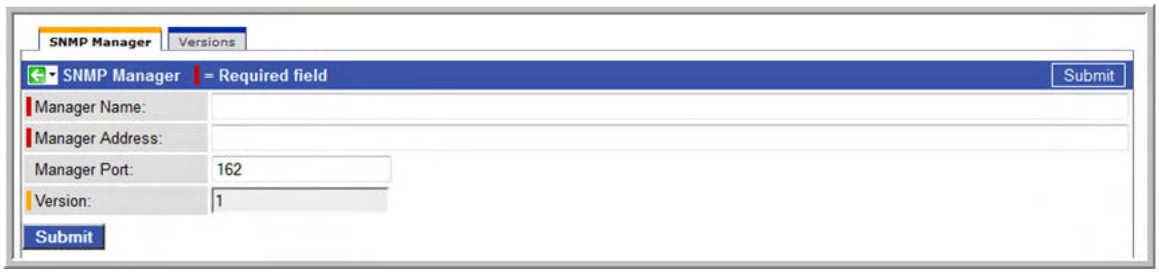
SNMP managers are used to generate SNMP notifications as follows:

- When a Universal Automation Center Agent (UAG) or outboard component (transporter or hub) goes down or comes back up. See Sending Notifications on Opswise Component Status.
- When the SNMP manager is being used to generate notifications associated with a task. See Creating SNMP Notifications.

Creating a New SNMP Manager

The SNMP manager provides all the information necessary for the system to send an SNMP message using SNMP Notifications.

1. Form the navigation pane, select **Automation Center Resources > SNMP Managers**. The SNMP Managers List screen displays.
2. Click **New**. The SNMP Manager Definition screen displays:



Manager Name:	
Manager Address:	
Manager Port:	162
Version:	1

3. Using the field descriptions provided below as a guide, complete the fields as needed.
4. Click **Submit** to save the record.
5. If appropriate, repeat these steps for any additional SNMP managers you want to add.

SNMP Manager Field Descriptions

The table below describes the fields and buttons on the SNMP Manager screen.

Field Name	Description
Manager Name	Required. Name used within Opswise to identify this resource. Up to 40 alphanumerics.It is the user's responsibility to develop a workable naming scheme for resources.
Manager Address	Required. The name or IP address of the SNMP manager.
Manager Port	Port number used by the SNMP manager. Any port number between 1 and 65535.
Version	Task definition only; system-supplied. The version number of the current record, which is incremented by the system every time a user updates a record. Click on the Versions tab to view previous versions. For details, see Record Versioning.
Submit button	Submits the new record to the database.

Update button	Saves updates to the record.
Delete button	Deletes the current record.
Versions tab	Stores copies of all previous versions of the current record (see Record Versioning).

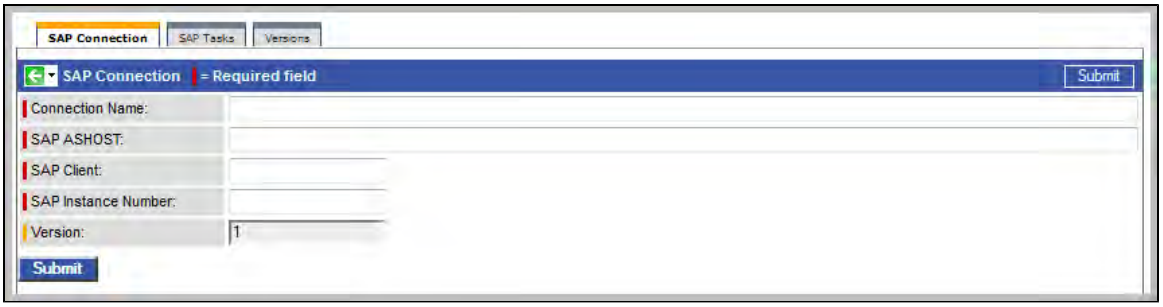
SAP Connection

Overview

The SAP Connection provides all the SAP server information necessary for Opswise to execute an SAP task on an SAP system. These instructions assume the user is familiar with SAP.

Creating a New SAP Connection

1. From the navigation pane, select **Automation Center Resources > SAP Connections**. The SAP Connections List screen displays.
2. Click **New**. The SAP Connection Definition screen displays.



Version: 1

3. Using the field descriptions provided below as a guide, complete the fields as needed.
4. Right-click and select **Save** to save the record and remain on the current display.
5. Click the **Submit** button to save the record.
6. If appropriate, repeat these steps for any additional SAP connections you want to add.

SAP Connection Field Descriptions

The table below describes the fields, buttons, and tabs on the SAP Connection screen.

Field Name	Description
Connection Name	Required. Name used within Opswise to identify this resource. Up to 40 alphanumerics.It is the user's responsibility to develop a workable naming scheme for resources.
SAP ASHOST	Host name of an SAP application server. If the path to the server goes through SAP routers, prefix the host name with the SAP router string.
SAP Client	SAP Client number.
SAP Instance Number	SAP instance number.
Version	Task definition only; system-supplied. The version number of the current record, which is incremented by the system every time a user updates a record. Click on the Versions tab to view previous versions. For details, see Record Versioning.

Submit button	Submits the new record to the database.
Update button	Saves updates to the record.
Delete button	Deletes the current record.
SAP Tasks tab	Provides a list of SAP tasks that are using this SAP connection.
Versions tab	Stores copies of all previous versions of the current record (see Record Versioning).

Configuring Agent Clusters

Overview

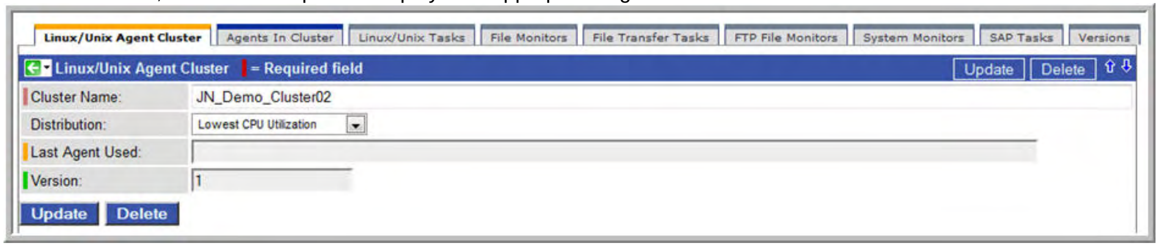
Agent Clusters allow you to configure a cluster of agents and a selection method, which you can then specify for use by a task. When you specify an Agent Cluster in a task, Opswise selects the best agent from the cluster, based on the selection method specified. If you specify both an agent and an agent cluster in a task, Opswise first attempts to run the task on the agent; if the agent is unavailable, the system selects the best agent from the agent cluster.

Agent Clusters are supported for Windows and Linux/Unix agents. The screens are the same for all platforms.

Creating a New Agent Cluster

The instructions and screens for creating Windows Agent Clusters and Linux/Unix Agent Clusters are the same.

1. From the Navigation Pane, select the appropriate Agent Clusters option (Windows or Linux/Unix).
2. From the wizard, select **New**. Opswise displays the appropriate Agent Clusters screen:

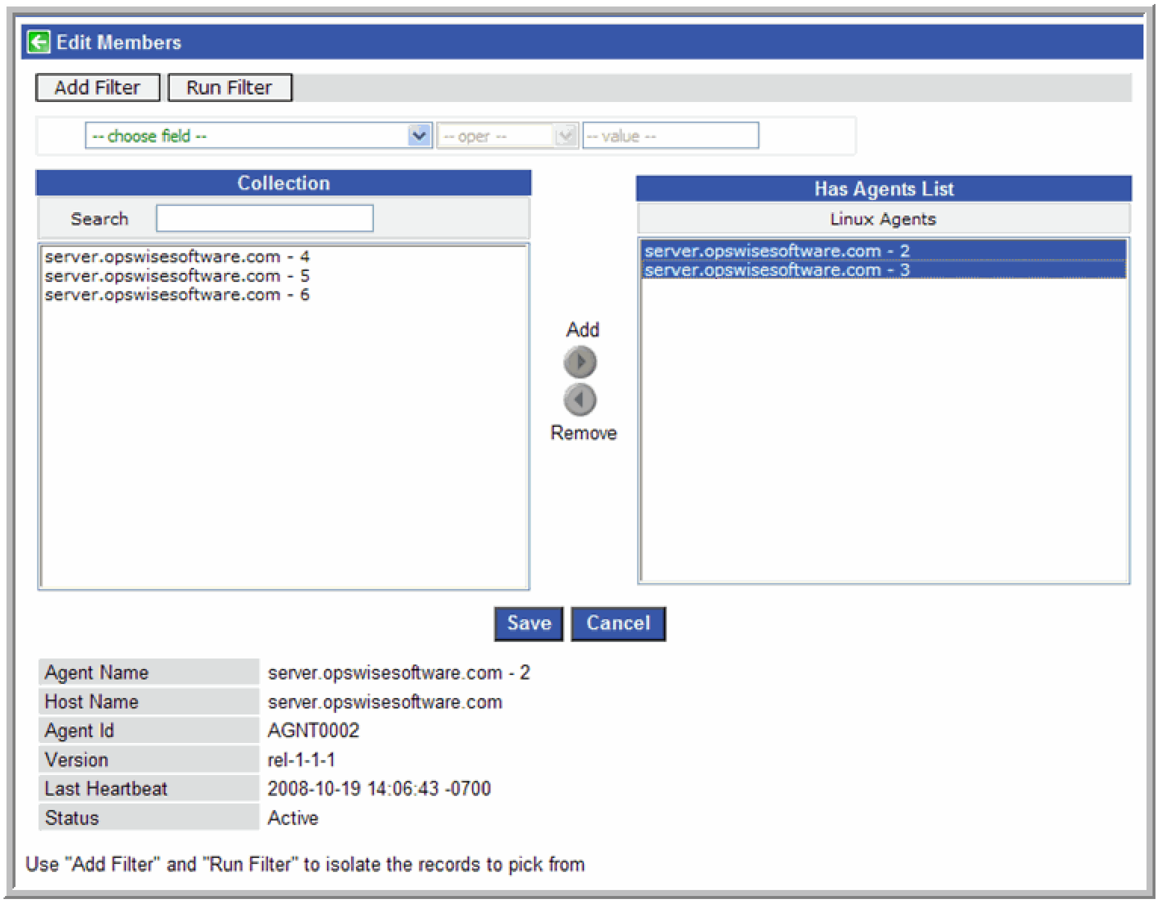


Cluster Name:	JN_Demo_Cluster02
Distribution:	Lowest CPU Utilization
Last Agent Used:	
Version:	1

3. Using the field descriptions provided below as a guide, complete the fields as needed.
4. Right-click on the title bar and select **Save** to save the record and remain on the current display.
5. To add agents to this cluster, see Assigning Agents to the Cluster.
6. If appropriate, repeat these steps for any additional agent clusters you want to add.

Assigning Agents to the Cluster

1. Open the cluster to which you want to assign one or more existing agents.
2. Click on the **Agent Cluster Relations** tab or scroll down to that section.
3. Click the **Edit** button. The Edit Members form displays:



server.opswisesoftware.com - 4
server.opswisesoftware.com - 5
server.opswisesoftware.com - 6

server.opswisesoftware.com - 2
server.opswisesoftware.com - 3

Agent Name	server.opswisesoftware.com - 2
Host Name	server.opswisesoftware.com
Agent Id	AGNT0002
Version	rel-1-1-1
Last Heartbeat	2008-10-19 14:06:43 -0700
Status	Active

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

4. The agents listed under Collection are existing agents of the current type that do not already belong to this cluster. The agents listed under Has Agents List are agents that belong to this cluster. You can filter the agents listed under Collection as follows:
   a. Click **Add Filter**. See To create a filter for information about how to construct a filter.
   b. When you have defined the filter you want, click **Run Filter**. The Collection list now displays only those agents that match the filter.
   c. To remove filtering, click the **X** to the right of the condition, then click **Run Filter** again.
5. To add to or remove agents from the Has Agents List:
   - To add an agent to the list, double-click on the agent name in the Collection list.
   - To remove an agent from the list, double-click on the name in the Has Agents List.
6. As you click on an agent, the system displays details about the agent at the bottom of the form.
7. When you are finished, click **Save**.

Agent Cluster Field Descriptions

The table below describes the fields, buttons, and tabs on the Agent cluster screen.

Field Name	Description
Cluster Name	Required. Name used within Opswise to identify this cluster. Up to 40 alphanumerics. It is the user's responsibility to develop a workable naming scheme for clusters.
Distribution.	The method used to select an agent. • **Any** - Select any agent in the cluster. • **Round Robin** - Select the next agent in a round robin series. • **Lowest CPU Utilization** - Selects the agent whose CPU utilization is currently the lowest.
Last Agent Used	System-supplied. Displays the agent selected the last time a task was sent to this cluster.

Version	Task definition only; system-supplied. The version number of the current record, which is incremented by the system every time a user updates a record. Click on the Versions tab to view previous versions. For details, see Record Versioning.
Submit button	Submits the new record to the database.
Update button	Saves updates to the record.
Delete button	Deletes the current record.
Agents in Cluster tab	The list of agents assigned to this cluster.
Task Type tabs	A tab is provided for each Task type that uses an agent. Each tab provides a list of tasks currently being dispatched to this agent cluster.
Versions tab	Stores copies of all previous versions of the current record (see Record Versioning).

Script Library

Overview

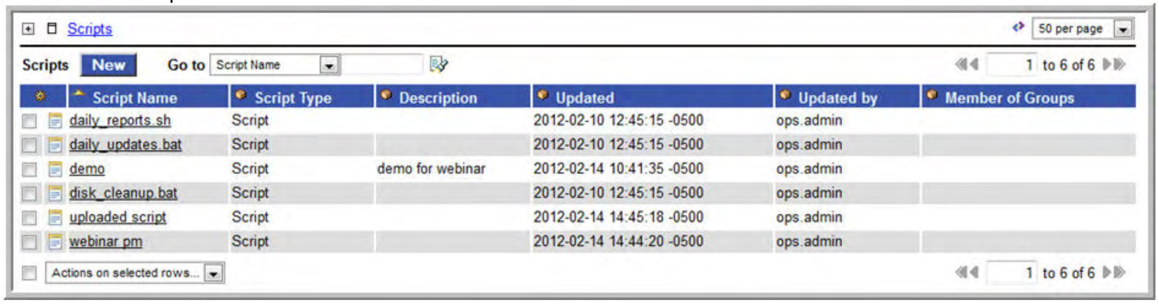
The Script Library allows you to store scripts and SAP definition files in the Opswise database for execution on a remote target. When you import a script or SAP definition file into Opswise and execute it via an Opswise task, it is transmitted to the remote machine for execution. You cannot import compiled executables into the Script Library. The content of scripts must be text that can be processed by some shell, script host or command interpreter.

You can use the Script Library with the following task types: Windows, Linux/Unix, and SAP tasks.

Click here to watch a video tutorial about the Script Library.

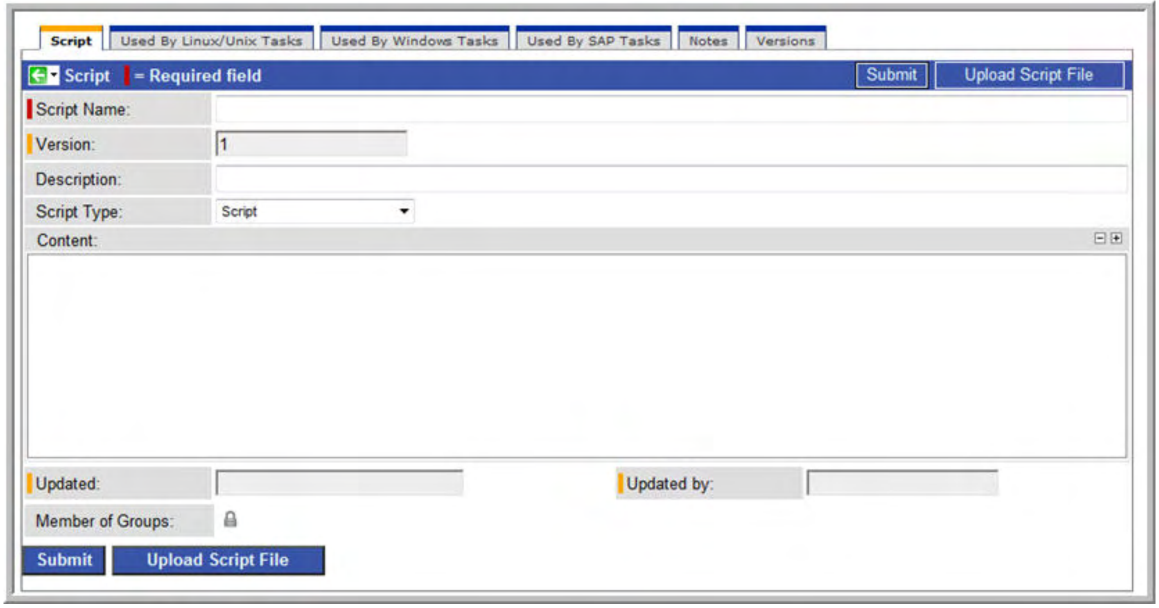
Import Procedure

1. From the navigation pane, select **Automation Center Resources > Script Library.** The system displays a list of all existing Scripts, as shown in the sample list below.



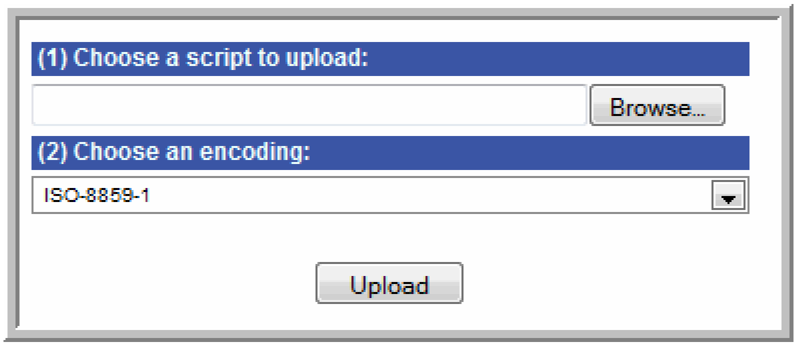
Script Name	Script Type	Description	Updated	Updated by	Member of Groups
daily_reports.sh	Script		2012-02-10 12:45:15 -0500	ops.admin	
daily_updates.bat	Script		2012-02-10 12:45:15 -0500	ops.admin	
demo	Script	demo for webinar	2012-02-14 10:41:35 -0500	ops.admin	
disk_cleanup.bat	Script		2012-02-10 12:45:15 -0500	ops.admin	
uploaded script	Script		2012-02-14 14:45:18 -0500	ops.admin	
webinar.pm	Script		2012-02-14 14:44:20 -0500	ops.admin	

2. To add a new script, click **New.** The system displays the blank Script screen, as shown below.



Script Name:	
Version:	1
Description:	
Script Type:	Script
Content:	
Updated:	
Updated by:	
Member of Groups:	

3. Using the field descriptions provided below, fill in the fields.
4. To import the script, click **Upload Script File**.



5. At the prompt, browse for and select the script you are importing.
6. In the encoding field, choose the character set of the script. It must be one of the following: ISO-8859-1, US-ASCII, UTF-8, UTF-16, UTF-16BE, UTF-16LE. To display a field tip about any of the options, click the down arrow to display the whole list, then hover over the option.
7. Click **Upload**.
8. Clicking the **Upload** button imports the script you selected and creates a new script record.

Script Library Field Descriptions

Field Name	Description
Script Name	The name of the script. This can but does not have to be the same as the name of the script file. You can but do not have to specify a file extension. The default file extension for Windows is .bat.
Version	Task definition only; system-supplied. The version number of the current record, which is incremented by the system every time a user updates a record. Click on the Versions tab to view previous versions. For details, see Record Versioning.
Description	User-supplied description of this record.
Script Type	The type of script. Options: Script (for use in Windows or Linux/Unix tasks) or SAP Definition (for use in an SAP task).
Content	The content of the script or batch file.
Updated	Date and time this record was last updated.
Updated by	The user who last updated this record.
Member of Groups	User-defined. Allows you to select one or more Opswise groups that this record definition belongs to. Click on the lock icon to unlock the field and select groups.
Submit button	Submits the new record to the database.
Update button	Saves updates to the record.
Delete button	Deletes the current record.

Upload Script File button	Uploads a script from the local file system.
Task type tabs	Each tab lists the tasks of that type that are using this script.
Notes tab	Displays all notes associated with this task. See Runbook Notes.
Versions tab	Stores copies of all previous versions of the current record (see Record Versioning).

Executing a Script Using a Task

1. Create one of the following task types: Windows, Linux/Unix, or SAP.
2. Complete the fields according to the field descriptions.
3. For Windows and Linux/Unix tasks, in the Command or Script field, select **Script**. For SAP, in the Library or File System field, select **Script Library**.
4. In the Script field, browse the Script Library and select the script you want to execute.

Creating Virtual Resources

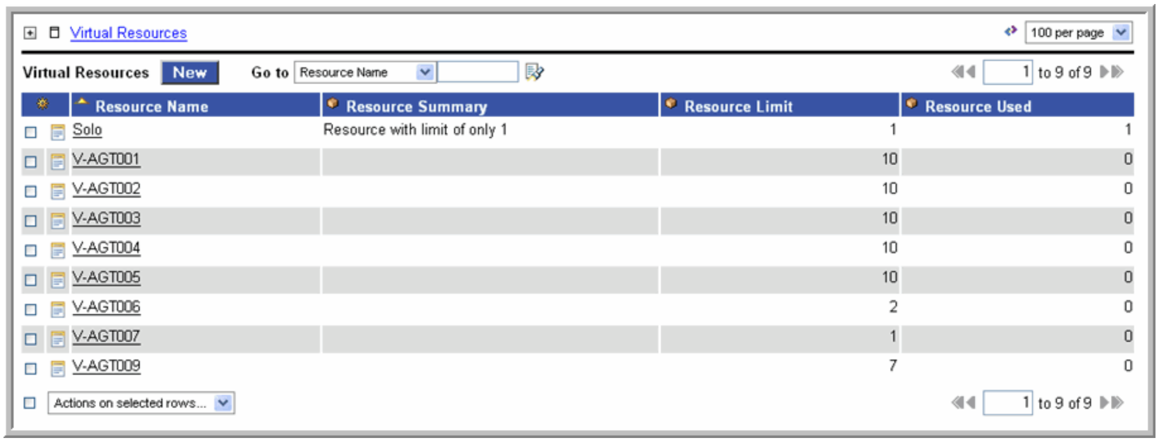
Overview

A Virtual Resource allows you to set up a "throttling" scheme that will manage how many and which tasks are sent to a particular resource at one time. Outlined below is the basic procedure and processing flow:

1. Create a virtual resource. A virtual resource normally represents an actual resource, but the virtual resource record is not actually tied to a resource record.
2. Assign the virtual resource a number. The number represents the maximum number of resource "units" that this resource can handle at any one time. If the resource has a high CPU, you will give it a higher number. For resources with limited processing power, give them a lower number. Save the Virtual Resource record(s).
3. For each Virtual Resource, identify those tasks that will be assigned to it.
4. For each of those tasks, use the **Task Virtual Resources** tab to add the Virtual Resource record to the task. Specify a number identifying how many resource units this task will consume. For example, a task that requires a small amount of processing power might consume one unit; a task that requires a high amount of resources might consume 3 units. The number of units you specify for each task is relative to the maximum number that you assign to the resource.
5. When a task with a Virtual Resource launches, the system checks the virtual resource to see if enough units are available to run the task, based on what other tasks assigned to that Virtual Resource are currently running.
    - If enough units are available, the task runs and the number of available units is decremented by the amount specified in the task. For example, if the resource has a maximum of 10 and the task uses 2, the remaining amount available on that Virtual Resource for use by other tasks is 8.
    - If there are not enough units available, the task is put into Resource Wait status and is listed in the Outstanding Requests tab in the Virtual Resource. When the required amount of resource becomes available, the task is launched.
6. Tabs on the Virtual Resource record keep track of tasks that are currently "running" on this Virtual Resource and tasks that are waiting to "run" on this Virtual Resource

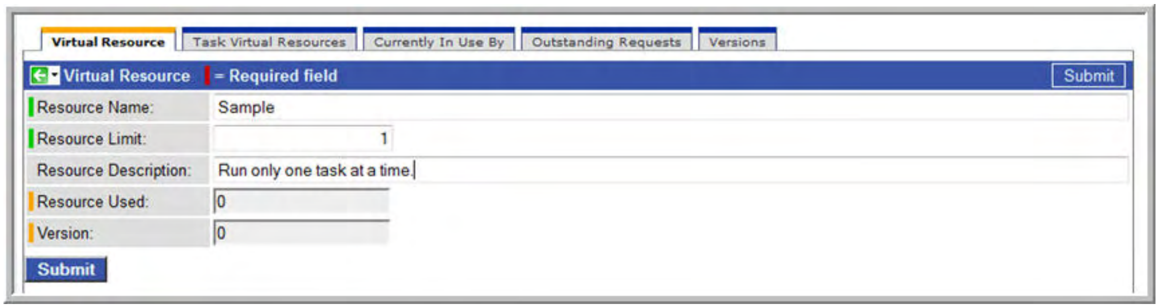
Creating a Virtual Resource

1. From the navigation pane, select **Automation Center Resources > Virtual Resources**. The Virtual Resources List screen displays:



Resource Name	Resource Summary	Resource Limit	Resource Used
Solo	Resource with limit of only 1	1	1
V-AGT001		10	0
V-AGT002		10	0
V-AGT003		10	0
V-AGT004		10	0
V-AGT005		10	0
V-AGT006		2	0
V-AGT007		1	0
V-AGT009		7	0

2. Click **New**. The Virtual Resource Definition screen displays.



Resource Name: Sample

Resource Limit: 1

Resource Description: Run only one task at a time.

Resource Used: 0

Version: 0

3. Using the field descriptions provided below, fill in the fields.
4. Click the **Submit** button to save the record and return to the menu or right-click on the title bar and select **Save** to save the record and remain on the current display.

> ⚠ **Note**
> This sample Virtual Resource Definition screen shows a Resource Limit of 1. Because each task has a minimum value of 1, this virtual resource would be limited to running only one task at a time.

Field Descriptions

The following table provides descriptions of the fields on the virtual resource form.

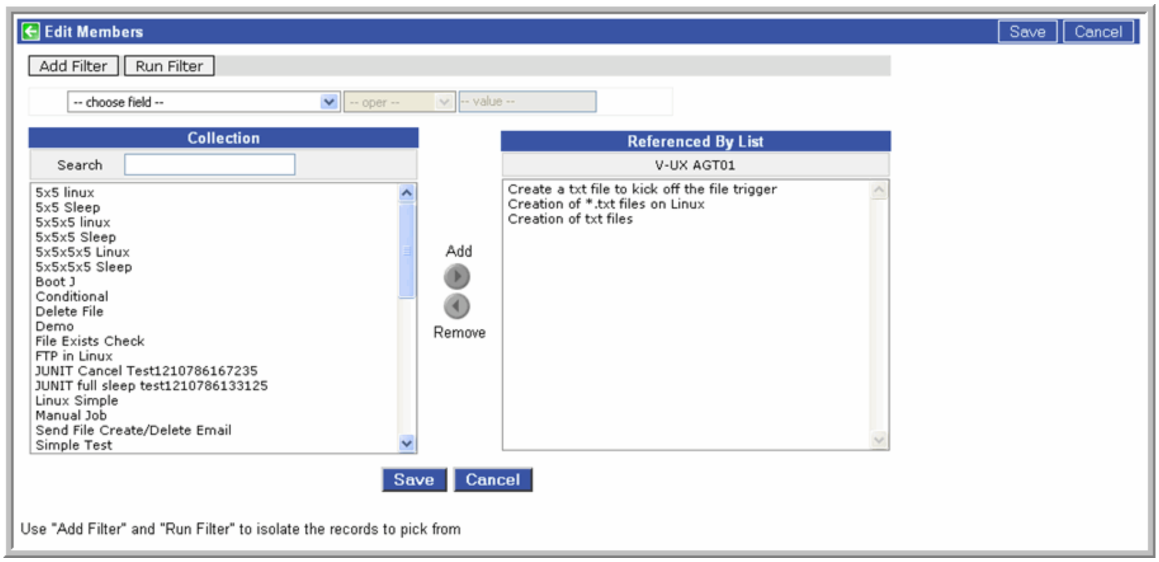
Field Name	Description
Resource Name	Required. Name used within Opswise to identify this resource. Up to 40 alphanumerics.It is the user's responsibility to develop a workable naming scheme for resources.
Resource Limit	Number of resource "units" that the resource can handle at any one time. Resources with a high amount of processing power generally have high numbers. This number is compared to the number consumed by tasks, which defaults to 1 for each task. For example, if a virtual resource has a limit of 2, then only two tasks, each consuming 1, can run on the virtual resource at a time. Or, a single task consuming 2 can run on the virtual resource at a time. If you want to make sure that only one task runs on a resource at a time, use a limit of 1 on your resource. However, any task that consumes 2 or more will not run when assigned to that resource. For information about changing the default for tasks, see Assigning Tasks to a Virtual Resource
Resource Description	Description of this virtual resource.
Resource Used	System-supplied. Shows how many units are currently in use, as of the time you opened the record.
Version	Task definition only; system-supplied. The version number of the current record, which is incremented by the system every time a user updates a record. Click on the Versions tab to view previous versions. For details, see Record Versioning.
Submit button	Submits the new record to the database.
Update button	Saves updates to the record.
Delete button	Deletes the current record.
Task Virtual Resources tab	Lists tasks that are assigned to this virtual resource.

Currently In Use By tab	Lists the tasks that are currently running on this virtual resource, at the time you opened this virtual resource record.
Outstanding Requests tab	Lists the tasks that are currently waiting to run on this virtual resource, at the time you opened this record.
Versions tab	Stores copies of all previous versions of the current record (see Record Versioning).

Assigning Tasks to a Virtual Resource

> ⚠ You can also assign a task to a virtual resource from the task screen.

1. Open the virtual resource to which you want to assign one or more existing tasks.
2. Click on the **Task Virtual Resources** tab or scroll down to that section.
3. Click the **Edit** button. The Edit Members form displays:



Collection:

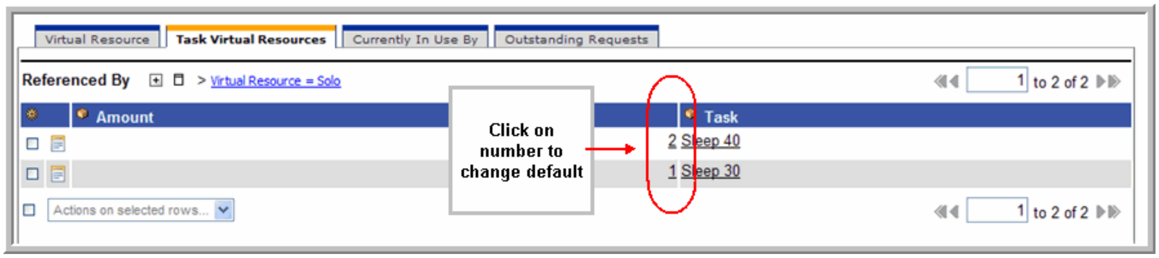
- 5x5 linux
- 5x5 Sleep
- 5x5x5 linux
- 5x5x5 Sleep
- 5x5x5x5 Linux
- 5x5x5x5 Sleep
- Boot J
- Conditional
- Delete File
- Demo
- File Exists Check
- FTP in Linux
- JUNIT Cancel Test1210786167235
- JUNIT full sleep test1210786133125
- Linux Simple
- Manual Job
- Send File Create/Delete Email
- Simple Test

Referenced By List: V-UX AGT01

- Create a txt file to kick off the file trigger
- Creation of *.txt files on Linux
- Creation of txt files

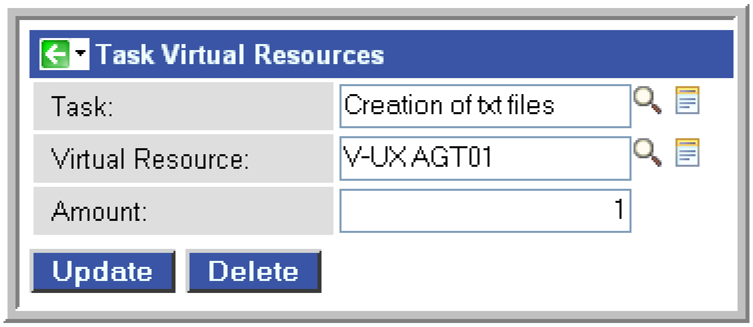
Use "Add Filter" and "Run Filter" to isolate the records to pick from

4. The tasks listed under Collection are existing tasks that are not assigned to this virtual resource. The tasks listed in the Referenced By list are tasks that refer to this virtual resource. You can filter the days listed under Collection as follows:
   a. Click **Add Filter**. See To create a filter for information about how to construct a filter.
   b. When you have defined the filter you want, click **Run Filter**. The Collection list now displays only those tasks that match the filter.
   c. To remove filtering, click the minus sign to the left of each condition, then click **Run Filter** again.
5. To add to or remove tasks from the Referenced By List:
   - To add a task to the list, double-click on the task in the Collection list.
   - To remove a task from the list, double-click on the name in the Referenced By List.
6. As you click on a task the system displays details about the task at the bottom of the form.
7. When you are finished, click **Save**.
8. Note that the default amount assigned to each task is 1, as shown.



Amount	Task
2	Sleep 40
1	Sleep 30

9. To change the amount, click on the 1. The task virtual resource record opens.



Task:	Creation of txt files
Virtual Resource:	V-UX AGT01
Amount:	1

10. Change the Amount field to the new number of units and click **Update**.

Creating Tasks

Tasks

An Opswise task executes a process on a machine, either local or remote. The process might be resident on the machine (agent-based process) or the task itself might embed the process, such as a File Monitor Task.

You can launch tasks within workflows, by way of triggers, or manually.

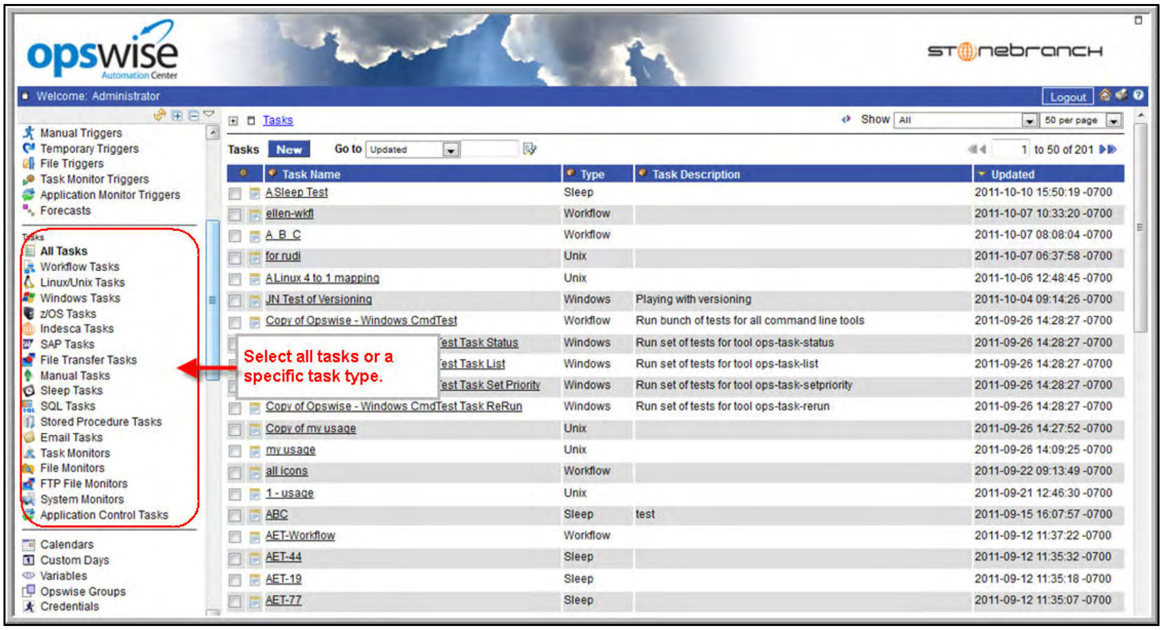
Supported task types are Workflows, Linux/Unix, Windows, z/OS, Indesca, File Transfer, Manual, Sleep, SQL, Stored Procedures, Email, Task Monitor, File Monitor, FTP File Monitor, System Monitor, SAP, and Application Control Tasks.

Several built-in variables are available for use in all task types; see Task and Task Instance Variables. Additional built-in variables exist for some specific task types; see the documentation for each task type.

The Task List screen displays a list of all currently defined tasks for all task types.

Accessing the Task List

To access the Task List screen, select **Automation Center > Tasks > All Tasks** from the Opswise navigation pane.



Select all tasks or a specific task type.

Task Name	Type	Task Description	Updated
A Sleep Test	Sleep		2011-10-10 15:50:19 -0700
ellen-wkfl	Workflow		2011-10-07 10:33:20 -0700
A B C	Workflow		2011-10-07 08:08:04 -0700
for rudi	Unix		2011-10-07 06:37:58 -0700
A Linux 4 to 1 mapping	Unix		2011-10-06 12:48:45 -0700
JN Test of Versioning	Windows	Playing with versioning	2011-10-04 09:14:26 -0700
Copy of Opswise - Windows CmdTest	Workflow	Run bunch of tests for all command line tools	2011-09-26 14:28:27 -0700
est Task Status	Windows	Run set of tests for tool ops-task-status	2011-09-26 14:28:27 -0700
est Task List	Windows	Run set of tests for tool ops-task-list	2011-09-26 14:28:27 -0700
est Task Set Priority	Windows	Run set of tests for tool ops-task-setpriority	2011-09-26 14:28:27 -0700
Copy of Opswise - Windows CmdTest Task ReRun	Windows	Run set of tests for tool ops-task-rerun	2011-09-26 14:28:27 -0700
Copy of my usage	Unix		2011-09-26 14:27:52 -0700
my usage	Unix		2011-09-26 14:09:25 -0700
all icons	Workflow		2011-09-22 09:13:49 -0700
1 - usage	Unix		2011-09-21 12:46:30 -0700
ABC	Sleep	test	2011-09-15 16:07:57 -0700
AET-Workflow	Workflow		2011-09-12 11:37:22 -0700
AET-44	Sleep		2011-09-12 11:35:32 -0700
AET-19	Sleep		2011-09-12 11:35:18 -0700
AET-77	Sleep		2011-09-12 11:35:07 -0700

The following table provides column descriptions for the default display of the Task List screen.

For information about customizing this list, including filtering, sorting, searching, and other list features, see Using Lists.

Field Name	Description
Task Name	User-defined. Name assigned to this task.

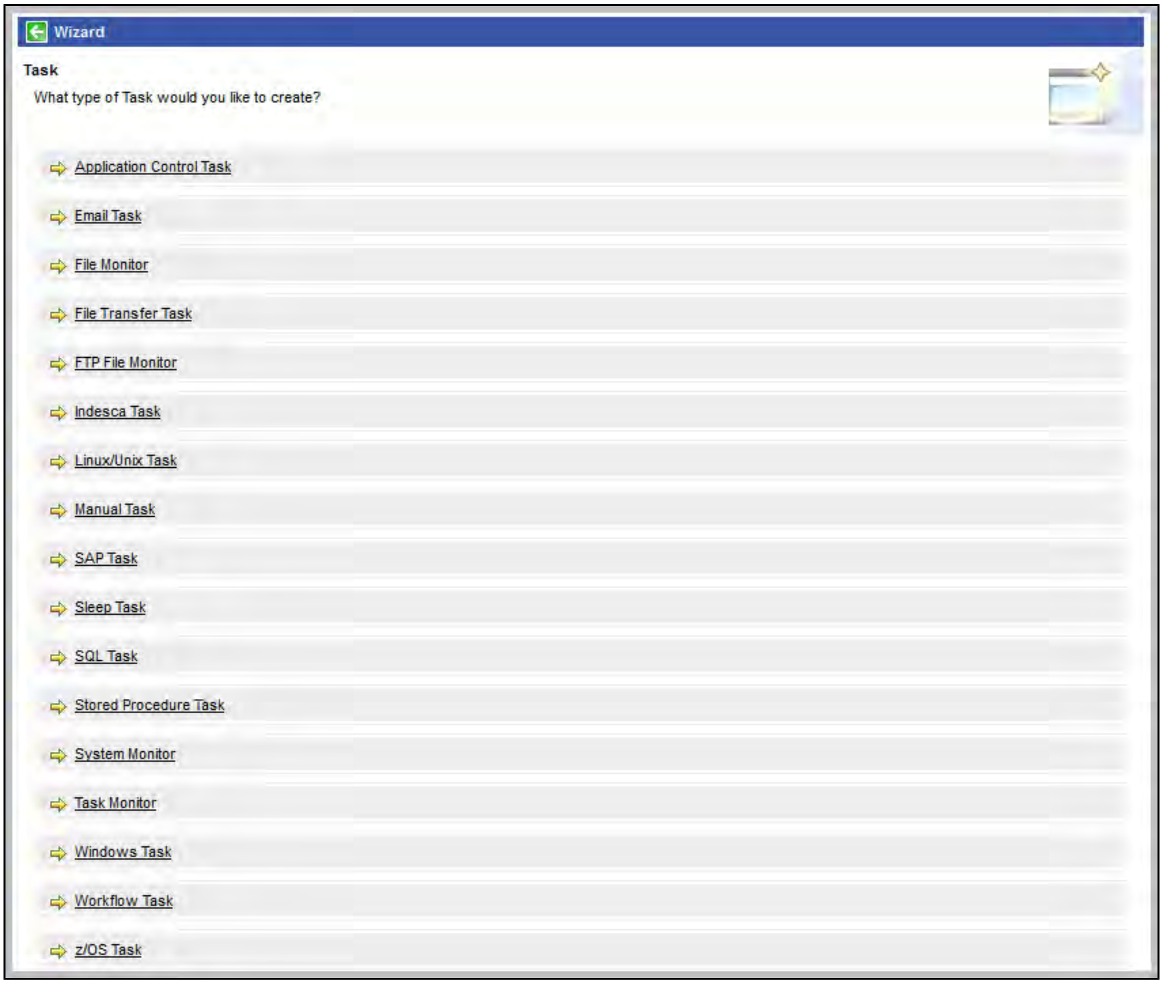
Type	The type of task. Options: • Workflow Tasks • Linux/Unix Tasks • Windows Tasks • z/OS Tasks • Indesca Tasks • SAP Tasks • File Transfer Tasks • Manual Tasks • Sleep Tasks • SQL Tasks • Stored Procedure Tasks • Email Tasks • Task Monitors • File Monitors • FTP File Monitors • System Monitors • Application Control Tasks
Task Description	User-defined. Copied from the Summary field in the task.
Updated	System-supplied. The date and time this record was last updated.

To see a list of tasks for a single task type, select that task type from the navigation pane.

Creating a Task

You can create a task either of two ways.

1. From the Navigation Pane:
    a. Select a task type from the navigation pane. The Task List screen for that task type displays.
    b. Click the **New** button. The Task Definition screen for that task type displays.
2. From the Task List screen:
    a. Click the New button on the Task List screen. The Task Wizard screen displays.



Task

What type of Task would you like to create?

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

1. a. Click the task type for task you want to create. The Task List screen for that task type displays.

For detailed information on creating tasks for specific task types, click that task type in the navigation panel on the left-hand side of this page.

Linux Unix Task

Before You Begin

The Linux/Unix agent task allows you to run a platform-specific application on a Linux/Unix machine. To run a Linux/Unix task, you must first complete the following tasks:

- Install an Opswise Linux/Unix agent on a Linux/Unix machine.
- Launch the agent. When the agent connects with the core processor, it automatically creates an agent resource definition in the database.
- Optionally, customize the agent heartbeat and log levels, as described in Linux/Unix Resource Field Definitions.

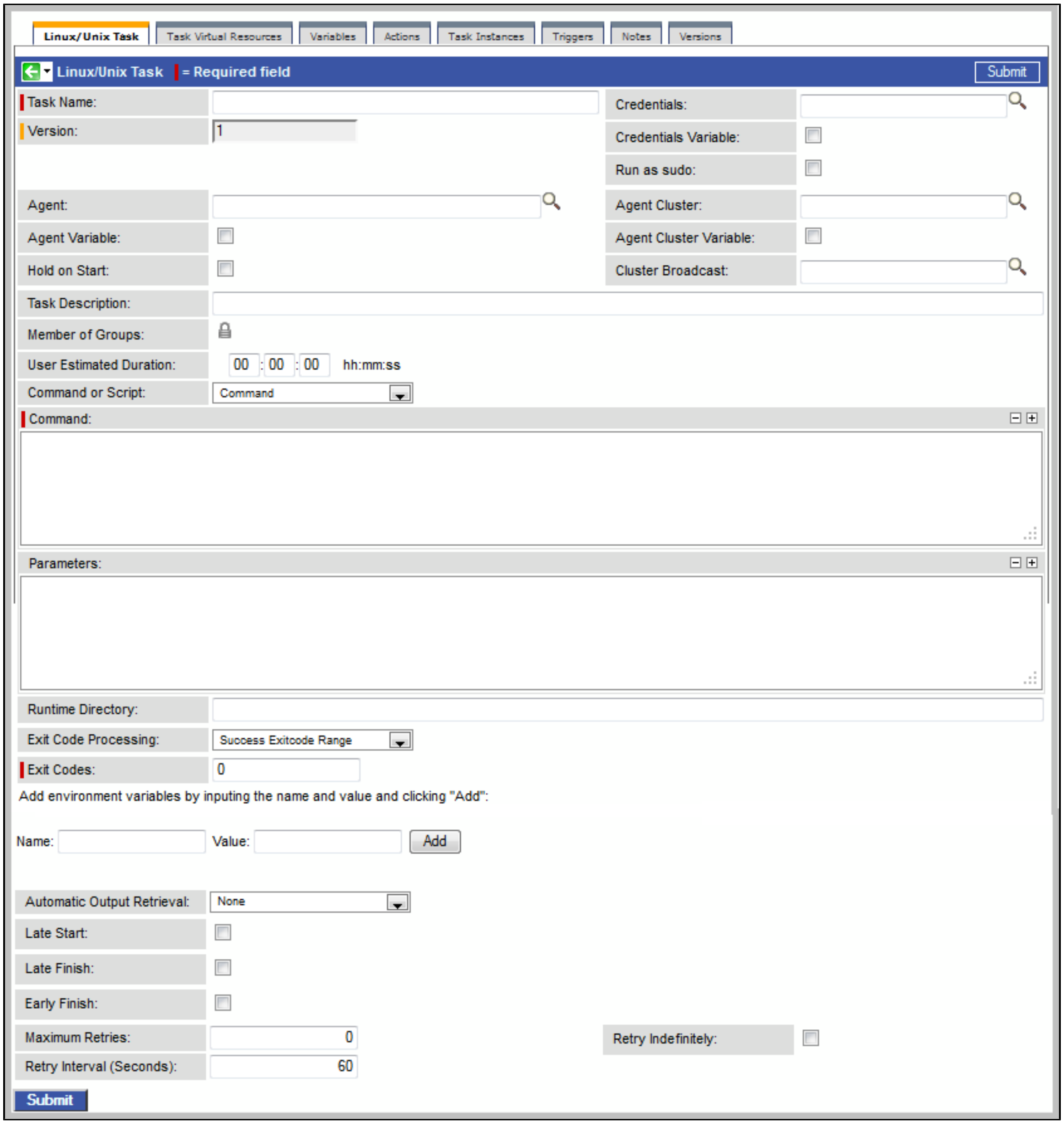
Built-In Variables

The built-in variables outlined below can be used in a Linux/Unix task to pass data where appropriate:

- Task and Task Instance Variables
- Script Variables
- Agent Variables

Creating a New Linux/Unix Task

1. From the navigation pane, select **Automation Center > Tasks > Linux/Unix Tasks**. The Linux/Unix Tasks List screen displays.
2. Click **New**. The Linux/Unix Task Definition screen displays.



Task Name:		Credentials:	
Version:	1	Credentials Variable:	□
		Run as sudo:	□
Agent:		Agent Cluster:	
Agent Variable:	□	Agent Cluster Variable:	□
Hold on Start:	□	Cluster Broadcast:	
Task Description:			
Member of Groups:	🔒		
User Estimated Duration:	00 : 00 : 00 hh:mm:ss		
Command or Script:	Command		
Command:			
Parameters:			
Runtime Directory:			
Exit Code Processing:	Success Exitcode Range		
Exit Codes:	0		

Add environment variables by inputing the name and value and clicking "Add":

Name: Value: Add

Automatic Output Retrieval:	None		
Late Start:	□		
Late Finish:	□		
Early Finish:	□		
Maximum Retries:	0	Retry Indefinitely:	□
Retry Interval (Seconds):	60		

3. Using the field descriptions provided below as a guide, complete the fields as needed.
4. Click the **Submit** button to save the record and return to the menu, or right-click on the title bar and select **Save** to save the record and remain on the current display.
5. If appropriate, repeat these steps for any additional tasks you want to add.

The task run statistics shown at the bottom of the screen appear after the first time this task has been launched.

Linux/Unix Task Field Descriptions

The table below describes the fields, buttons, and tabs on the task definition and task instance screens. Color coding is provided that differentiates the following three types of fields:

- Fields that display on the task definition and task instance screens are shown in black.
- Fields that display only on the task definition screen are shown in green.
- Fields that display only on the task instance screen are shown in maroon.

Field Name	Description
Task/Instance Name	Required. Name used within Opswise to identify this task. Up to 40 alphanumerics; variables supported. It is the user's responsibility to develop a workable naming scheme for tasks.

Invoked by	Task instance only; system-supplied. How the task instance was launched. One of the following: • Trigger: (Trigger Name) - The instance was launched by the named trigger. • Workflow: (Workflow Name) - The instance was launched by the named workflow. • Manually Launched - The instance was launched by a user. To determine the name of the user: 1. From the Activity or All Task Instances screen, click the task instance name to open the record. 2. The Execution User field identifies the user who launched the task instance.
Task	Task instance only; system-supplied. Hover over the paper icon to display more information about the task instance. Click the paper icon to display the task definition record.
Execution User	Task instance only; system-supplied. If the task was launched manually, the ID of the user who launched it.
Instance Reference Id	Task instance only; system-supplied. The system increments this number each time the task is run.
Credentials	Login credentials Opswise Automation Center will use to access the remote machine. They must be provided either in the resource definition or task definition, or both. Credentials provided in the task definition override any credentials provided in the agent resource definition. See Credentials.
Version	Task definition only; system-supplied. The version number of the current record, which is incremented by the system every time a user updates a record. Click on the Versions tab to view previous versions. For details, see Record Versioning.
Credentials Variable	Optional. If enabled, the Credentials field (see above) converts from a reference field (where you browse and select a record) into a text field that allows you to enter a variable. Use the format: ${variable name}. The variable must be a supported type as described in Using Variables and Functions .
Run as sudo	Optional. Run the command as Sudo (superuser do).
Agent	Optional. The name of the agent resource definition that identifies the machine where the operation will run. If you do not specify an agent, you must specify an agent cluster (see below).
Agent Cluster	Optional. You can specify an agent cluster in addition to or in place of a specific agent. An agent cluster is a group of agents, one of which the system will choose to run this task. If you specify an agent and an agent cluster, Opswise Automation Center first tries to run the task on the specific agent. If the agent is not available, the system reverts to the agent cluster. See Configuring Agent Clusters for more information.
Agent Variable	Optional. If enabled, the Agent field (see above) converts from a reference field (where you browse and select a record) into a text field that allows you to enter a variable. Use the format: ${variable name}. The variable must be a supported type as described in Using Variables and Functions.
Agent Cluster Variable	Optional. If enabled, the Agent Cluster field (see above) converts from a reference field (where you browse and select a record) into a text field that allows you to enter a variable. Use the format: ${variable name}. The variable must be a supported type as described in Using Variables and Functions.
Hold on Start	If enabled, when the task is launched it appears in the Activity display with a status of Held. The task runs when the user releases it.
Hold Reason	Information about why the task will be put on hold when it starts.

Cluster Broadcast	Task definition only; optional. You can specify a Cluster Broadcast in addition to or in place of a specific agent and/or Agent Cluster. When you specify an agent cluster in the Cluster Broadcast field, Opswise Automation Center runs the task on all the agents in the cluster. Each instance of the task running on its own agent becomes a separate task instance record in the database and displays separately in the Activity monitor. See Configuring Agent Clusters for more information about defining agent clusters.
Task Description	User-supplied description of this record.
Status	Task instance only; system-supplied. See Task Statuses.
Exit Code	Task instance only; system-supplied. The exit code captured by the agent when executing the task (for example, a command or script).
Status Description	Task instance only; system-supplied. Provides additional information, if any, about the status of the task.
Queued Time	Task instance only; system-supplied. The time the task was queued for processing.
Process ID	Task instance only; system-supplied. The ID of the process that was launched.
Start Time	Task instance only; system-supplied. The date and time the task started.
CPU Time	Task instance only; system-supplied. The amount of CPU time the task took to run.
End Time	Task instance only; system-supplied. The date and time the task instance completed.
Duration	Task instance only; system-supplied. The amount of time the task took to run.
Member of Groups	User-defined. Allows you to select one or more Opswise groups that this record definition belongs to. Click on the lock icon to unlock the field and select groups.
User Estimated Duration	Task definition only; optional. The estimated amount of time it should normally take to run this task. The system uses this information to calculate the User Estimated End Time on a task instance record.
Command or Script	Specifies whether a single command or a script is being executed. Options: • Command (default) • Script
Script	Name of the script that has been uploaded into the Script Library and will be executed by this task.
Command	Command being executed on the remote machine. Variables supported.
Parameters	Optional. Any arguments needed by the program to execute properly. Variables supported.

Runtime Directory	Optional. The directory from which the application should be executed. Variables supported.
Exit Code Processing	Required. Specifies how Opswise Automation Center should determine whether the executed command failed or completed successfully. Options: - **Success Exitcode Range** - The command is considered completed successfully if its exit code falls within the range specified in the Exit Codes field (see below). - **Failure Exitcode Range** - The command is considered failed if its exit code falls within the range specified in the Exit Codes field (see below). - **Success Output Contains** - The command is considered completed successfully if its output contains the text specified in the Scan Output For field (see below). - **Failure Output Contains** - The command is considered failed if its output contains the text specified in the Scan Output For field (see below). - **Step Conditions** - The command is considered completed successfully/failed if any of its specified condition codes falls within the range specified under the Step Conditions tab (see Creating Step Conditions).
Output Type-Exit Code	Required if Exit Code Processing = Success Output Contains or Failure Output Contains. This field specifies the type of output. Options: - Standard Output (STDOUT) - Standard Error (STDERR) - File
Exit Codes	Required if Exit Code Processing = Success Exitcode Range or Failure Exit Code Range. This field specifies the range. Format: Numeric. Use commas to list a series of exitcodes; use hyphens to specify a range. Example: 1,5, 22-30.
Scan Output For	Required if Exit Code Processing = Success Output Contains or Failure Output Contains. This field specifies the text for which Opswise should scan the output file. Opswise will process this field as a regular expression.
Output File-Exit Codes	Required if Output Type = File. This field specifies the path and file name of the output file that should be scanned for the text in the Scan Output For field.
Environment Variables-Name and Value	Optional. Allows you to enter environment variables needed by the program to run. Type in the name of the variable and its value, and click **Add**. The variable is listed in the space underneath. You can add as many variables as needed. To delete a variable, click the **X** button.
Automatic Output Retrieval	Optional. Allows you to specify whether you want the system to automatically retrieve any output from the job and attach it to the task instance record. Options: - **None** - Do not attach any output to the task instance record. - **Standard Output** - Attach all standard output. - **Standard Error** - Attach standard error output. - **File** - Attach the file specified in the Output File field.
Output File-Auto Retrieve	Required if Automatic Output Retrieval=File. This field specifies the path and filename containing the output that you want automatically retrieved and attached to the task instance.
Start Line	Optional. Allows you to instruct the system to fetch data beginning at the line indicated.
Number of Lines	Optional. Allows you to limit the retrieved data to the number of lines specified. The default is 100. You can change the default by modifying the record **Configuration > Properties > Retrieve Output Default Maximum Lines**.

Scan Text-Automatic Output Retrieval	Optional. Instructs the system to scan the data for the text specified and retrieve only that. Opswise will process this field as a regular expression.
Late Start	If enabled, and if the task instance starts after the time specified in the Late Start Time field (see below), the task instance is flagged as late. To determine whether a task instance started late, open the task instance and locate the Started Late field; the field is checked if the instance started late.
Late Start Type	Required if Late Start is enabled. Options are: - **Time** - Flag the task if it starts after the specified time. - **Duration** - Flag the task if it starts a certain amount of time after the programmed start time. The task must have a specific start time.
Late Start Time	Time after which the task start time is considered late. Use hh:mm, 24-hour time
Late Start Duration	Duration (amount of relative time) after which the task is considered to have started late. For a task within a workflow, the duration is the period between the time the workflow starts and the time the task itself starts. For example, a task might have a Late Start Duration of 60 minutes. If the workflow starts at 9:00 a.m. but the task itself does not start until 10:30, the task has started late. For a task that is not within a workflow, Late Start Duration has meaning only if the task has been held upon starting. For example, if a task has a Late Start Duration of 60 minutes and the Hold on Start field is enabled, if the task is not released from hold within the amount of time specified in the Late Start Duration field, the task has started late.
Started Late	Task instance only; system-supplied. This field is flagged if the task started later than the time specified in the Late Start fields.
Late Finish	If enabled, and if the task instance finishes after the time or period specified, the task instance is flagged as late. You can specify a time or duration to determine a late finish (see Late Finish Type below). To determine whether a task instance finished late, open the task instance and locate the Finished Late field; the field is checked if the instance finished after the specified time or lasted longer than expected. This field only appears on the task instance if the user specified a Late Finish in the task definition.
Late Finish Type	If Late Finish is enabled, use this field to specify whether you want to use time (see Late Finish Time below) or duration (see Late Finish Duration below) to determine whether the task instance finished late.
Late Finish Time	If Late Finish Type is Time, use this to specify the time after which the task finish time is considered late. Use hh:mm, 24-hour time.
Late Finish Duration	If Late Finish Type is Duration, use this to specify the longest amount of time this task instance should take to run. You can specify any combination of hours, minutes, and seconds.
Finished Late	Task instance only; system-supplied. This field is flagged if the task finished later than the time or duration specified in the Late Finish fields.
Early Finish	If enabled, and if the task instance finishes before the time or period specified, the task instance is flagged as early. You can specify a time or duration to determine an early finish (see Early Finish Type below). To determine whether a task instance finished early, open the task instance and locate the Finished Early field; the field is checked if the instance finished before the specified time or did not last as long as expected. This field only appears on the task instance if the user added Early Finish specifications to the task definition.

Early Finish Type	If Early Finish is enabled, use this field to specify whether you want to use time (see Early Finish Time below) or duration (see Early Finish Duration below) to determine whether the task instance finished early.
Early Finish Time	If Early Finish Type is Time, use this to specify the time before which the task finish time is considered early. That is, enter a time at which the task should still be running. Use hh:mm, 24-hour time.
Early Finish Duration	If Early Finish Type is Duration, use this to specify the shortest amount of time this task instance should take to run. You can specify any combination of hours, minutes, and seconds.
Finished Early	Task instance only; system-supplied. This field is flagged if the task finished earlier than the time specified in the Early Finish fields.
Maximum Retries	User-defined. The maximum number of times the system should retry this task after it has started and gone to a failed state.
Task Priority	Task instance only; the priority of this task instance, as set by the user via the Set Priority command. Options are: HIGH, MEDIUM, LOW.
Retry Indefinitely	User-defined. Enabled or disabled. Indicates whether the system should continue trying indefinitely to run this task. If you enable this field, it overrides any value placed in the Maximum Retries field (above).
Retry Interval	User-defined. The number of seconds between each retry.
Current Retry Count	Task instance only; system-supplied. Displays only for a running task instance. Displays the current number of times the system has retried the task after it first went to failure status.
First Time Ran	Task definition only; system-supplied. Displays after the first time the task runs. The date and time this task first ran.
Lowest Instance Time	Task definition only; system-supplied. Displays after the first time the task runs. The shortest amount of time this task has taken to run.
Last Time Ran	Task definition only; system-supplied. Displays after the first time the task runs. The most recent date and time the task ran.
Average Instance Time	Task definition only; system-supplied. Displays after the first time the task runs. Shows the average amount of time this task takes to run.
Number of Instances	Task definition only; system-supplied. Displays after the first time the task runs. Shows the number of times this task has run.
Highest Instance Time	Task definition only; system-supplied. Displays after the first time the task runs. The longest amount of time this task has taken to run.
Last Instance Duration	Task definition only; system-supplied. Displays after the first time the task runs. The amount of time the task took to run the last time it ran.

Submit button	Submits the new record to the database.
Update button	Saves updates to the record.
Launch Task button	Manually launches the task.
Delete button	Deletes the current record.
View Parent button	Task instance only; Displays this task's parent task (workflow), if any.
Show Details button	Task instance only; displays detailed information about this task instance.
Skip button	Task instance only; for tasks loaded into the schedule that haven't run yet. Allows you to tell the system to skip this task. See Skipping a Task.
Hold button	Task instance only; see Putting a Task on Hold.
Force Finish button	Task instance only; see Force Finishing a Task.
Re-run button	Task instance only; see Re-running a Task.
Output tab	Task instance only. Displays output generated from the process, if any, based on specifications provided by the user in the Automatic Output Retrieval fields in the task definition.
Task Virtual Resources tab	Lists Virtual Resources to which this task is assigned. See Creating Virtual Resources.
Variables tab	Displays all variables associated with this record.
Actions tab	Allows you to specify actions that the system will take automatically based on events that occur during the execution of this task. Events are task instance status, exit codes, late start, late finish, and early finish. Actions are: • **Abort Action** - Abort the task if certain events occur. For details, see Setting Up Abort Actions. • **Email Notification** - Send an email if certain events occur. For details, see Creating Email Notifications. • **SNMP Notification** - Send an email if certain events occur. For details, see Creating SNMP Notifications. • **Set Variable** - Used in workflows to set a variable based on the occurrence of certain events. For details, see Setting Variables in a Workflow.
Task Instances tab	System-supplied. Displays a list of all task instances generated by this task definition.
Triggers tab	Displays a list of all triggers that have been defined to launch this task. Also allows you to add new triggers. If you add a new trigger from this location, the system automatically constructs a default trigger name as follows: <current task name>#TRIGGER#. You can change the default name if desired. For instructions on creating triggers, see Creating Triggers.

Notes tab	Displays all notes associated with this task. See Runbook Notes.
Versions tab	Stores copies of all previous versions of the current record (see Record Versioning).

Specifying When a Task Runs

You can run the task as part of a workflow, specify triggers that run the task automatically based on times or events, or run the task manually.

Monitoring Task Execution

You can monitor all system activity from the Activity display and can view activity history from the Activity History screen.

Windows Task

Before You Begin

The Windows task allows you to run a platform-specific application on a Windows machine. To run a Windows task, you must first complete the following tasks:

- Install an Opswise Windows agent on a Windows machine
- Launch the agent. When the agent connects with the core processor, it automatically creates an agent resource definition in the database.
- Optionally, customize the agent heartbeat and log levels, as described in Windows Resource Field Definitions.

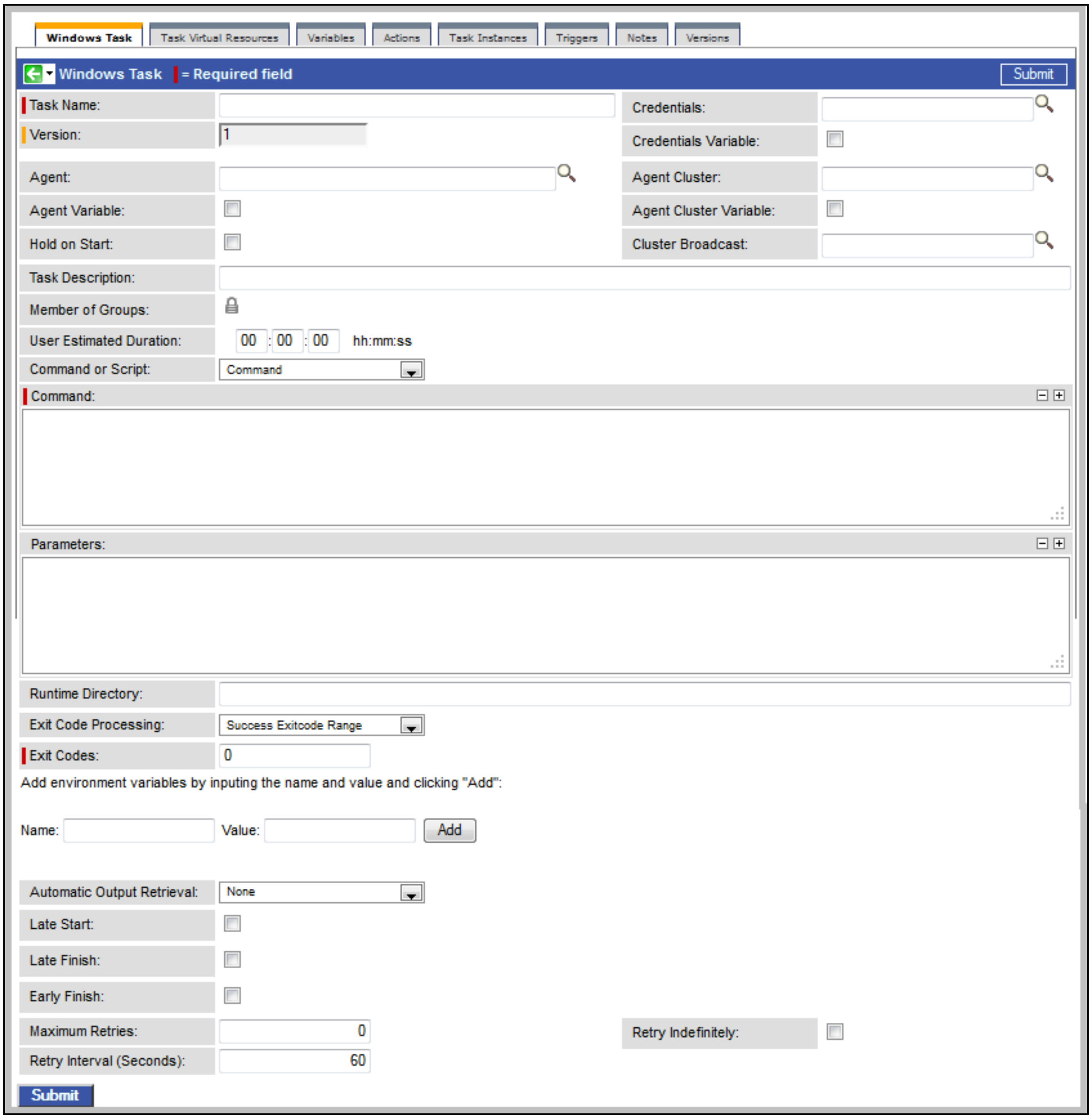
Built-In Variables

The built-in variables outlined below can be used in a Windows task to pass data where appropriate:

- Task and Task Instance Variables
- Script Variables
- Agent Variables

Creating a New Windows Task

1. From the navigation pane, select **Automation Center > Tasks > Windows Tasks**. The Windows Tasks List screen displays.
2. Click **New**. The Windows Task Definition screen displays.



Version: 1

User Estimated Duration: 00 : 00 : 00 hh:mm:ss

Command or Script: Command

Exit Code Processing: Success Exitcode Range

Exit Codes: 0

Automatic Output Retrieval: None

Maximum Retries: 0

Retry Interval (Seconds): 60

3. Using the field descriptions provided below as a guide, complete the fields as needed.
4. Click the **Submit** button to save the record and return to the menu, or right-click on the title bar and select **Save** to save the record and remain on the current display.
5. If appropriate, repeat these steps for any additional tasks you want to add.

Task run statistics appear after the first time this task has been launched.

Windows Task Field Descriptions

The table below describes the fields, buttons, and tabs on the task definition and task instance screens. Color coding is provided that differentiates the following three types of fields:

- Fields that display on the task definition and task instance screens are shown in black.
- Fields that display only on the task definition screen are shown in green.
- Fields that display only on the task instance screen are shown in maroon.

Field Name	Description
Task/Instance Name	Required. Name used within Opswise to identify this task. Up to 40 alphanumerics; variables supported. It is the user's responsibility to develop a workable naming scheme for tasks.

Invoked by	Task instance only; system-supplied. How the task instance was launched. One of the following: • Trigger: (Trigger Name) - The instance was launched by the named trigger. • Workflow: (Workflow Name) - The instance was launched by the named workflow. • Manually Launched - The instance was launched by a user. To determine the name of the user: 1. From the Activity or All Task Instances screen, click the task instance name to open the record. 2. The Execution User field identifies the user who launched the task instance.
Task	Task instance only; system-supplied. Hover over the paper icon to display more information about the task instance. Click the paper icon to display the task definition record.
Execution User	Task instance only; system-supplied. If the task was launched manually, the ID of the user who launched it.
Instance Reference Id	Task instance only; system-supplied. The system increments this number each time the task is run.
Credentials	Login credentials Opswise Automation Center will use to access the remote machine. They must be provided either in the resource definition or task definition, or both. Credentials provided in the task definition override any credentials provided in the agent resource definition. See Credentials.
Version	Task definition only; system-supplied. The version number of the current record, which is incremented by the system every time a user updates a record. Click on the Versions tab to view previous versions. For details, see Record Versioning.
Credentials Variable	Optional. If enabled, the Credentials field (see above) converts from a reference field (where you browse and select a record) into a text field that allows you to enter a variable. Use the format: ${variable name}. The variable must be a supported type as described in Using Variables and Functions .
Agent	Optional. The name of the agent resource definition that identifies the machine where the operation will run. If you do not specify an agent, you must specify an agent cluster (see below).
Agent Cluster	Optional. You can specify an agent cluster in addition to or in place of a specific agent. An agent cluster is a group of agents, one of which the system will choose to run this task. If you specify an agent and an agent cluster, Opswise Automation Center first tries to run the task on the specific agent. If the agent is not available, the system reverts to the agent cluster. See Configuring Agent Clusters for more information.
Agent Variable	Optional. If enabled, the Agent field (see above) converts from a reference field (where you browse and select a record) into a text field that allows you to enter a variable. Use the format: ${variable name}. The variable must be a supported type as described in Using Variables and Functions.
Agent Cluster Variable	Optional. If enabled, the Agent Cluster field (see above) converts from a reference field (where you browse and select a record) into a text field that allows you to enter a variable. Use the format: ${variable name}. The variable must be a supported type as described in Using Variables and Functions.
Hold on Start	If enabled, when the task is launched it appears in the Activity display with a status of Held. The task runs when the user releases it.
Hold Reason	Information about why the task will be put on hold when it starts.

Cluster Broadcast	Task definition only; optional. You can specify a Cluster Broadcast in addition to or in place of a specific agent and/or Agent Cluster. When you specify an agent cluster in the Cluster Broadcast field, Opswise Automation Center runs the task on all the agents in the cluster. Each instance of the task running on its own agent becomes a separate task instance record in the database and displays separately in the Activity monitor. See Configuring Agent Clusters for more information about defining agent clusters.
Task Description	User-supplied description of this record.
Status	Task instance only; system-supplied. See Task Statuses.
Exit Code	Task instance only; system-supplied. The exit code captured by the agent when executing the task (for example, a command or script).
Status Description	Task instance only; system-supplied. Provides additional information, if any, about the status of the task.
Queued Time	Task instance only; system-supplied. The time the task was queued for processing.
Process ID	Task instance only; system-supplied. The ID of the process that was launched.
Start Time	Task instance only; system-supplied. The date and time the task started.
CPU Time	Task instance only; system-supplied. The amount of CPU time the task took to run.
End Time	Task instance only; system-supplied. The date and time the task instance completed.
Duration	Task instance only; system-supplied. The amount of time the task took to run.
Member of Groups	User-defined. Allows you to select one or more Opswise groups that this record definition belongs to. Click on the lock icon to unlock the field and select groups.
User Estimated Duration	Task definition only; optional. The estimated amount of time it should normally take to run this task. The system uses this information to calculate the User Estimated End Time on a task instance record.
Command or Script	Specifies whether a single command or a script is being executed. Options: • Command (default) • Script
Command	Command being executed on the remote machine. Variables supported.
Script	Name of the script that has been uploaded into the Script Library and will be executed by this task.
Parameters	Optional. Any arguments needed by the program to execute properly. Variables supported.

Runtime Directory	Optional. The directory from which the application should be executed. Variables supported.
Exit Code Processing	Required. Specifies how Opswise Automation Center should determine whether the executed command failed or completed successfully. Options: • **Success Exitcode Range** - The command is considered completed successfully if its exit code falls within the range specified in the Exit Codes field (see below). • **Failure Exitcode Range** - The command is considered failed if its exit code falls within the range specified in the Exit Codes field (see below). • **Success Output Contains** - The command is considered completed successfully if its output contains the text specified in the Scan Output For field (see below). • **Failure Output Contains** - The command is considered failed if its output contains the text specified in the Scan Output For field (see below). • **Step Conditions** - The command is considered completed successfully/failed if any of its specified condition codes falls within the range specified under the Step Conditions tab (see Creating Step Conditions).
Output Type-Exit Code	Required if Exit Code Processing = Success Output Contains or Failure Output Contains. This field specifies the type of output. Options: • Standard Output (STDOUT) • Standard Error (STDERR) • File
Exit Codes	Required if Exit Code Processing = Success Exitcode Range or Failure Exit Code Range. This field specifies the range. Format: Numeric. Use commas to list a series of exitcodes; use hyphens to specify a range. Example: 1,5, 22-30.
Scan Output For	Required if Exit Code Processing = Success Output Contains or Failure Output Contains. This field specifies the text for which Opswise should scan the output file. Opswise will process this field as a regular expression.
Output File-Exit Codes	Required if Output Type = File. This field specifies the path and file name of the output file that should be scanned for the text in the Scan Output For field.
Environment Variables-Name and Value	Optional. Allows you to enter environment variables needed by the program to run. Type in the name of the variable and its value, and click **Add**. The variable is listed in the space underneath. You can add as many variables as needed. To delete a variable, click the **X** button.
Automatic Output Retrieval	Optional. Allows you to specify whether you want the system to automatically retrieve any output from the job and attach it to the task instance record. Options: • **None** - Do not attach any output to the task instance record. • **Standard Output** - Attach all standard output. • **Standard Error** - Attach standard error output. • **File** - Attach the file specified in the Output File field.
Output File-Auto Retrieve	Required if Automatic Output Retrieval=File. This field specifies the path and filename containing the output that you want automatically retrieved and attached to the task instance.
Start Line	Optional. Allows you to instruct the system to fetch data beginning at the line indicated.
Number of Lines	Optional. Allows you to limit the retrieved data to the number of lines specified. The default is 100. You can change the default by modifying the record **Configuration > Properties > Retrieve Output Default Maximum Lines**.

Scan Text-Automatic Output Retrieval	Optional. Instructs the system to scan the data for the text specified and retrieve only that. Opswise will process this field as a regular expression.
Late Start	If enabled, and if the task instance starts after the time specified in the Late Start Time field (see below), the task instance is flagged as late. To determine whether a task instance started late, open the task instance and locate the Started Late field; the field is checked if the instance started late.
Late Start Type	Required if Late Start is enabled. Options are: - **Time** - Flag the task if it starts after the specified time. - **Duration** - Flag the task if it starts a certain amount of time after the programmed start time. The task must have a specific start time.
Late Start Time	Time after which the task start time is considered late. Use hh:mm, 24-hour time
Late Start Duration	Duration (amount of relative time) after which the task is considered to have started late. For a task within a workflow, the duration is the period between the time the workflow starts and the time the task itself starts. For example, a task might have a Late Start Duration of 60 minutes. If the workflow starts at 9:00 a.m. but the task itself does not start until 10:30, the task has started late. For a task that is not within a workflow, Late Start Duration has meaning only if the task has been held upon starting. For example, if a task has a Late Start Duration of 60 minutes and the Hold on Start field is enabled, if the task is not released from hold within the amount of time specified in the Late Start Duration field, the task has started late.
Started Late	Task instance only; system-supplied. This field is flagged if the task started later than the time specified in the Late Start fields.
Late Finish	If enabled, and if the task instance finishes after the time or period specified, the task instance is flagged as late. You can specify a time or duration to determine a late finish (see Late Finish Type below). To determine whether a task instance finished late, open the task instance and locate the Finished Late field; the field is checked if the instance finished after the specified time or lasted longer than expected. This field only appears on the task instance if the user specified a Late Finish in the task definition.
Late Finish Type	If Late Finish is enabled, use this field to specify whether you want to use time (see Late Finish Time below) or duration (see Late Finish Duration below) to determine whether the task instance finished late.
Late Finish Time	If Late Finish Type is Time, use this to specify the time after which the task finish time is considered late. Use hh:mm, 24-hour time.
Late Finish Duration	If Late Finish Type is Duration, use this to specify the longest amount of time this task instance should take to run. You can specify any combination of hours, minutes, and seconds.
Finished Late	Task instance only; system-supplied. This field is flagged if the task finished later than the time or duration specified in the Late Finish fields.
Early Finish	If enabled, and if the task instance finishes before the time or period specified, the task instance is flagged as early. You can specify a time or duration to determine an early finish (see Early Finish Type below). To determine whether a task instance finished early, open the task instance and locate the Finished Early field; the field is checked if the instance finished before the specified time or did not last as long as expected. This field only appears on the task instance if the user added Early Finish specifications to the task definition.

Early Finish Type	If Early Finish is enabled, use this field to specify whether you want to use time (see Early Finish Time below) or duration (see Early Finish Duration below) to determine whether the task instance finished early.
Early Finish Time	If Early Finish Type is Time, use this to specify the time before which the task finish time is considered early. That is, enter a time at which the task should still be running. Use hh:mm, 24-hour time.
Early Finish Duration	If Early Finish Type is Duration, use this to specify the shortest amount of time this task instance should take to run. You can specify any combination of hours, minutes, and seconds.
Finished Early	Task instance only; system-supplied. This field is flagged if the task finished earlier than the time specified in the Early Finish fields.
Maximum Retries	User-defined. The maximum number of times the system should retry this task after it has started and gone to a failed state.
Retry Indefinitely	User-defined. Enabled or disabled. Indicates whether the system should continue trying indefinitely to run this task. If you enable this field, it overrides any value placed in the Maximum Retries field (above).
Retry Interval	User-defined. The number of seconds between each retry.
Current Retry Count	Task instance only; system-supplied. Displays only for a running task instance. Displays the current number of times the system has retried the task after it first went to failure status.
First Time Ran	Task definition only; system-supplied. Displays after the first time the task runs. The date and time this task first ran.
Lowest Instance Time	Task definition only; system-supplied. Displays after the first time the task runs. The shortest amount of time this task has taken to run.
Last Time Ran	Task definition only; system-supplied. Displays after the first time the task runs. The most recent date and time the task ran.
Average Instance Time	Task definition only; system-supplied. Displays after the first time the task runs. Shows the average amount of time this task takes to run.
Number of Instances	Task definition only; system-supplied. Displays after the first time the task runs. Shows the number of times this task has run.
Highest Instance Time	Task definition only; system-supplied. Displays after the first time the task runs. The longest amount of time this task has taken to run.
Last Instance Duration	Task definition only; system-supplied. Displays after the first time the task runs. The amount of time the task took to run the last time it ran.
Submit button	Submits the new record to the database.
Update button	Saves updates to the record.

Launch Task button	Manually launches the task.
Delete button	Deletes the current record.
View Parent button	Task instance only; Displays this task's parent task (workflow), if any.
Show Details button	Task instance only; displays detailed information about this task instance.
Skip button	Task instance only; for tasks loaded into the schedule that haven't run yet. Allows you to tell the system to skip this task. See Skipping a Task.
Hold button	Task instance only; see Putting a Task on Hold.
Force Finish button	Task instance only; see Force Finishing a Task.
Re-run button	Task instance only; see Re-running a Task.
Output tab	Task instance only. Displays output generated from the process, if any, based on specifications provided by the user in the Automatic Output Retrieval fields in the task definition.
Task Virtual Resources tab	Lists Virtual Resources to which this task is assigned. See Creating Virtual Resources.
Variables tab	Displays all variables associated with this record.
Actions tab	Allows you to specify actions that the system will take automatically based on events that occur during the execution of this task. Events are task instance status, exit codes, late start, late finish, and early finish. Actions are: • **Abort Action** - Abort the task if certain events occur. For details, see Setting Up Abort Actions. • **Email Notification** - Send an email if certain events occur. For details, see Creating Email Notifications. • **SNMP Notification** - Send an email if certain events occur. For details, see Creating SNMP Notifications. • **Set Variable** - Used in workflows to set a variable based on the occurrence of certain events. For details, see Setting Variables in a Workflow.
Task Instances tab	System-supplied. Displays a list of all task instances generated by this task definition.
Triggers tab	Displays a list of all triggers that have been defined to launch this task. Also allows you to add new triggers. If you add a new trigger from this location, the system automatically constructs a default trigger name as follows: <current task name>#TRIGGER#. You can change the default name if desired. For instructions on creating triggers, see Creating Triggers.
Notes tab	Displays all notes associated with this task. See Runbook Notes.
Versions tab	Stores copies of all previous versions of the current record (see Record Versioning).

Specifying When a Task Runs

You can run the task as part of a workflow, specify triggers that run the task automatically based on times or events, or run the task manually.

Monitoring Task Execution

You can monitor all system activity from the Activity display and can view activity history from the Activity History screen.

zOS Task

Before You Begin

The z/OS task allow you to run a platform-specific application on a z/OS machine. To run a z/OS task, you must first complete the following tasks:

- Install an Opswise z/OS agent on a z/OS machine.
- Launch the agent. When the agent connects with the core processor, it automatically creates a z/OS agent resource definition in the database.
- Optionally, customize the agent heartbeat and log levels, as described in z/OS Resource Field Definitions.

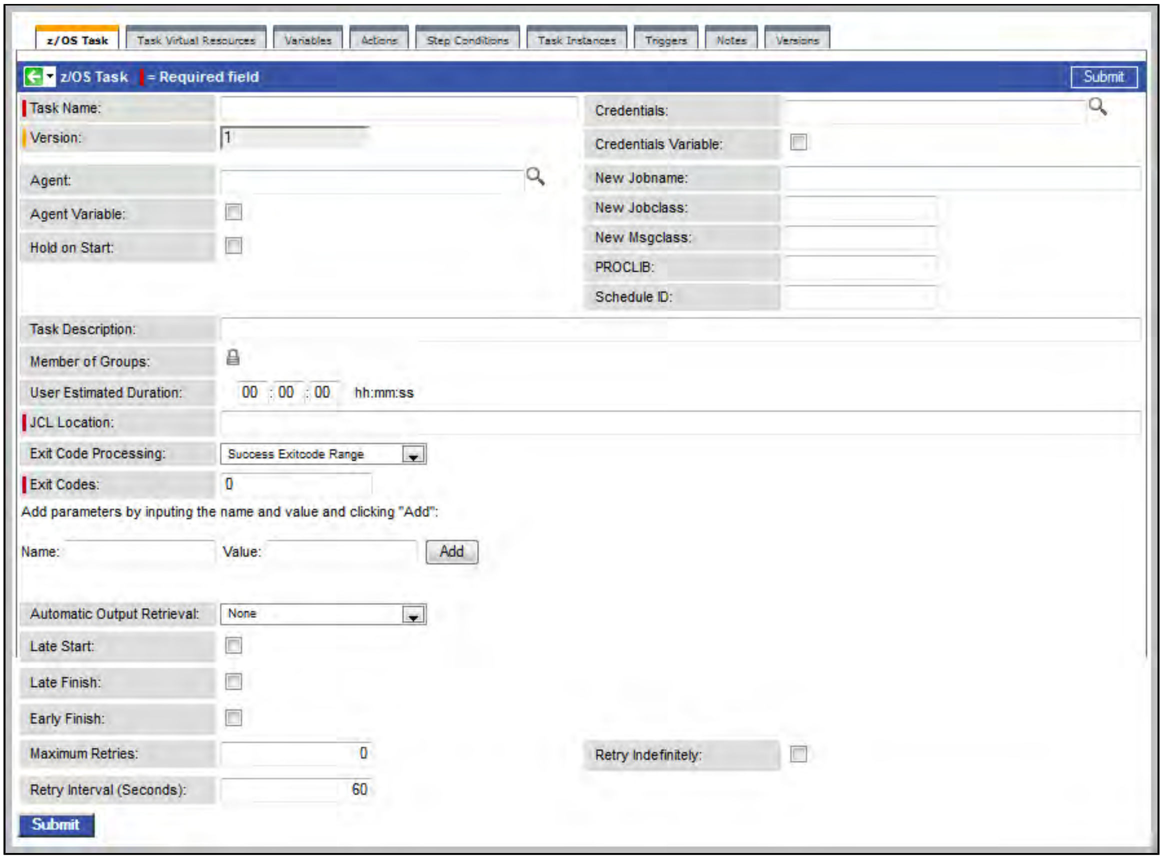
Built-In Variables

The built-in variables outlined below can be used in a z/OS task to pass data where appropriate:

- Task and Task Instance Variables
- Agent Variables

Creating a New z/OS Task

1. From the navigation pane, select **Automation Center > Tasks > z/OS Tasks**. The z/OS Tasks List screen displays.
2. Click **New**. The z/OS Task Definition screen displays.



Task Name:
Version: 1
Agent:
Agent Variable: □
Hold on Start: □
Credentials:
Credentials Variable: □
New Jobname:
New Jobclass:
New Msgclass:
PROCLIB:
Schedule ID:
Task Description:
Member of Groups:
User Estimated Duration: 00 : 00 : 00 hh:mm:ss
JCL Location:
Exit Code Processing: Success Exitcode Range
Exit Codes: 0
Add parameters by inputing the name and value and clicking "Add":
Name: Value:
Automatic Output Retrieval: None
Late Start: □
Late Finish: □
Early Finish: □
Maximum Retries: 0
Retry Indefinitely: □
Retry Interval (Seconds): 60

3. Using the field descriptions provided below as a guide, complete the fields as needed.
4. Click the **Submit** button to save the record and return to the menu, or right-click on the title bar and select **Save** to save the record and remain on the current display.
5. If appropriate, repeat these steps for any additional tasks you want to add.

Task run statistics appear after the first time this task has been launched.

z/OS Task Field Descriptions

The table below describes the fields, buttons, and tabs on the task definition and task instance screens. Color coding is provided that differentiates the following three types of fields:

- Fields that display on the task definition and task instance screens are shown in black.
- Fields that display only on the task definition screen are shown in green.
- Fields that display only on the task instance screen are shown in maroon.

Field Name	Description
Task/Instance Name	Required. Name used within Opswise to identify this task. Up to 40 alphanumerics; variables supported. It is the user's responsibility to develop a workable naming scheme for tasks.
Invoked by	Task instance only; system-supplied. How the task instance was launched. One of the following: • Trigger: (Trigger Name) - The instance was launched by the named trigger. • Workflow: (Workflow Name) - The instance was launched by the named workflow. • Manually Launched - The instance was launched by a user. To determine the name of the user: 1. From the Activity or All Task Instances screen, click the task instance name to open the record. 2. The Execution User field identifies the user who launched the task instance.
Task	Task instance only; system-supplied. Hover over the paper icon to display more information about the task instance. Click the paper icon to display the task definition record.

Execution User	Task instance only; system-supplied. If the task was launched manually, the ID of the user who launched it.
Instance Reference Id	Task instance only; system-supplied. The system increments this number each time the task is run.
Credentials	Login credentials Opswise Automation Center will use to access the remote machine. They must be provided either in the resource definition or task definition, or both. Credentials provided in the task definition override any credentials provided in the agent resource definition. See Credentials.
Version	Task definition only; system-supplied. The version number of the current record, which is incremented by the system every time a user updates a record. Click on the Versions tab to view previous versions. For details, see Record Versioning.
Credentials Variable	Optional. If enabled, the Credentials field (see above) converts from a reference field (where you browse and select a record) into a text field that allows you to enter a variable. Use the format: ${variable name}. The variable must be a supported type as described in Using Variables and Functions .
Agent	Required. The name of the agent resource definition that identifies the machine where the operation will run.
New Jobname	Optional. Jobname that will replace the one in the JCL member. This allows you to override the value in your JCL from Opswise without having to modify the JCL.
Agent Variable	Optional. If enabled, the Agent field (see above) converts from a reference field (where you browse and select a record) into a text field that allows you to enter a variable. Use the format: ${variable name}. The variable must be a supported type as described in Using Variables and Functions.
New Jobclass	New Jobclass to replace the one in the JCL member. This allows you to override the value in your JCL from Opswise without having to modify the JCL.
Hold on Start	If enabled, when the task is launched it appears in the Activity display with a status of Held. The task runs when the user releases it.
New Msgclass	Optional. New MSGCLASS to replace the one in the JCL member. This allows you to override the value in your JCL from Opswise without having to modify the JCL.
PROCLIB	Optional. When you use this parameter to specify the new value "PROC001," the submitted JCL will be modified to show the following statement: `/*JESPARM PROCLIB=PROC001` This allows you to alter the default PROCLIB used by JES2 so that Opswise supplies your catalogued procedures and you do not have to modify the JCL.
Schedule ID	Optional. The CA7 Schedule ID; for CA7 toleration only. Click here for details.
Hold Reason	Information about why the task will be put on hold when it starts.

Task Description	User-supplied description of this record.
Member of Groups	User-defined. Allows you to select one or more Opswise groups that this record definition belongs to. Click on the lock icon to unlock the field and select groups.
Status	Task instance only; system-supplied. See Task Statuses.
Exit Code	Task instance only; system-supplied. The exit code captured by the agent when executing the task (for example, a command or script).
Status Description	Task instance only; system-supplied. Provides additional information, if any, about the status of the task.
Queued Time	Task instance only; system-supplied. The time the task was queued for processing.
CPU Time	Task instance only; system-supplied. The amount of CPU time the task took to run.
Start Time	Task instance only; system-supplied. The date and time the task started.
Duration	Task instance only; system-supplied. The amount of time the task took to run.
End Time	Task instance only; system-supplied. The date and time the task instance completed.
User Estimated Duration	Task definition only; optional. The estimated amount of time it should normally take to run this task. The system uses this information to calculate the User Estimated End Time on a task instance record.
JCL Location	Required. The file and member name containing the JCL script. When you are using the JCL_LIBRARY feature, you can substitute the name of the library with a string starting with "&", that names the library specified in the uags.conf file with the JCL_library definitions. For example, the name of a job might look like the following: `&PRODLIB(PAYJOB01)`
Exit Code Processing	Required. Specifies how Opswise Automation Center should determine whether the executed command failed or completed successfully. Options: • **Success Exitcode Range** - The command is considered completed successfully if its exit code falls within the range specified in the Exit Codes field (see below). • **Failure Exitcode Range** - The command is considered failed if its exit code falls within the range specified in the Exit Codes field (see below). • **Success Output Contains** - The command is considered completed successfully if its output contains the text specified in the Scan Output For field (see below). • **Failure Output Contains** - The command is considered failed if its output contains the text specified in the Scan Output For field (see below). • **Step Conditions** - The command is considered completed successfully/failed if any of its specified condition codes falls within the range specified under the Step Conditions tab (see Creating Step Conditions).

Exit Codes	Required if Exit Code Processing = Success Exitcode Range or Failure Exit Code Range. This field specifies the range. Format: Numeric. Use commas to list a series of exitcodes; use hyphens to specify a range. Example: 1,5, 22-30.
Scan Output For	Required if Exit Code Processing = Success Output Contains or Failure Output Contains. This field specifies the text for which Opswise should scan the output file. Opswise will process this field as a regular expression.
Output Type-Exit Code	Required if Exit Code Processing = Success Output Contains or Failure Output Contains. This field specifies the type of output. Options: - Standard Output (STDOUT) - Standard Error (STDERR) - File
Output File-Exit Codes	Required if Output Type = File. This field specifies the path and file name of the output file that should be scanned for the text in the Scan Output For field.
Parameters-zos	Optional. Allows you to enter parameters that will be inserted into the JCL. The parameter consists of a keyword and a value, commonly referred to as a key-value pair. You can enter as many key-value pairs as needed. The parameters you enter each create a separate JCL construct called the SET command. Each one appears as a new line inserted dynamically into the JCL submitted to the system for the current execution. The JCL is not permanently modified. For example, you might specify the key-value pair of RUNTYPE=PROD. This results in the following JCL SET statement being inserted in the job after the job card: `// SET RUNTYPE=PROD` To enter a key-value pair, type in the name of the key and the value, and click **Add**. The system displays the parameter in the space underneath. You can add as many parameters as needed. To delete an entry, click the **X** button. The parameters fields also support two additional special functions: - They allow you to specify any steps you want skipped during the job run. Click here for detailed instructions. - They allow you to add data to DD* input itreams. Click here for detailed instructions.
Automatic Output Retrieval-zOS	Optional. Allows you to specify whether you want the system to automatically retrieve output from the job and attach it to the task instance record. Options: - **None** - Do not attach any output to the task instance record. - **File** - Attach the file specified in the Output File field. - **Joblog** - Attach output from the z/OS joblog.
Output File-Auto Retrieve	Required if Automatic Output Retrieval=File. This field specifies the path and filename containing the output that you want automatically retrieved and attached to the task instance.
Start Line	Optional. Allows you to instruct the system to fetch data beginning at the line indicated.
Number of Lines	Optional. Allows you to limit the retrieved data to the number of lines specified. The default is 100. You can change the default by modifying the record **Configuration > Properties > Retrieve Output Default Maximum Lines**.
Scan Text-Automatic Output Retrieval	Optional. Instructs the system to scan the data for the text specified and retrieve only that. Opswise will process this field as a regular expression.

Task Priority	Task instance only; the priority of this task instance, as set by the user via the Set Priority command. Options are: HIGH, MEDIUM, LOW.
Late Start	If enabled, and if the task instance starts after the time specified in the Late Start Time field (see below), the task instance is flagged as late. To determine whether a task instance started late, open the task instance and locate the Started Late field; the field is checked if the instance started late.
Late Start Type	Required if Late Start is enabled. Options are: • **Time** - Flag the task if it starts after the specified time. • **Duration** - Flag the task if it starts a certain amount of time after the programmed start time. The task must have a specific start time.
Late Start Time	Time after which the task start time is considered late. Use hh:mm, 24-hour time
Late Start Duration	Duration (amount of relative time) after which the task is considered to have started late. For a task within a workflow, the duration is the period between the time the workflow starts and the time the task itself starts. For example, a task might have a Late Start Duration of 60 minutes. If the workflow starts at 9:00 a.m. but the task itself does not start until 10:30, the task has started late. For a task that is not within a workflow, Late Start Duration has meaning only if the task has been held upon starting. For example, if a task has a Late Start Duration of 60 minutes and the Hold on Start field is enabled, if the task is not released from hold within the amount of time specified in the Late Start Duration field, the task has started late.
Started Late	Task instance only; system-supplied. This field is flagged if the task started later than the time specified in the Late Start fields.
Late Finish	If enabled, and if the task instance finishes after the time or period specified, the task instance is flagged as late. You can specify a time or duration to determine a late finish (see Late Finish Type below). To determine whether a task instance finished late, open the task instance and locate the Finished Late field; the field is checked if the instance finished after the specified time or lasted longer than expected. This field only appears on the task instance if the user specified a Late Finish in the task definition.
Late Finish Type	If Late Finish is enabled, use this field to specify whether you want to use time (see Late Finish Time below) or duration (see Late Finish Duration below) to determine whether the task instance finished late.
Late Finish Time	If Late Finish Type is Time, use this to specify the time after which the task finish time is considered late. Use hh:mm, 24-hour time.
Late Finish Duration	If Late Finish Type is Duration, use this to specify the longest amount of time this task instance should take to run. You can specify any combination of hours, minutes, and seconds.
Finished Late	Task instance only; system-supplied. This field is flagged if the task finished later than the time or duration specified in the Late Finish fields.
Early Finish	If enabled, and if the task instance finishes before the time or period specified, the task instance is flagged as early. You can specify a time or duration to determine an early finish (see Early Finish Type below). To determine whether a task instance finished early, open the task instance and locate the Finished Early field; the field is checked if the instance finished before the specified time or did not last as long as expected. This field only appears on the task instance if the user added Early Finish specifications to the task definition.

Early Finish Type	If Early Finish is enabled, use this field to specify whether you want to use time (see Early Finish Time below) or duration (see Early Finish Duration below) to determine whether the task instance finished early.
Early Finish Time	If Early Finish Type is Time, use this to specify the time before which the task finish time is considered early. That is, enter a time at which the task should still be running. Use hh:mm, 24-hour time.
Early Finish Duration	If Early Finish Type is Duration, use this to specify the shortest amount of time this task instance should take to run. You can specify any combination of hours, minutes, and seconds.
Finished Early	Task instance only; system-supplied. This field is flagged if the task finished earlier than the time specified in the Early Finish fields.
Maximum Retries	User-defined. The maximum number of times the system should retry this task after it has started and gone to a failed state.
Retry Indefinitely	User-defined. Enabled or disabled. Indicates whether the system should continue trying indefinitely to run this task. If you enable this field, it overrides any value placed in the Maximum Retries field (above).
Retry Interval	User-defined. The number of seconds between each retry.
Current Retry Count	Task instance only; system-supplied. Displays only for a running task instance. Displays the current number of times the system has retried the task after it first went to failure status.
User Estimated End Time	Task instance only; system-supplied. If the user entered information into the User Estimated Duration field in the task definition, the system uses this information to calculate an end time for the task instance, based on the date/time the task instance started.
Shortest Estimated End Time	Task instance only; system-supplied.
Average Estimated End Time	Task instance only; system-supplied.
Longest Estimated End Time	Task instance only; system-supplied.
Submit button	Submits the new record to the database.
Update button	Saves updates to the record.
Launch Task button	Manually launches the task.
Delete button	Deletes the current record.
View Parent button	Task instance only; Displays this task's parent task (workflow), if any.

Show Details button	Task instance only; displays detailed information about this task instance.
Skip button	Task instance only; for tasks loaded into the schedule that haven't run yet. Allows you to tell the system to skip this task. See Skipping a Task.
Hold button	Task instance only; see Putting a Task on Hold.
Force Finish button	Task instance only; see Force Finishing a Task.
Re-run button	Task instance only; see Re-running a Task.
Output tab	Task instance only. Displays output generated from the process, if any, based on specifications provided by the user in the Automatic Output Retrieval fields in the task definition.
z/OS Restartable JobSteps tab	Task instance only; see Rerunning a z/OS Task and Confirming any JCL Changes.
z/OS Restart Confirmation tab	Task instance only; see Rerunning a zOS Task and Confirming any JCL Changes.
Task Virtual Resources tab	Lists Virtual Resources to which this task is assigned. See Creating Virtual Resources.
Variables tab	Displays all variables associated with this record.
Actions tab	Allows you to specify actions that the system will take automatically based on events that occur during the execution of this task. Events are task instance status, exit codes, late start, late finish, and early finish. Actions are: • **Abort Action** - Abort the task if certain events occur. For details, see Setting Up Abort Actions. • **Email Notification** - Send an email if certain events occur. For details, see Creating Email Notifications. • **SNMP Notification** - Send an email if certain events occur. For details, see Creating SNMP Notifications. • **Set Variable** - Used in workflows to set a variable based on the occurrence of certain events. For details, see Setting Variables in a Workflow.
Step Conditions tab	Displays a list of all step conditions defined for this task.
Task Instances tab	System-supplied. Displays a list of all task instances generated by this task definition.
Triggers tab	Displays a list of all triggers that have been defined to launch this task. Also allows you to add new triggers. If you add a new trigger from this location, the system automatically constructs a default trigger name as follows: <current task name>#TRIGGER#. You can change the default name if desired. For instructions on creating triggers, see Creating Triggers.
Notes tab	Displays all notes associated with this task. See Runbook Notes.
Versions tab	Stores copies of all previous versions of the current record (see Record Versioning).

Specifying When a Task Runs

You can run the task as part of a workflow, specify triggers that run the task automatically based on times or events, or run the task manually.

Monitoring Task Execution

You can monitor all system activity from the Activity display and can view activity history from the Activity History screen.

Special Processing on z/OS Tasks

The following special processing features are available for running z/OS tasks:

- Adding Data to DD* Input Streams
- Skipping Steps during Initial Run
- Overriding Key JCL Parameters from Opswise

The following failure processing features are available for handling job failures:

- Rerunning a z/OS Task
- Interactively Ignoring a Stepcode to Force a Task to Complete

Opswise supports compatibility with other schedulers:

- CA7/CA11 Toleration

Opswise provides the following reports that track error processing:

- Viewing Ops Rerun Reports
- Viewing Audit Trails on a Restart

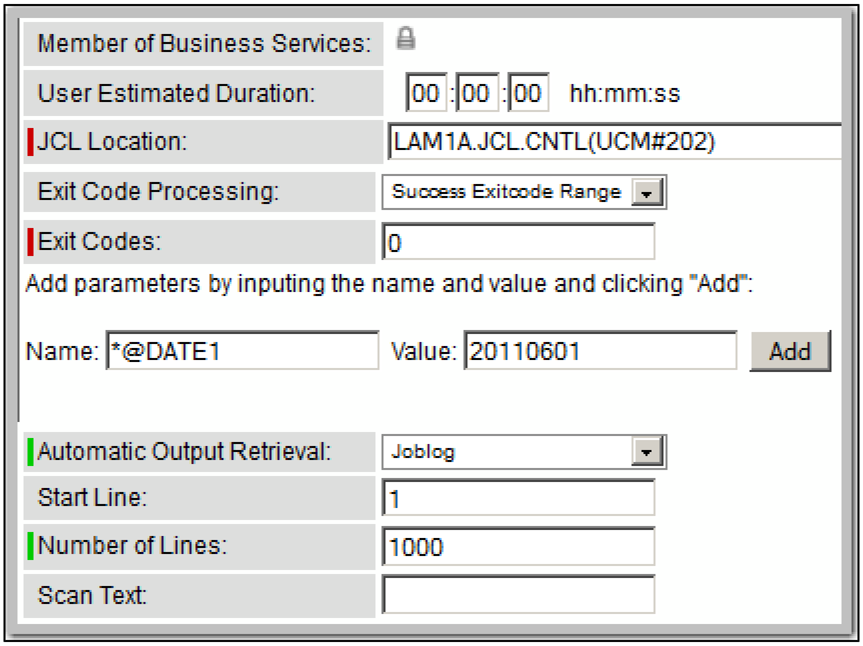
Each of these features is described in detail below.

Adding Data to DD* Input Streams

This feature allows you to use the Parameters fields to pass variable data into the input data stream in the JCL. Follow these steps:

1. Open the task record.
2. Use the **Add Parameters** feature to add the first parameter in the following format:
    - Name is the name of a variable preceded with *@. For example: *@DATE1
    - Value is the value you want to set to the variable. For example: 20110601
3. Click the **Add** button. The new parameter is added to your parameter list.
4. Repeat the above step for each variable you want to set.

For example, you could specify the following in the Add Parameters field:



Member of Business Services:

User Estimated Duration: 00 : 00 : 00 hh:mm:ss

JCL Location: LAM1A.JCL.CNTL(UCM#202)

Exit Code Processing: Success Exitcode Range

Exit Codes: 0

Add parameters by inputing the name and value and clicking "Add":

Name: *@DATE1 Value: 20110601 Add

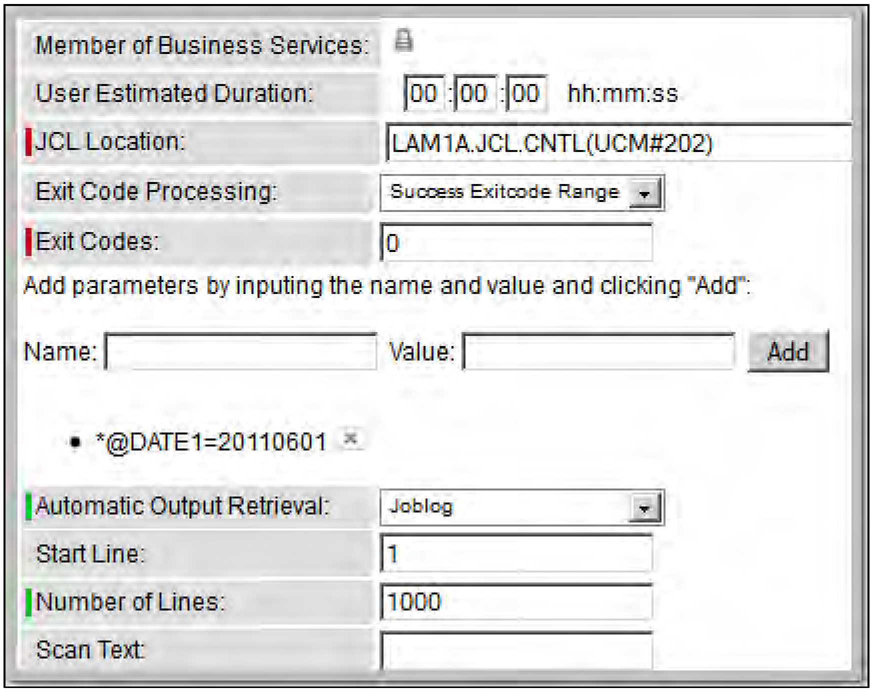
Automatic Output Retrieval: Joblog

Start Line: 1

Number of Lines: 1000

Scan Text:

After you click the Add button, the screen looks like this:



Member of Business Services:

User Estimated Duration: 00 : 00 : 00 hh:mm:ss

JCL Location: LAM1A.JCL.CNTL(UCM#202)

Exit Code Processing: Success Exitcode Range

Exit Codes: 0

Add parameters by inputing the name and value and clicking "Add":

Name: Value: Add

- *@DATE1=20110601 ×

Automatic Output Retrieval: Joblog

Start Line: 1

Number of Lines: 1000

Scan Text:

When the JCL is submitted for execution, the parameter @DATE1 (shown in the example below) will be substituted with the value 20110601 in the JCL input stream:

```
//INPUT DD *
@DATE1

/*
```

Skipping Steps during Initial Run

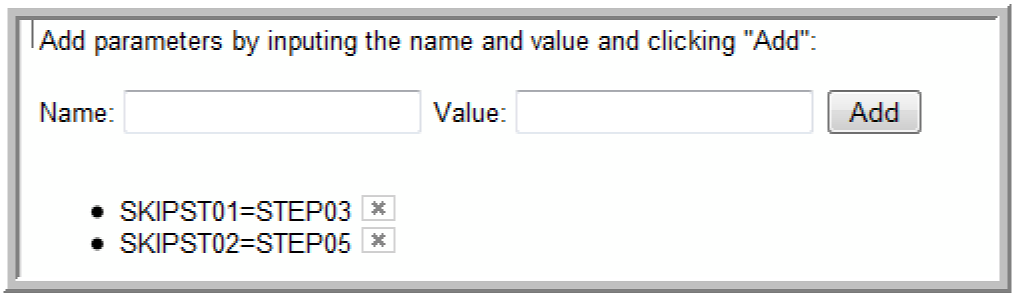
You can specify in a z/OS task that one or more steps from the JCL should be skipped when Opswise launches the job. You achieve this by adding SKIPSTNN variables (or parameters) to your z/OS task record.

To configure your z/OS task to skip specific JCL steps:

1. Open the task record.
2. Use the 'add parameters' feature to add the first parameter in the following format:

```
SKIPSTAA=STEPNAME
```

   Where
   - SKIPST is a required string.
   - AA is any unique combination of alphanumerics, used only to make this SKIPST command unique. (You can add as many SKIPST commands as needed.)
   - STEPNAME is the JCL step name.
3. Click the **Add** button. The new parameter is added to your parameter list.
4. Repeat the above step for each step you want to skip. Change the AA portion of the SKIPST parameter for each parameter you add. That is, each parameter name must be unique. In the example below, two skipstep parameters have been added to the record, instructing Opswise to skip JCL STEP03 and STEP05.



Add parameters by inputing the name and value and clicking "Add":

- SKIPST01=STEP03 ☒
- SKIPST02=STEP05 ☒

To delete a parameter, click the **X**.

Overriding Key JCL Parameters from Opswise

When you launch a z/OS task from Opswise, you can specify a different Jobname, Jobclass, Msgclass, Schedule ID or add a JOBPARM card. This enables you to run your JCL jobs from Opswise without having to go in and modify your JCL. You can do so by entering the new value into the appropriate field on the z/OS Task definition screen. See the screen shot under Creating a New z/OS Task for the location of these fields.

Rerunning a z/OS Task

Basic Rules for a Restart

When you rerun a z/OS task, Opswise allows you to rerun jobsteps using the methods described below. The Restartable Steps tab provides a list of steps and highlights in green from which steps you can start the rerun.

When you rerun a z/OS task, the system automatically performs the following:

- Deletes data sets that were created in dependent steps.
- Maintains Generation data group.

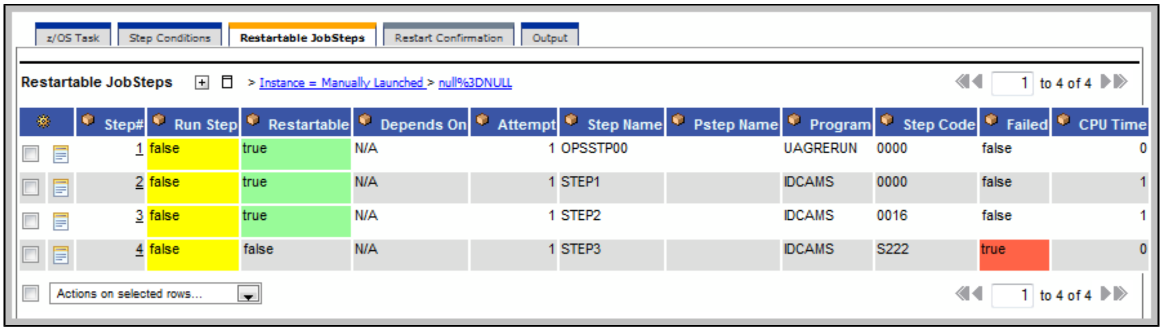
> **Note**
> You must restart the z/OS task from Opswise in order for these clean-up procedures to be performed. Do not restart the task from the z/OS prompt.

Confirming Any JCL Changes

If you make any JCL changes, Opswise will prompt you for a confirmation, as described in the procedures below.

To Restart a z/OS Task from a Specific Step

1. Open the z/OS task instance by clicking its name on the Activity screen.
2. Click the **Restartable JobSteps** tab. This displays each step of the z/OS job on a separate line in the list. If applicable, the Failed column indicates "true", highlighted in red, to show where the job failed.
3. Assuming the failure was caused by error(s) in the JCL, examine the JCL and make your corrections. Opswise uses a background process to determine whether changes have been made to the JCL. Any changes trigger the confirmation process.
4. When you have corrected the JCL, consult the **Restartable** column to determine which steps are available as restart points. A value of True means you can restart from this JCL step; a value of False means you cannot restart from this step. If you choose to rerun specific steps, you can only run steps flagged as True in the Restartable column. In the example shown below, the job failed at step 4 and the latest step you can restart from is step 3.



Restartable JobSteps > Instance = Manually Launched > null%3DNULL 1 to 4 of 4

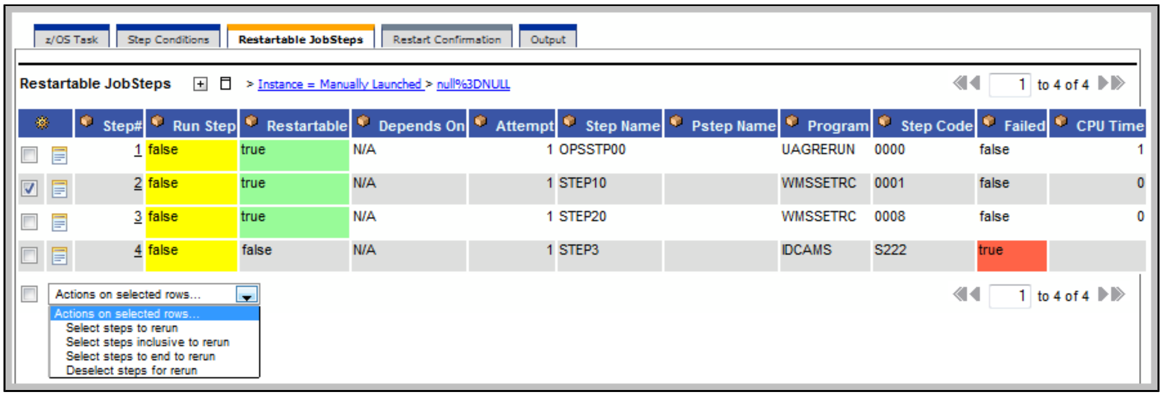
		Step#	Run Step	Restartable	Depends On	Attempt	Step Name	Pstep Name	Program	Step Code	Failed	CPU Time
□		1	false	true	N/A	1	OPSSTP00		UAGRERUN	0000	false	0
□		2	false	true	N/A	1	STEP1		IDCAMS	0000	false	1
□		3	false	true	N/A	1	STEP2		IDCAMS	0016	false	1
□		4	false	false	N/A	1	STEP3		IDCAMS	S222	true	0

□ Actions on selected rows... 1 to 4 of 4

5. You can restart the job from a specific step or select specific steps to re-run.

   To restart the task from a specific step and run it to the end:

   a. Click the box to the left of the step from which you want to restart the task, as shown in the illustration below. For example, to restart the job from step 2 to the end, select 2.



Restartable JobSteps > Instance = Manually Launched > null%3DNULL 1 to 4 of 4

		Step#	Run Step	Restartable	Depends On	Attempt	Step Name	Pstep Name	Program	Step Code	Failed	CPU Time
□		1	false	true	N/A	1	OPSSTP00		UAGRERUN	0000	false	1
☑		2	false	true	N/A	1	STEP10		WMSSETRC	0001	false	0
□		3	false	true	N/A	1	STEP20		WMSSETRC	0008	false	0
□		4	false	false	N/A	1	STEP3		IDCAMS	S222	true	

□ Actions on selected rows... 1 to 4 of 4

Actions on selected rows...
Select steps to rerun
Select steps inclusive to rerun
Select steps to end to rerun
Deselect steps for rerun

   b. Click the down-arrow in **Actions on selected rows...** and select **Select steps to end to rerun**.
   c. Return to the z/OS Task main screen and click the **Re-run** button.

   To re-run one or more specific steps:

   a. Click the box to the left of the steps you want to re-run.
   b. Click the down-arrow in **Actions on selected rows...** and select **Select steps to rerun**.
   c. Return to the z/OS Task main screen and click the **Re-run** button.

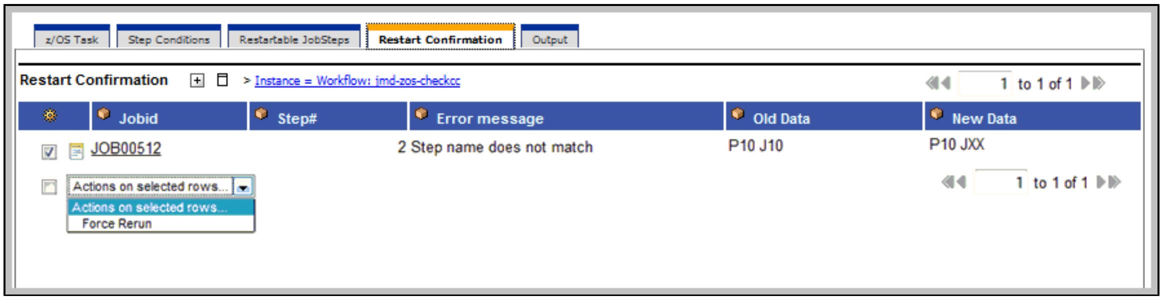
   To re-run a range of steps:

   a. Click the boxes to the left of the first and last steps you want to re-run. For example, to run steps 1 through 3, click 1 and 3.
   b. Click the down-arrow in **Actions on selected rows...** and select **Select steps inclusive to rerun**.
   c. Return to the z/OS Task main screen and click the **Re-run** button.

   To start over before clicking on the Re-run button, click the down-arrow in **Actions on selected rows..**. and select **Deselect steps for rerun**.

6. Once you have made your restart selection, Opswise reruns the task.
7. If you have made changes to the JCL, the task goes into a status of **Confirmation Required** on the Activity screen. Confirm the change as follows:
   a. From the Activity screen, click on the task name to open the record.

b. Click the **Restart Confirmation** tab. This displays a record for each change that was made to the JCL, with a description in the Error Message column, as shown in the following example.

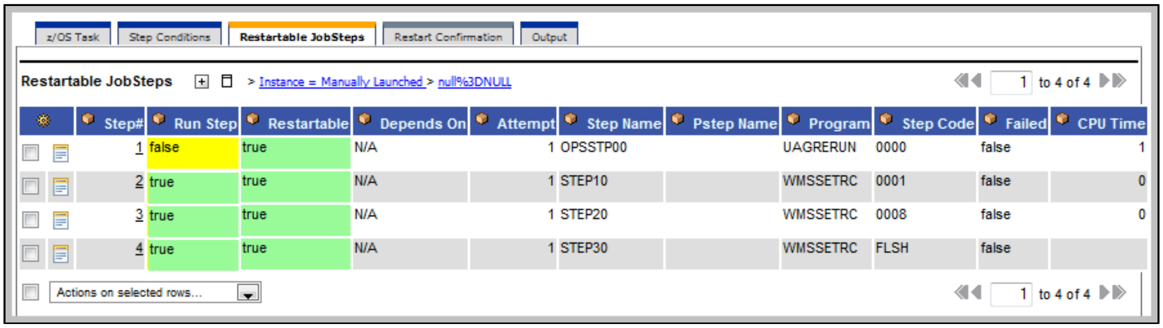


Restart Confirmation > Instance = Workflow: jmd-zos-checkcc 1 to 1 of 1

	Jobid	Step#	Error message	Old Data	New Data
☑	JOB00512	2	Step name does not match	P10 J10	P10 JXX

c. To confirm the change(s), click the box to the left of each listed change and click the **Actions on selected rows**.
d. Select **Force Rerun**.
e. Return to the z/OS Task main screen and click the **Re-run** button.

8. After the rerun is complete, the Failed column should show false for each step.



Restartable JobSteps > Instance = Manually Launched > null%3DNULL 1 to 4 of 4

	Step#	Run Step	Restartable	Depends On	Attempt	Step Name	Pstep Name	Program	Step Code	Failed	CPU Time
□	1	false	true	N/A	1	OPSSTP00		UAGRERUN	0000	false	1
□	2	true	true	N/A	1	STEP10		WMSSETRC	0001	false	0
□	3	true	true	N/A	1	STEP20		WMSSETRC	0008	false	0
□	4	true	true	N/A	1	STEP30		WMSSETRC	FLSH	false	

> If you repeat the above process on the same task instance, the previous list of JCL changes, if any, is wiped clean and replaced with the most recent list of changes.

z/OS Restartable JobSteps Tab Column Descriptions

The following table describes each column on the z/OS Retartable Job Steps screen.

Column Name	Description
Step #	Number assigned to this step by Opswise.
Run Step	Indicates whether this step ran during the last run or restart of this job. For example, if you just re-started the job from step 4, steps 1, 2, and 3 would indicate false, and steps 4 to the end would indicate true.
Restartable	When the JCL job fails, Opswise determines the latest step you can restart from. A step indicating **True** and highlighted in green means you can restart from this step.
Depends On	Indicates which step or steps must be completed successfully before you can run this step.
Attempt	Number of times this step has been run or attempted to run.
Step Name	Extracted from the JCL. The name of the JCL step.
Pstep Name	Process step name from within the JCL step.
Program	Name of the program being executed by the step.
Step Code	Extracted from the JCL. Exit code for this step of the program.
Failed	True or false. Indicates whether or not this step failed. True means the step failed.

CPU Time	Number of CPU seconds it took for the JCL step to run.
IO Total	Total input/output operations for this step.
Memory Peak	Peak amount of memory used during the execution of this step.

z/OS Restart Confirmation Tab Column Descriptions

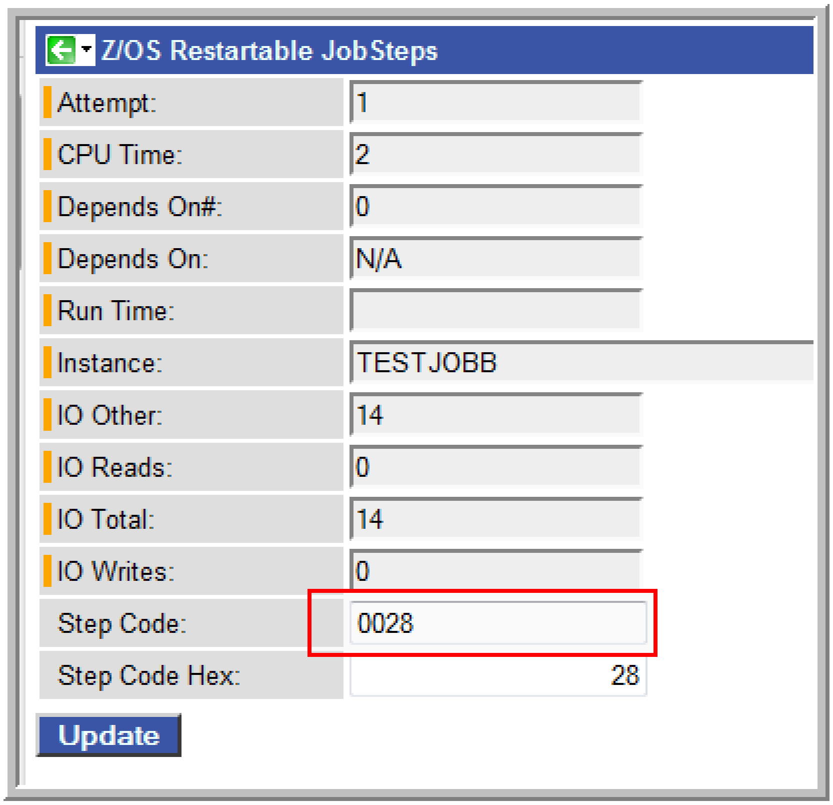
The following table describes each column on the Z/OS Retartable Job Steps screen.

Column Name	Description
Jobid	Number assigned to this step by Opswise.
Step#	JCL step number that was modified.
Error message	Description of the change.
Old Data	JCL before the change.
New Data	JCL after the change.

Interactively Ignoring a Stepcode to Force a Task to Complete

If the exit code on a previous step is causing a step failure and you have determined that you want to finish the job run anyway, you can change the exit code to force finish the task.

1. Open the task instance from the Activity screen.
2. From the **Restartable Steps** tab, click on the step whose exit code you want to change. Opswise opens the record for this step, as shown in the following example:



Z/OS Restartable JobSteps

Attempt:	1
CPU Time:	2
Depends On#:	0
Depends On:	N/A
Run Time:	
Instance:	TESTJOBB
IO Other:	14
IO Reads:	0
IO Total:	14
IO Writes:	0
Step Code:	0028
Step Code Hex:	28

3. Enter the new code in the Step Code field and click **Update**.
4. Return to the main task screen and click **Rerun**.

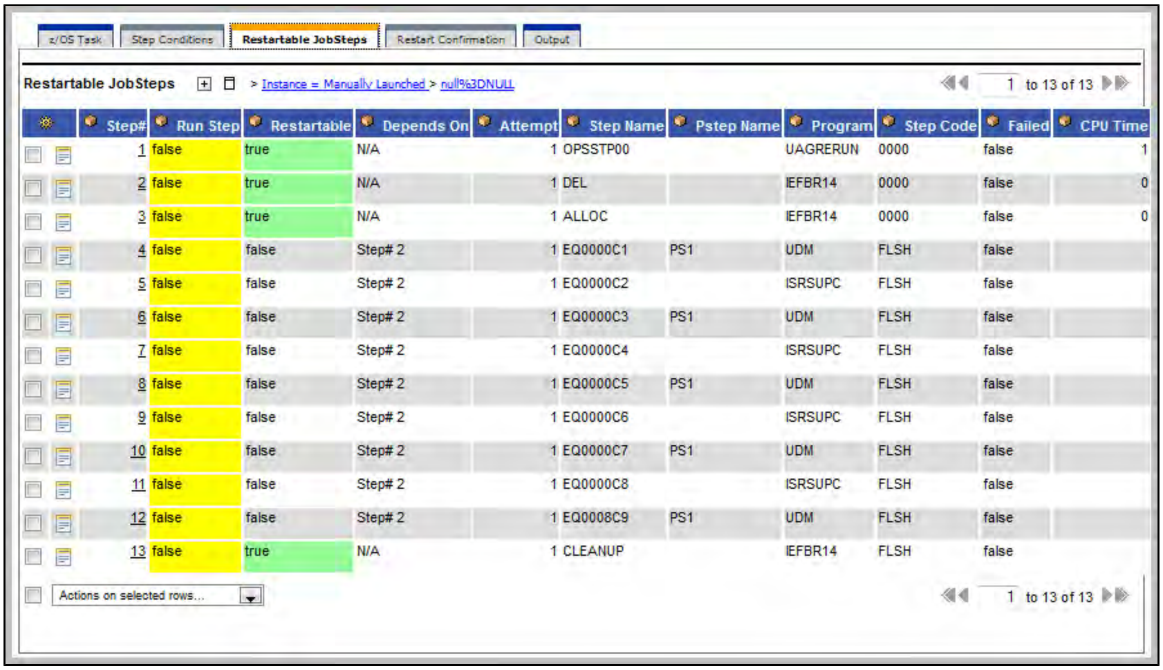
CA7/CA11 Toleration

Non-Restartable Steps

Opswise can read and interpret JCL step names that have been customized for CA11 and UCC. This allows you to launch your existing CA11 and UCC jobs from Opswise without modifying the JCL. When Opswise encounters one of the step names listed below in your JCL, Opswise will skip the step during a restart:

- CA11NR - CA11 Non-Restartable
- UCC11NR - UCC11 Non-Restartable
- OPSNR000 - Opswise Non-Restartable

In the example shown below, steps 4 to 12 each have one of the above DD Names and are therefore non-restartable steps.



Restartable JobSteps > Instance = Manually Launched > null%3DNULL   1 to 13 of 13

	Step#	Run Step	Restartable	Depends On	Attempt	Step Name	Pstep Name	Program	Step Code	Failed	CPU Time
□	1	false	true	N/A	1	OPSSTP00		UAGRERUN	0000	false	1
□	2	false	true	N/A	1	DEL		IEFBR14	0000	false	0
□	3	false	true	N/A	1	ALLOC		IEFBR14	0000	false	0
□	4	false	false	Step# 2	1	EQ0000C1	PS1	UDM	FLSH	false	
□	5	false	false	Step# 2	1	EQ0000C2		ISRSUPC	FLSH	false	
□	6	false	false	Step# 2	1	EQ0000C3	PS1	UDM	FLSH	false	
□	7	false	false	Step# 2	1	EQ0000C4		ISRSUPC	FLSH	false	
□	8	false	false	Step# 2	1	EQ0000C5	PS1	UDM	FLSH	false	
□	9	false	false	Step# 2	1	EQ0000C6		ISRSUPC	FLSH	false	
□	10	false	false	Step# 2	1	EQ0000C7	PS1	UDM	FLSH	false	
□	11	false	false	Step# 2	1	EQ0000C8		ISRSUPC	FLSH	false	
□	12	false	false	Step# 2	1	EQ0008C9	PS1	UDM	FLSH	false	
□	13	false	true	N/A	1	CLEANUP		IEFBR14	FLSH	false	

□ Actions on selected rows...   1 to 13 of 13

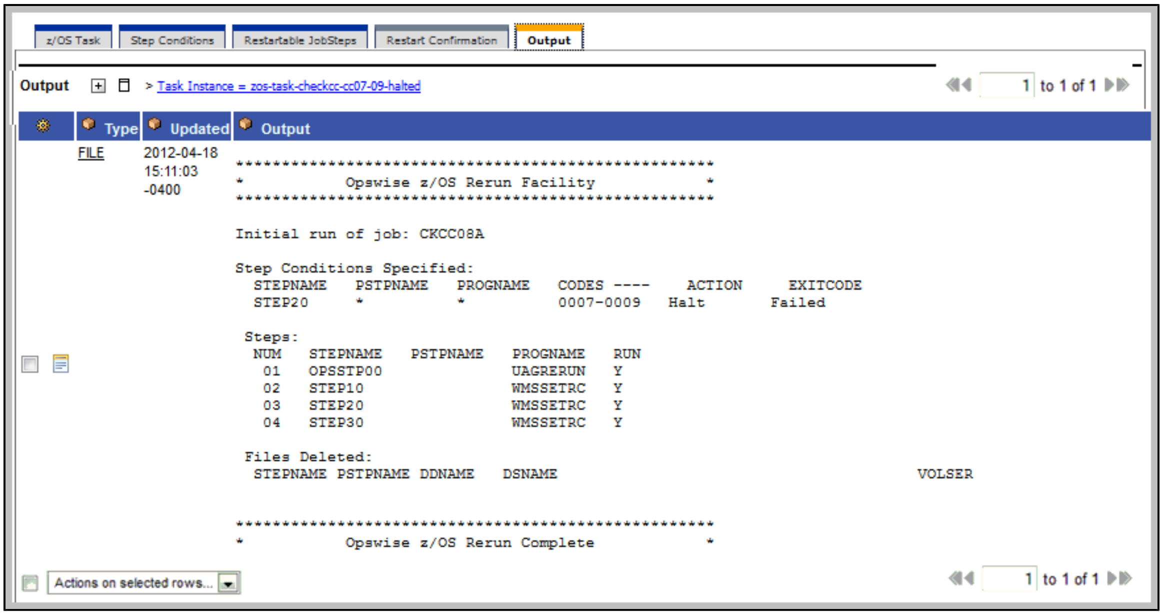
SCHID - Overriding the CA7 Schedule ID

The Schedule ID field allows you to override the CA7 SCHID, or Schedule ID. For example, the JCL shown below contains CA7 Scheduled Overrides statements #JI and #JEND. This JCL will set CLASS =A if the SCHID is between 1 thru 39, and set CLASS=B if the SCHID is between 40 thru 79. The user can set the SCHID by entering it into the Schedule ID field on the z/OS Task definition screen. The Opswise agent scans for #JI and #JEND, and generates the appropriate JCL, as shown in the example below.

```
//SCHID JOB (IMS,001),JIM,MSGCLASS=X,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
#JI,ID=1-39
//             CLASS=A
#JEND
#JI,ID=40-79
//             CLASS=B
#JEND
//S1      EXEC PGM=IGWSPZAP
//SYSLIB   DD DSN=OPS01.JS01.LOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
DUMPT WMSSETRC WMSSETRC
/*
//
```

Viewing Ops Rerun Reports

Opswise keeps a detailed record of task restarts. This data is written to the **Output** tab on the task instance record, as shown in the sample below:

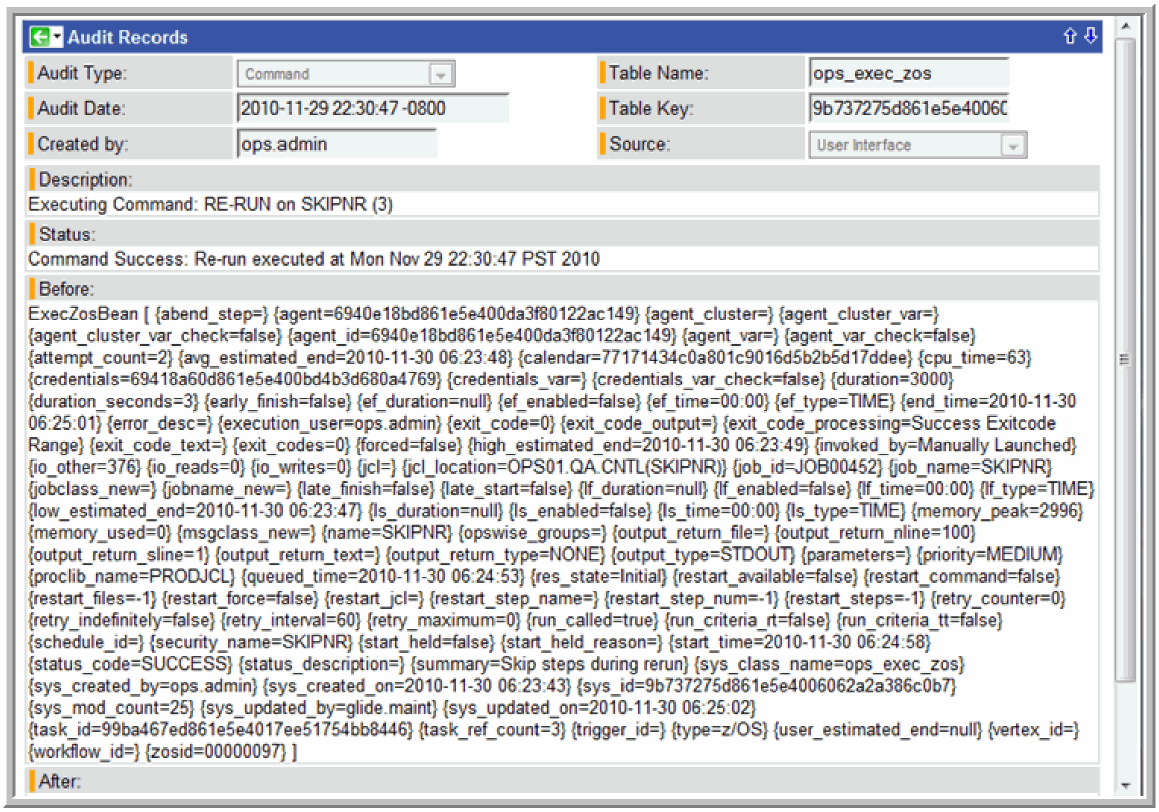


Type	Updated	Output
FILE	2012-04-18 15:11:03 -0400	

```
*******************************************************
*           Opswise z/OS Rerun Facility               *
*******************************************************

Initial run of job: CKCC08A

Step Conditions Specified:
  STEPNAME    PSTPNAME    PROGNAME    CODES ----    ACTION      EXITCODE
  STEP20      *           *           0007-0009   Halt        Failed

 Steps:
  NUM    STEPNAME    PSTPNAME    PROGNAME    RUN
   01    OPSSTP00                UAGRERUN    Y
   02    STEP10                  WMSSETRC    Y
   03    STEP20                  WMSSETRC    Y
   04    STEP30                  WMSSETRC    Y

 Files Deleted:
  STEPNAME PSTPNAME DDNAME   DSNAME                                         VOLSER


*******************************************************
*           Opswise z/OS Rerun Complete               *
```

Viewing Audit Trails on a Restart

Opswise maintains detailed audit records on all system activity. The sample below shows an audit record for a re-run on a z/OS task called SKIPNR.



Audit Type:	Command	Table Name:	ops_exec_zos
Audit Date:	2010-11-29 22:30:47 -0800	Table Key:	9b737275d861e5e4006
Created by:	ops.admin	Source:	User Interface

Description:
Executing Command: RE-RUN on SKIPNR (3)

Status:
Command Success: Re-run executed at Mon Nov 29 22:30:47 PST 2010

Before:
ExecZosBean [{abend_step=} {agent=6940e18bd861e5e400da3f80122ac149} {agent_cluster=} {agent_cluster_var=} {agent_cluster_var_check=false} {agent_id=6940e18bd861e5e400da3f80122ac149} {agent_var=} {agent_var_check=false} {attempt_count=2} {avg_estimated_end=2010-11-30 06:23:48} {calendar=77171434c0a801c9016d5b2b5d17ddee} {cpu_time=63} {credentials=69418a60d861e5e400bd4b3d680a4769} {credentials_var=} {credentials_var_check=false} {duration=3000} {duration_seconds=3} {early_finish=false} {ef_duration=null} {ef_enabled=false} {ef_time=00:00} {ef_type=TIME} {end_time=2010-11-30 06:25:01} {error_desc=} {execution_user=ops.admin} {exit_code=0} {exit_code_output=} {exit_code_processing=Success Exitcode Range} {exit_code_text=} {exit_codes=0} {forced=false} {high_estimated_end=2010-11-30 06:23:49} {invoked_by=Manually Launched} {io_other=376} {io_reads=0} {io_writes=0} {jcl=} {jcl_location=OPS01.QA.CNTL(SKIPNR)} {job_id=JOB00452} {job_name=SKIPNR} {jobclass_new=} {jobname_new=} {late_finish=false} {late_start=false} {lf_duration=null} {lf_enabled=false} {lf_time=00:00} {lf_type=TIME} {low_estimated_end=2010-11-30 06:23:47} {ls_duration=null} {ls_enabled=false} {ls_time=00:00} {ls_type=TIME} {memory_peak=2996} {memory_used=0} {msgclass_new=} {name=SKIPNR} {opswise_groups=} {output_return_file=} {output_return_nline=100} {output_return_sline=1} {output_return_text=} {output_return_type=NONE} {output_type=STDOUT} {parameters=} {priority=MEDIUM} {proclib_name=PRODJCL} {queued_time=2010-11-30 06:24:53} {res_state=Initial} {restart_available=false} {restart_command=false} {restart_files=-1} {restart_force=false} {restart_jcl=} {restart_step_name=} {restart_step_num=-1} {restart_steps=-1} {retry_counter=0} {retry_indefinitely=false} {retry_interval=60} {retry_maximum=0} {run_called=true} {run_criteria_rt=false} {run_criteria_tt=false} {schedule_id=} {security_name=SKIPNR} {start_held=false} {start_held_reason=} {start_time=2010-11-30 06:24:58} {status_code=SUCCESS} {status_description=} {summary=Skip steps during rerun} {sys_class_name=ops_exec_zos} {sys_created_by=ops.admin} {sys_created_on=2010-11-30 06:23:43} {sys_id=9b737275d861e5e4006062a2a386c0b7} {sys_mod_count=25} {sys_updated_by=glide.maint} {sys_updated_on=2010-11-30 06:25:02} {task_id=99ba467ed861e5e4017ee51754bb8446} {task_ref_count=3} {trigger_id=} {type=z/OS} {user_estimated_end=null} {vertex_id=} {workflow_id=} {zosid=00000097}]

After:

Indesca Task

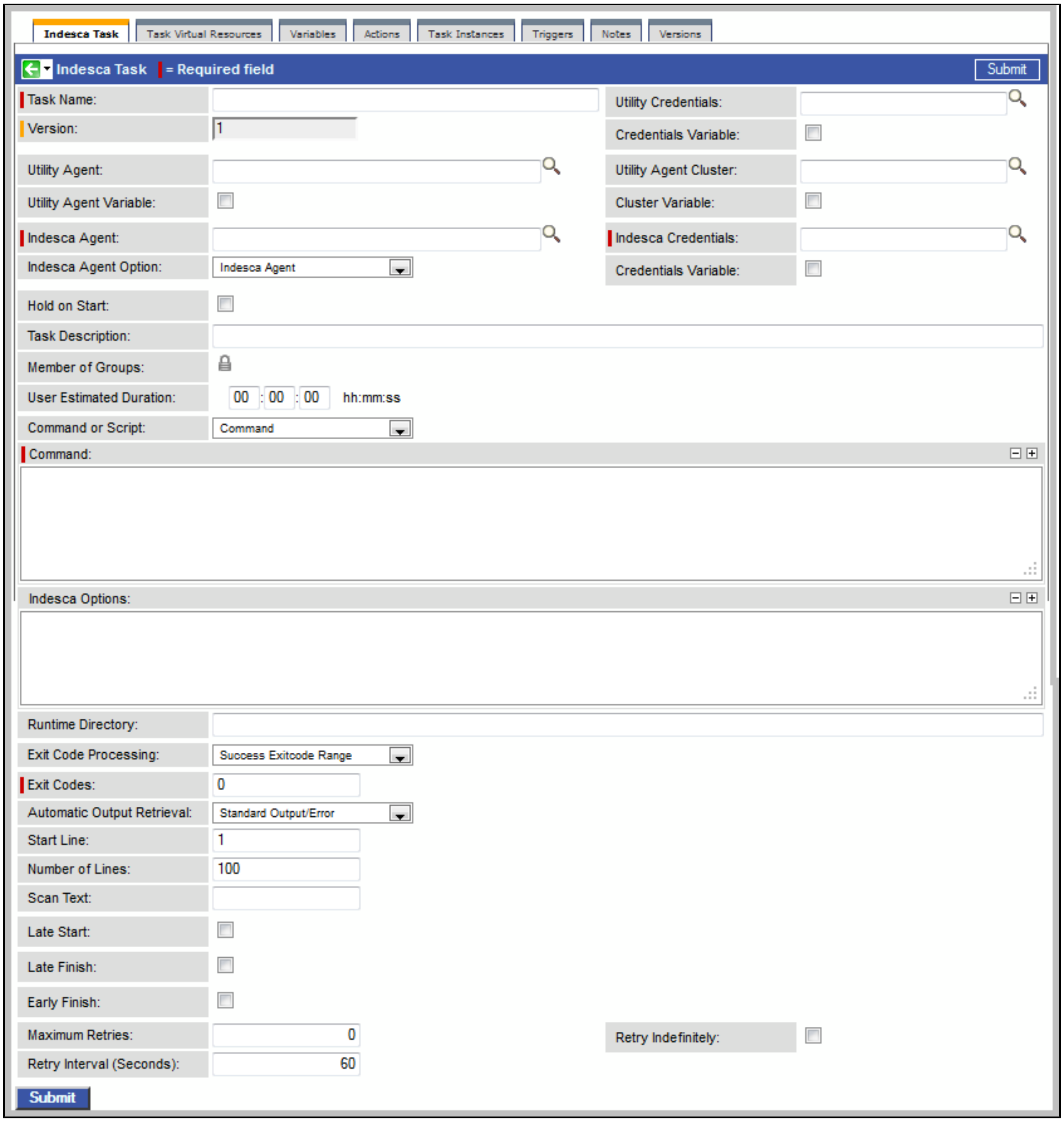
Before You Begin

The Indesca task allows you to run a platform-specific application on a machine where Indesca is running. From the Opswise perspective, Indesca serves as an agent process. Indesca runs on any platform including z/OS, Linux/Unix, and Windows. To run an Indesca task, you must first complete the following tasks:

- Install Indesca on the target machine.
- Launch the Indesca agent. When the agent connects with the core processor, it automatically creates an agent resource definition in the Opswise database.
- Optionally, customize the agent heartbeat and credentials, as described in Indesca Resource Field Definitions.

Creating a New Indesca Task

1. From the navigation pane, select **Automation Center > Tasks > Indesca Tasks**. The Indesca Tasks List screen displays.
2. Click **New**. The Indesca Task Definition screen displays.



Version: 1

Indesca Agent Option: Indesca Agent

User Estimated Duration: 00 : 00 : 00 hh:mm:ss

Command or Script: Command

Exit Code Processing: Success Exitcode Range

Exit Codes: 0

Automatic Output Retrieval: Standard Output/Error

Start Line: 1

Number of Lines: 100

Maximum Retries: 0

Retry Interval (Seconds): 60

3. Using the field descriptions provided below as a guide, complete the fields as needed.
4. Click the **Submit** button to save the record and return to the menu, or right-click on the title bar and select **Save** to save the record and remain on the current display.
5. If appropriate, repeat these steps for any additional tasks you want to add.

Task run statistics will appear after the first time this task has been launched.

Indesca Task Field Descriptions

The table below describes the fields, buttons, and tabs on the task definition and task instance screens. Color coding is provided that differentiates the following three types of fields:

- Fields that display on the task definition and task instance screens are shown in black.
- Fields that display only on the task definition screen are shown in green.
- Fields that display only on the task instance screen are shown in maroon.

Field Name	Description
Task/Instance Name	Required. Name used within Opswise to identify this task. Up to 40 alphanumerics; variables supported. It is the user's responsibility to develop a workable naming scheme for tasks.

Invoked by	Task instance only; system-supplied. How the task instance was launched. One of the following: - Trigger: (Trigger Name) - The instance was launched by the named trigger. - Workflow: (Workflow Name) - The instance was launched by the named workflow. - Manually Launched - The instance was launched by a user. To determine the name of the user: 1. From the Activity or All Task Instances screen, click the task instance name to open the record. 2. The Execution User field identifies the user who launched the task instance.
Task	Task instance only; system-supplied. Hover over the paper icon to display more information about the task instance. Click the paper icon to display the task definition record.
Execution User	Task instance only; system-supplied. If the task was launched manually, the ID of the user who launched it.
Instance Reference Id	Task instance only; system-supplied. The system increments this number each time the task is run.
Utility Credentials	Login credentials the Opswise agent will use to access the Indesca server machine.
Version	Task definition only; system-supplied. The version number of the current record, which is incremented by the system every time a user updates a record. Click on the Versions tab to view previous versions. For details, see Record Versioning.
Credentials Variable	Optional. If enabled, the Utility Credentials field (see above) converts from a reference field (where you browse and select a record) into a text field that allows you to enter a variable. Use the format: ${variable name}. . The variable must be a supported type as described in Using Variables and Functions.
Utility Agent	Required. The name of the Windows, Linux/Unix, or z/OS agent resource that will communicate with the Indesca agent. If you do not specify an agent, you must specify an agent cluster (see below).
Utility Agent Variable	Optional. If enabled, the Utility Agent field (see above) converts from a reference field (where you browse and select a record) into a text field that allows you to enter a variable. Use the format: ${variable name}. The variable must be a supported type as described in Using Variables and Functions.
Utility Agent Cluster	Optional. You can specify an agent cluster in addition to or in place of a specific agent. An agent cluster is a group of agents, one of which the system will choose to run this task. If you specify an agent and an agent cluster, Opswise Automation Center first tries to run the task on the specific agent. If the agent is not available, the system reverts to the agent cluster. See Configuring Agent Clusters for more information
Cluster Variable	Optional. If enabled, the Agent Cluster field (see above) converts from a reference field (where you browse and select a record) into a text field that allows you to enter a variable. Use the format: ${variable name}. The variable must be a supported type as described in Using Variables and Functions.
Indesca Agent	Required. Depending on the value in the Indesca Agent Option field (below), this field might contain a record name from the Indesca Agent table, a variable that will be resolved when the task is launched, or the hostname of a machine where the Indesca agent is running.
Indesca Credentials	Optional. The login credentials Opswise Automation Center will use to access the remote machine where the Indesca Agent is running. See Credentials.

Indesca Agent Option	Specifies how the name of the Indesca Agent is being supplied in the Indesca Agent field. Options: • **Indesca Agent** - The Indesca agent record is selected from the Indesca Agent table. • **Indesca Agent Variable** - The Indesca Agent field contains a variable that will be resolved when the task is launched. • **Indesca Agent Hostname** - The Indesca Agent field contains the hostname where the Indesca agent is running. The hostname must be accessible to Opswise.
Indesca Credentials Variable	Optional. If enabled, the Indesca (Agent) Credentials field (see above) converts from a reference field (where you browse and select a record) into a text field that allows you to enter a variable. Use the format: ${variablename}. The variable must be a supported type as described in Using Variables and Functions .
Hold on Start	If enabled, when the task is launched it appears in the Activity display with a status of Held. The task runs when the user releases it.
Hold Reason	Information about why the task will be put on hold when it starts.
Task Description	User-supplied description of this record.
Member of Groups	User-defined. Allows you to select one or more Opswise groups that this record definition belongs to. Click on the lock icon to unlock the field and select groups.
Status	Task instance only; system-supplied. See Task Statuses.
Exit Code	Task instance only; system-supplied. The exit code captured by the agent when executing the task (for example, a command or script).
Status Description	Task instance only; system-supplied. Provides additional information, if any, about the status of the task.
Queued Time	Task instance only; system-supplied. The time the task was queued for processing.
Process ID	Task instance only; system-supplied. The ID of the process that was launched.
Start Time	Task instance only; system-supplied. The date and time the task started.
CPU Time	Task instance only; system-supplied. The amount of CPU time the task took to run.
End Time	Task instance only; system-supplied. The date and time the task instance completed.
Duration	Task instance only; system-supplied. The amount of time the task took to run.
User Estimated Duration	Task definition only; optional. The estimated amount of time it should normally take to run this task. The system uses this information to calculate the User Estimated End Time on a task instance record.

Command or Script	Specifies whether a single command or a script is being executed. Options: • Command (default) • Script
Command	Command being executed on the remote machine. Variables supported.
Script Options	Optional. One or more command line options to pass to the script file.
Script File	Path and filename of the script file that will be executed on the remote machine.
Indesca Options	Optional. Any Indesca options needed by the program to execute properly. Variables supported.
Runtime Directory	Optional. The directory from which the application should be executed. Variables supported.
Exit Code Processing	Required. Specifies how Opswise Automation Center should determine whether the executed command failed or completed successfully. Options: • **Success Exitcode Range** - The command is considered completed successfully if its exit code falls within the range specified in the Exit Codes field (see below). • **Failure Exitcode Range** - The command is considered failed if its exit code falls within the range specified in the Exit Codes field (see below). • **Success Output Contains** - The command is considered completed successfully if its output contains the text specified in the Scan Output For field (see below). • **Failure Output Contains** - The command is considered failed if its output contains the text specified in the Scan Output For field (see below). • **Step Conditions** - The command is considered completed successfully/failed if any of its specified condition codes falls within the range specified under the Step Conditions tab (see Creating Step Conditions).
Output Type-Exit Code	Required if Exit Code Processing = Success Output Contains or Failure Output Contains. This field specifies the type of output. Options: • Standard Output (STDOUT) • Standard Error (STDERR) • File
Exit Codes	Required if Exit Code Processing = Success Exitcode Range or Failure Exit Code Range. This field specifies the range. Format: Numeric. Use commas to list a series of exitcodes; use hyphens to specify a range. Example: 1,5, 22-30.
Scan Output For	Required if Exit Code Processing = Success Output Contains or Failure Output Contains. This field specifies the text for which Opswise should scan the output file. Opswise will process this field as a regular expression.
Output File-Exit Codes	Required if Output Type = File. This field specifies the path and file name of the output file that should be scanned for the text in the Scan Output For field.
Automatic Output Retrieval	Optional. Allows you to specify whether you want the system to automatically retrieve any output from the job and attach it to the task instance record. Options: • **None** - Do not attach any output to the task instance record. • **Standard Output** - Attach all standard output. • **Standard Error** - Attach standard error output. • **File** - Attach the file specified in the Output File field.

Output File-Auto Retrieve	Required if Automatic Output Retrieval=File. This field specifies the path and filename containing the output that you want automatically retrieved and attached to the task instance.
Start Line	Optional. Allows you to instruct the system to fetch data beginning at the line indicated.
Number of Lines	Optional. Allows you to limit the retrieved data to the number of lines specified. The default is 100. You can change the default by modifying the record **Configuration > Properties > Retrieve Output Default Maximum Lines**.
Scan Text-Automatic Output Retrieval	Optional. Instructs the system to scan the data for the text specified and retrieve only that. Opswise will process this field as a regular expression.
Late Start	If enabled, and if the task instance starts after the time specified in the Late Start Time field (see below), the task instance is flagged as late. To determine whether a task instance started late, open the task instance and locate the Started Late field; the field is checked if the instance started late.
Late Start Type	Required if Late Start is enabled. Options are: • **Time** - Flag the task if it starts after the specified time. • **Duration** - Flag the task if it starts a certain amount of time after the programmed start time. The task must have a specific start time.
Late Start Time	Time after which the task start time is considered late. Use hh:mm, 24-hour time
Late Start Duration	Duration (amount of relative time) after which the task is considered to have started late. For a task within a workflow, the duration is the period between the time the workflow starts and the time the task itself starts. For example, a task might have a Late Start Duration of 60 minutes. If the workflow starts at 9:00 a.m. but the task itself does not start until 10:30, the task has started late. For a task that is not within a workflow, Late Start Duration has meaning only if the task has been held upon starting. For example, if a task has a Late Start Duration of 60 minutes and the Hold on Start field is enabled, if the task is not released from hold within the amount of time specified in the Late Start Duration field, the task has started late.
Started Late	Task instance only; system-supplied. This field is flagged if the task started later than the time specified in the Late Start fields.
Late Finish	If enabled, and if the task instance finishes after the time or period specified, the task instance is flagged as late. You can specify a time or duration to determine a late finish (see Late Finish Type below). To determine whether a task instance finished late, open the task instance and locate the Finished Late field; the field is checked if the instance finished after the specified time or lasted longer than expected. This field only appears on the task instance if the user specified a Late Finish in the task definition.
Late Finish Type	If Late Finish is enabled, use this field to specify whether you want to use time (see Late Finish Time below) or duration (see Late Finish Duration below) to determine whether the task instance finished late.
Late Finish Time	If Late Finish Type is Time, use this to specify the time after which the task finish time is considered late. Use hh:mm, 24-hour time.
Late Finish Duration	If Late Finish Type is Duration, use this to specify the longest amount of time this task instance should take to run. You can specify any combination of hours, minutes, and seconds.

Finished Late	Task instance only; system-supplied. This field is flagged if the task finished later than the time or duration specified in the Late Finish fields.
Early Finish	If enabled, and if the task instance finishes before the time or period specified, the task instance is flagged as early. You can specify a time or duration to determine an early finish (see Early Finish Type below). To determine whether a task instance finished early, open the task instance and locate the Finished Early field; the field is checked if the instance finished before the specified time or did not last as long as expected. This field only appears on the task instance if the user added Early Finish specifications to the task definition.
Early Finish Type	If Early Finish is enabled, use this field to specify whether you want to use time (see Early Finish Time below) or duration (see Early Finish Duration below) to determine whether the task instance finished early.
Early Finish Time	If Early Finish Type is Time, use this to specify the time before which the task finish time is considered early. That is, enter a time at which the task should still be running. Use hh:mm, 24-hour time.
Early Finish Duration	If Early Finish Type is Duration, use this to specify the shortest amount of time this task instance should take to run. You can specify any combination of hours, minutes, and seconds.
Finished Early	Task instance only; system-supplied. This field is flagged if the task finished earlier than the time specified in the Early Finish fields.
Task Priority	Task instance only; the priority of this task instance, as set by the user via the Set Priority command. Options are: HIGH, MEDIUM, LOW.
User Estimated End Time	Task instance only; system-supplied. If the user entered information into the User Estimated Duration field in the task definition, the system uses this information to calculate an end time for the task instance, based on the date/time the task instance started.
Shortest Estimated End Time	Task instance only; system-supplied.
Average Estimated End Time	Task instance only; system-supplied.
Longest Estimated End Time	Task instance only; system-supplied.
Maximum Retries	User-defined. The maximum number of times the system should retry this task after it has started and gone to a failed state.
Retry Indefinitely	User-defined. Enabled or disabled. Indicates whether the system should continue trying indefinitely to run this task. If you enable this field, it overrides any value placed in the Maximum Retries field (above).
Retry Interval	User-defined. The number of seconds between each retry.
Current Retry Count	Task instance only; system-supplied. Displays only for a running task instance. Displays the current number of times the system has retried the task after it first went to failure status.

Submit button	Submits the new record to the database.
Update button	Saves updates to the record.
Launch Task button	Manually launches the task.
Delete button	Deletes the current record.
View Parent button	Task instance only; Displays this task's parent task (workflow), if any.
Show Details button	Task instance only; displays detailed information about this task instance.
Skip button	Task instance only; for tasks loaded into the schedule that haven't run yet. Allows you to tell the system to skip this task. See Skipping a Task.
Hold button	Task instance only; see Putting a Task on Hold.
Force Finish button	Task instance only; see Force Finishing a Task.
Re-run button	Task instance only; see Re-running a Task.
Task Virtual Resources tab	Lists Virtual Resources to which this task is assigned. See Creating Virtual Resources.
Variables tab	Displays all variables associated with this record.
Actions tab	Allows you to specify actions that the system will take automatically based on events that occur during the execution of this task. Events are task instance status, exit codes, late start, late finish, and early finish. Actions are: • **Abort Action** - Abort the task if certain events occur. For details, see Setting Up Abort Actions. • **Email Notification** - Send an email if certain events occur. For details, see Creating Email Notifications. • **SNMP Notification** - Send an email if certain events occur. For details, see Creating SNMP Notifications. • **Set Variable** - Used in workflows to set a variable based on the occurrence of certain events. For details, see Setting Variables in a Workflow.
Task Instances tab	System-supplied. Displays a list of all task instances generated by this task definition.
Triggers tab	Displays a list of all triggers that have been defined to launch this task. Also allows you to add new triggers. If you add a new trigger from this location, the system automatically constructs a default trigger name as follows: <current task name>#TRIGGER#. You can change the default name if desired. For instructions on creating triggers, see Creating Triggers.
Notes tab	Displays all notes associated with this task. See Runbook Notes.

Versions tab	Stores copies of all previous versions of the current record (see Record Versioning).
Output tab	Task instance only. Displays output generated from the process, if any, based on specifications provided by the user in the Automatic Output Retrieval fields in the task definition.

Specifying When a Task Runs

You can run the task as part of a workflow, specify triggers that run the task automatically based on times or events, or run the task manually.

Monitoring Task Execution

You can monitor all system activity from the Activity display and can view activity history from the Activity History screen.

SAP Task

Overview

These instructions assume the user has a working knowledge of SAP.

The SAP task allows you to send commands to an SAP system and gather status information and output back from SAP. The SAP task uses Stonebranch's proprietary Universal Connector (USAP) to communicate with SAP. Universal Connector allows Opswise to connect to an SAP system and manage SAP background processing tasks.

Universal Connector Commands

The following table identifies supported Universal Connector commands, describes the actions that each command performs, and lists each command's related options.

Command Name	Description	Options
Run	Performs the following actions: 1. Defines a new SAP, job based on either a USAP Definition file or an SAP Model Job. 2. Starts the defined job. 3. Waits for the job to complete. 4. Prints the job's joblog to standard error and the spoollists to standard output. 5. Purges the job from the SAP system.	- **Definition or Model** Specifies how the new SAP job will be created, based either on a USAP Definition File or an SAP Model Job. - **Library or File System** Specifies whether the USAP definition file exists in the file system of the machine where the agent is running or in the Script Library. - **Script** Name of the script that has been uploaded into the Script Library and will be executed by this task. - **Definition File** If you selected USAP Definition File above, use this field to provide the path and file name of the file. - **SAP Job Name** Job name of the SAP job. Variables supported. - **SAP Job ID** Job ID of the SAP job. Variables supported. - **Target Job Name** If you selected SAP Model Job above, use this field to provide the name of the new SAP job being created. If you leave this field blank, the system uses the same name as the SAP Model Job.

		- **Start Immediately**
		Enabled or disabled. Enabling the Start Immediately flag will cause the job to fail if SAP resources are not available to start the job immediately (for example, a background work process). Otherwise, the job will wait for SAP resources to become available.
		- **SAP Target Server**
		Name of an SAP instance at which a background job should be run. The name has the following format:
		`[host name]_[SAP System name]_[SAP System number]`
		Where host name is the name of the server computer on which the instance is running, as specified in the system profile parameter SAPLOCALHOST.
		Example:
		`hs0123_C11_55`
		- **Print Application Log**
		Enabled or disabled. Specifies whether or not the job’s application log, if one was generated, is returned.
		- **Print Application RC**
		Enabled or disabled. Specifies whether or not the job’s application return codes, if they were set, are returned.
		- **Use Application RC**
		Specifies whether or not the SAP job's application return codes will be used to determine the return code for the Opswise task.
		- **SAP Command Options**
		Use this field to specify any additional command options supported by Universal Connector (USAP) .

Submit	Defines a new SAP job.	- **Definition or Model** Specifies how the new SAP job will be created, based either on a USAP Definition File or an SAP Model Job. - **Library or File System** Specifies whether the USAP definition file exists in the file system of the machine where the agent is running or in the Script Library. - **Script** Name of the script that has been uploaded into the Script Library and will be executed by this task. - **Definition File** If you selected USAP Definition File above, use this field to provide the path and file name of the file. - **SAP Job Name** Job name of the SAP job. Variables supported.

- **SAP Job ID**

  Job ID of the SAP job. Variables supported.

- **Target Job Name**

  If you selected SAP Model Job above, use this field to provide the name of the new SAP job being created. If you leave this field blank, the system uses the same name as the SAP Model Job.

- **Start**

  Enabled or disabled. Specifies whether or not the newly-defined SAP job should be started.

- **Start Immediately**

  Enabled or disabled. Enabling the Start Immediately flag will cause the job to fail if SAP resources are not available to start the job immediately (for example, a background work process). Otherwise, the job will wait for SAP resources to become available.

- **SAP Target Server**

  Name of an SAP instance at which a background job should be run. The name has the following format:

  ```
  [host name]_[SAP System name]_[SAP System
  number]
  ```

  Where host name is the name of the server computer on which the instance is running, as specified in the system profile parameter SAPLOCALHOST.

  Example:

  ```
  hs0123_C11_55
  ```

- **Wait**

  Specifies whether the system should wait for the SAP job to complete processing.

- **Print Spooled Output**

  Enabled or disabled. Specifies whether or not the spoollists of all job steps are returned.

- **Print Application Log**

  Enabled or disabled. Specifies whether or not the job’s application log, if one was generated, is returned.

- **Print Application RC**

  Enabled or disabled. Specifies whether or not the job’s application return codes, if they were set, are returned.

- **Use Application RC**

  Specifies whether or not the SAP job's application return codes will be used to determine the return code for the Opswise task.

- **SAP Command Options**

  Use this field to specify any additional command options supported by Universal Connector (USAP) .

Modify	Modifies an SAP job that already exists in an SAP system. A USAP job definition file is used to specify the modifications.	- **Library or File System** Specifies whether the USAP definition file exists in the file system of the machine where the agent is running or in the Script Library. - **Script** Name of the script that has been uploaded into the Script Library and will be executed by this task. - **Definition File** If you selected USAP Definition File above, use this field to provide the path and file name of the file. - **SAP Job ID** Job ID of the SAP job. Variables supported. - **SAP Command Options** Use this field to specify any additional command options supported by Universal Connector (USAP) .

Start	Starts a currently defined SAP job.	- **SAP Job Name**
		Job name of the SAP job. Variables supported.
		- **SAP Job ID**
		Job ID of the SAP job. Variables supported.
		- **Start Immediately**
		Enabled or disabled. Enabling the Start Immediately flag will cause the job to fail if SAP resources are not available to start the job immediately (for example, a background work process). Otherwise, the job will wait for SAP resources to become available.
		- **SAP Target Server**
		Name of an SAP instance at which a background job should be run. The name has the following format:
		`[host name]_[SAP System name]_[SAP System number]`
		Where host name is the name of the server computer on which the instance is running, as specified in the system profile parameter SAPLOCALHOST.
		Example:
		`hs0123_C11_55`
		- **Wait**
		Specifies whether the system should wait for the SAP job to complete processing.
		- **SAP Command Options**
		Use this field to specify any additional command options supported by Universal Connector (USAP) .

Wait	Reconnects to a started job and monitors it through completion.	• **SAP Job Name** Job name of the SAP job. Variables supported. • **SAP Job ID** Job ID of the SAP job. Variables supported. • **Print Job Log** Enabled or disabled. Specifies whether or not the job’s joblog is returned. • **Print Spooled Output** Enabled or disabled. Specifies whether or not the spoollists of all job steps are returned. • **Print Application Log** Enabled or disabled. Specifies whether or not the job’s application log, if one was generated, is returned. • **Print Application RC** Enabled or disabled. Specifies whether or not the job’s application return codes, if they were set, are returned. • **Use Application RC** Specifies whether or not the SAP job's application return codes will be used to determine the return code for the Opswise task. • **SAP Command Options** Use this field to specify any additional command options supported by Universal Connector (USAP) .
Abort	Cancels a running SAP job.	• **SAP Job Name** Job name of the SAP job. Variables supported. • **SAP Job ID** Job ID of the SAP job. Variables supported. • **SAP Command Options** Use this field to specify any additional command options supported by Universal Connector (USAP) .
Purge Job	Deletes a defined SAP job, its joblog, and all of its spoollists. This command is not available on SAP 3.1 and SAP 4.0.	• **SAP Job Name** Job name of the SAP job. Variables supported. • **SAP Job ID** Job ID of the SAP job. Variables supported. • **SAP Command Options** Use this field to specify any additional command options supported by Universal Connector (USAP) .

Purge Variant	Deletes a variant from an SAP system.	• **SAP ABAP Program Name** Name of the ABAP program for which the variant will be deleted. • **SAP Variant Name** Name of the variant to be deleted. • **SAP Command Options** Use this field to specify any additional command options supported by Universal Connector (USAP) .
Raise Event	Raises the specified SAP background processing event.	• **SAP Event** Name of the event. • **SAP Event Parameter** Optional parameter value for the event. • **SAP Command Options** Use this field to specify any additional command options supported by Universal Connector (USAP) .

Display	Displays the data specified in the Display Command field. The data is written to standard output.	• **Display Command** One of the following: • **Job Log** - Displays the job log for a specified SAP job. • **Spool List** - Displays the spoollist for a job step. • **Status** - Displays the current status for an SAP job. • **Variants** - Displays the variants available for the specified ABAP program • **Variant** - Displays the contents of a specified variant. **Note:** Requires XBP interface 2.0 or greater. • **Job Definition** - Displays the definition of the specified SAP job. • **Select** - Displays a variety of attributes for a list of SAP jobs that match the specified criteria. • **System Log** - Displays a portion of an SAP syslog that meets the specified date/time constraints. • **Intercept Table** - Displays the contents of the job intercept criteria table for the connected SAP system. • **Intercepted Jobs** - Displays intercepted jobs for the connected SAP system. • **Reports** - Displays a list of ABAP reports that match the specified criteria. • **Commands** - Displays a list of SAP external commands that match the specified criteria. • **Output Devices** - Displays a list of SAP output devices that match the specified criteria. • **Print Formats** - Displays a list of print formats that are available for the specified printer. • **Selection Screen** - Displays information about the selection fields of an ABAP program. • **Event History** - Displays a list of events that were logged in an SAP system's event history. The retrieved events can optionally be set to "Confirmed". • **Criteria Manager Profiles** - Displays a list of Criteria Manager profiles. • **Criteria Manager Criteria** - Displays the criteria hierarchy of a particular profile in XML format. • **SAP Job Name** Job name of the SAP job. Variables supported. • **SAP Job ID** Job ID of the SAP job. Variables supported. • **SAP Command Options** - Use this field to specify any additional command options supported by Universal Connector (USAP) .
Generate Variant Definition	Generates a USAP variant definition file based on a model SAP variant. The generated definition file is written to standard output. Requires XBP interface 2.0 or greater.	• **SAP ABAP Program Name** Name of an ABAP program in an SAP system to which the model variant belongs. • **SAP Variant Name** Pre-existing SAP variant name to use as the model variant. • **SAP Command Options** Use this field to specify any additional command options supported by Universal Connector (USAP) .

Generate Job Definition	Generates a USAP job definition file based on a model SAP job. The generated definition file is written to standard output.	• **SAP Job Name** Job name of the SAP job. Variables supported. • **SAP Job ID** Job ID of the SAP job. Variables supported. • **SAP Command Options** Use this field to specify any additional command options supported by Universal Connector (USAP) .
Create CM Profile	Creates a new Criteria Manager profile.	• **SAP Criteria Manager XML File** Name of the file that contains the Criteria Manager information. • **SAP Command Options** Use this field to specify any additional command options supported by Universal Connector (USAP) .
Set CM Criteria	Sets the criteria for a profile.	• **SAP Criteria Manager XML File** Name of the file that contains the Criteria Manager information. • **SAP Criteria Manager Profile ID** ID of the profile. • **SAP Criteria Manager Profile Type** Type of profile. For the default criteria types provided by SAP, the values are: • **EVTHIS** - Identifies a criteria type for event history. • **EVHIRO** - Identifies a criteria type for the reorganization of raised events. • **INTERC** - Identifies a criteria type for job interception. • **SAP Command Options** Use this field to specify any additional command options supported by Universal Connector (USAP) .
Activate CM Profile	Activates a criteria profile of the specified type.	• **SAP Criteria Manager Profile ID** ID of the profile. • **SAP Criteria Manager Profile Type** Type of profile. For the default criteria types provided by SAP, the values are: • **EVTHIS** - Identifies a criteria type for event history. • **EVHIRO** - Identifies a criteria type for the reorganization of raised events. • **INTERC** - Identifies a criteria type for job interception. • **SAP Command Options** Use this field to specify any additional command options supported by Universal Connector (USAP) .

Deactivate CM Profile	Deactivates a criteria profile of the specified type.	• **SAP Criteria Manager Profile Type** Type of profile. For the default criteria types provided by SAP, the values are: ◦ **EVTHIS** - Identifies a criteria type for event history. ◦ **EVHIRO** - Identifies a criteria type for the reorganization of raised events. ◦ **INTERC** - Identifies a criteria type for job interception. • **SAP Command Options** Use this field to specify any additional command options supported by Universal Connector (USAP) .
Delete CM Profile	Deletes a criteria profile from an SAP system.	• **SAP Criteria Manager Profile ID** ID of the profile. • **SAP Criteria Manager Profile Type** Type of profile. For the default criteria types provided by SAP, the values are: ◦ **EVTHIS** - Identifies a criteria type for event history. ◦ **EVHIRO** - Identifies a criteria type for the reorganization of raised events. ◦ **INTERC** - Identifies a criteria type for job interception. • **SAP Command Options** Use this field to specify any additional command options supported by Universal Connector (USAP) .

Built-In Variables

The built-in variables outlined below can be used in an SAP task to pass data where appropriate:

- Task and Task Instance Variables
- Script Variables
- SAP Task Variables
- Agent Variables

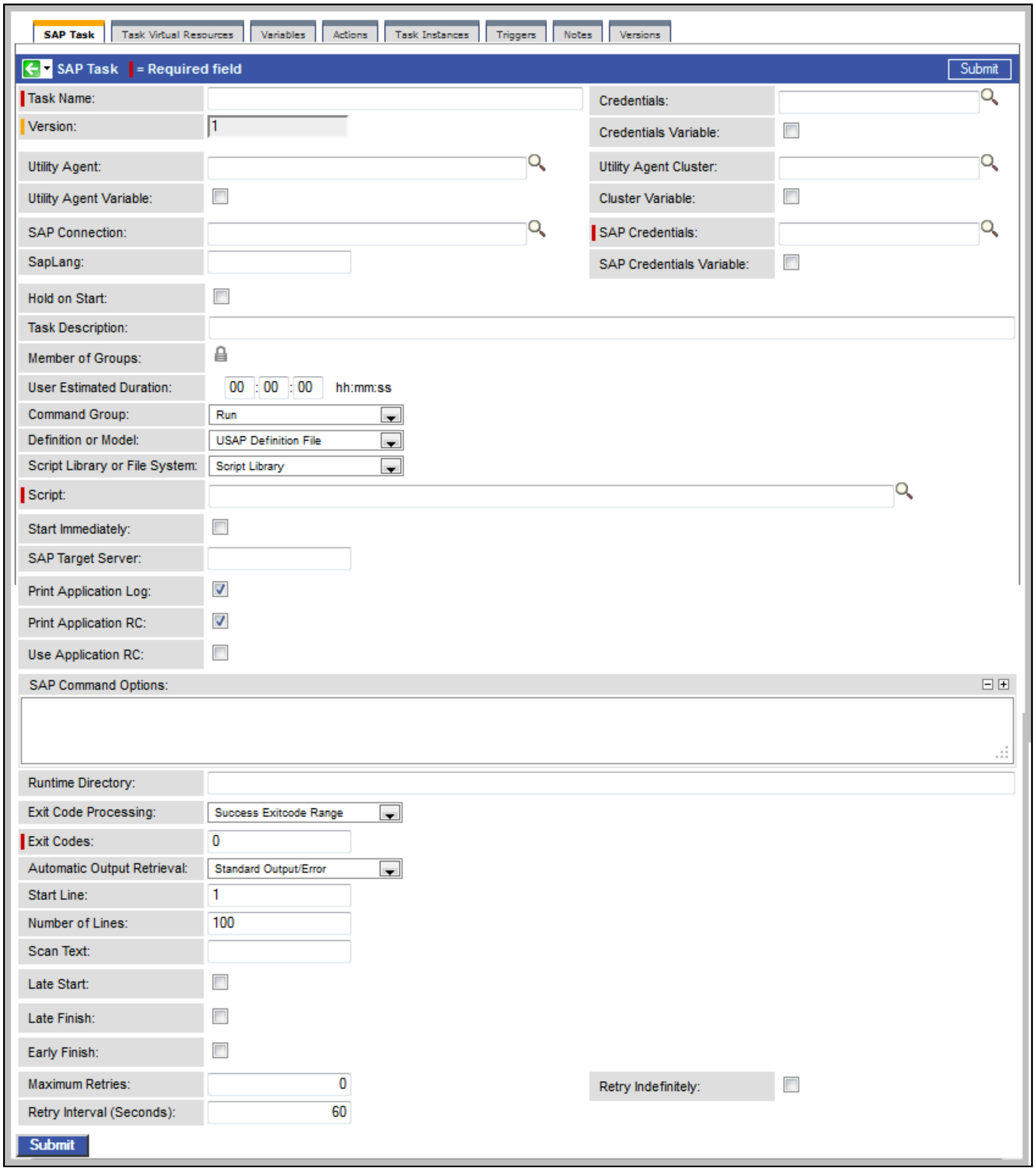
Before You Begin

To run an SAP task, you must first complete the following:

- Identify an Opswise Linux/Unix agent that will interface with the SAP system.
- Define an SAP connection in the Opswise database.

Creating a New SAP Task

1. From the navigation pane, select **Automation Center > Tasks > SAP Tasks**. The SAP Tasks List displays.
2. Click **New**. The SAP Task Definition screen displays.



SAP Task = Required field

Task Name:		Credentials:	
Version:	1	Credentials Variable:	□
Utility Agent:		Utility Agent Cluster:	
Utility Agent Variable:	□	Cluster Variable:	□
SAP Connection:		SAP Credentials:	
SapLang:		SAP Credentials Variable:	□
Hold on Start:	□		
Task Description:			
Member of Groups:			
User Estimated Duration:	00 : 00 : 00 hh:mm:ss		
Command Group:	Run		
Definition or Model:	USAP Definition File		
Script Library or File System:	Script Library		
Script:			
Start Immediately:	□		
SAP Target Server:			
Print Application Log:	☑		
Print Application RC:	☑		
Use Application RC:	□		
SAP Command Options:			
Runtime Directory:			
Exit Code Processing:	Success Exitcode Range		
Exit Codes:	0		
Automatic Output Retrieval:	Standard Output/Error		
Start Line:	1		
Number of Lines:	100		
Scan Text:			
Late Start:	□		
Late Finish:	□		
Early Finish:	□		
Maximum Retries:	0	Retry Indefinitely:	□
Retry Interval (Seconds):	60		

3. Using the field descriptions provided below as a guide, complete the fields as needed.
4. Click the **Submit** button to save the record and return to the menu, or right-click on the title bar and select **Save** to save the record and remain on the current display.
5. If appropriate, repeat these steps for any additional tasks you want to add.

Task run statistics appear after the first time this task has been launched.

SAP Task Field Descriptions

The table below describes the fields, buttons, and tabs on the task definition and task instance screens. Color coding is provided that differentiates the following three types of fields:

- Fields that display on the task definition and task instance screens are shown in black.
- Fields that display only on the task definition screen are shown in green.
- Fields that display only on the task instance screen are shown in maroon.

Field Name	Description

Task/Instance Name	Required. Name used within Opswise to identify this task. Up to 40 alphanumerics; variables supported. It is the user's responsibility to develop a workable naming scheme for tasks.
Invoked by	Task instance only; system-supplied. How the task instance was launched. One of the following: • Trigger: (Trigger Name) - The instance was launched by the named trigger. • Workflow: (Workflow Name) - The instance was launched by the named workflow. • Manually Launched - The instance was launched by a user. To determine the name of the user: 1. From the Activity or All Task Instances screen, click the task instance name to open the record. 2. The Execution User field identifies the user who launched the task instance.
Task	Task instance only; system-supplied. Hover over the paper icon to display more information about the task instance. Click the paper icon to display the task definition record.
Execution User	Task instance only; system-supplied. If the task was launched manually, the ID of the user who launched it.
Instance Reference Id	Task instance only; system-supplied. The system increments this number each time the task is run.
Credentials	Login credentials Opswise Automation Center will use to access the remote machine. They must be provided either in the resource definition or task definition, or both. Credentials provided in the task definition override any credentials provided in the agent resource definition. See Credentials.
Version	Task definition only; system-supplied. The version number of the current record, which is incremented by the system every time a user updates a record. Click on the Versions tab to view previous versions. For details, see Record Versioning.
Credentials Variable	Optional. If enabled, the Credentials field (see above) converts from a reference field (where you browse and select a record) into a text field that allows you to enter a variable. Use the format: ${variable name}. The variable must be a supported type as described in Using Variables and Functions .
Utility Agent	Required. The name of the Linux/Unix agent resource that will communicate with the SAP system. If you do not specify an agent, you must specify an agent cluster (see below).
Utility Agent Variable	Optional. If enabled, the Utility Agent field (see above) converts from a reference field (where you browse and select a record) into a text field that allows you to enter a variable. Use the format: ${variable name}. The variable must be a supported type as described in Using Variables and Functions.
Utility Agent Cluster	Optional. You can specify an agent cluster in addition to or in place of a specific agent. An agent cluster is a group of agents, one of which the system will choose to run this task. If you specify an agent and an agent cluster, Opswise Automation Center first tries to run the task on the specific agent. If the agent is not available, the system reverts to the agent cluster. See Configuring Agent Clusters for more information
Cluster Variable	Optional. If enabled, the Agent Cluster field (see above) converts from a reference field (where you browse and select a record) into a text field that allows you to enter a variable. Use the format: ${variable name}. The variable must be a supported type as described in Using Variables and Functions.
SAP Connection	Required. The name of the SAP connection defined using the SAP connection screen. The SAP connection specifies information about the SAP server. Type in a name, or click the magnifying glass to browse to an existing SAP server definition or create a new one.

SAP Credentials	Login credentials Opswise will use to access the SAP system. The credentials are stored in the Opswise credentials table; see Credentials.
SAPLang	SAP logon language used when executing the SAP task. Valid values are: • Any valid 1-character SAP language identifier • Any valid 2-character ISO language identifier • (no value). SAP will use the default language set up for the user. If there is no such default, the default is EN (English).
SAP Credentials Variable	Optional. If enabled, the Credentials field (see above) converts from a reference field (where you browse and select a record) into a text field that allows you to enter a variable. Use the format: ${variable name}. The variable must be a supported type as described in Using Variables and Functions .
Hold on Start	If enabled, when the task is launched it appears in the Activity display with a status of Held. The task runs when the user releases it.
Hold Reason	Information about why the task will be put on hold when it starts.
Task Description	User-supplied description of this record.
Status	Task instance only; system-supplied. See Task Statuses.
Exit Code	Task instance only; system-supplied. The exit code captured by the agent when executing the task (for example, a command or script).
Status Description	Task instance only; system-supplied. Provides additional information, if any, about the status of the task.
Member of Groups	User-defined. Allows you to select one or more Opswise groups that this record definition belongs to. Click on the lock icon to unlock the field and select groups.
User Estimated Duration	Task definition only; optional. The estimated amount of time it should normally take to run this task. The system uses this information to calculate the User Estimated End Time on a task instance record.
Command Group	See Universal Connector Commands, above, for a description of all supported commands and their contingent fields.
Runtime Directory	Optional. The directory from which the application should be executed. Variables supported.

Exit Code Processing	Required. Specifies how Opswise Automation Center should determine whether the executed command failed or completed successfully. Options: • **Success Exitcode Range** - The command is considered completed successfully if its exit code falls within the range specified in the Exit Codes field (see below). • **Failure Exitcode Range** - The command is considered failed if its exit code falls within the range specified in the Exit Codes field (see below). • **Success Output Contains** - The command is considered completed successfully if its output contains the text specified in the Scan Output For field (see below). • **Failure Output Contains** - The command is considered failed if its output contains the text specified in the Scan Output For field (see below). • **Step Conditions** - The command is considered completed successfully/failed if any of its specified condition codes falls within the range specified under the Step Conditions tab (see Creating Step Conditions).
Output Type-Exit Code	Required if Exit Code Processing = Success Output Contains or Failure Output Contains. This field specifies the type of output. Options: • Standard Output (STDOUT) • Standard Error (STDERR) • File
Exit Codes	Required if Exit Code Processing = Success Exitcode Range or Failure Exit Code Range. This field specifies the range. Format: Numeric. Use commas to list a series of exitcodes; use hyphens to specify a range. Example: 1,5, 22-30.
Scan Output For	Required if Exit Code Processing = Success Output Contains or Failure Output Contains. This field specifies the text for which Opswise should scan the output file. Opswise will process this field as a regular expression.
Output File-Exit Codes	Required if Output Type = File. This field specifies the path and file name of the output file that should be scanned for the text in the Scan Output For field.
Environment Variables-Name and Value	Optional. Allows you to enter environment variables needed by the program to run. Type in the name of the variable and its value, and click **Add**. The variable is listed in the space underneath. You can add as many variables as needed. To delete a variable, click the **X** button.
Automatic Output Retrieval	Optional. Allows you to specify whether you want the system to automatically retrieve any output from the job and attach it to the task instance record. Options: • **None** - Do not attach any output to the task instance record. • **Standard Output** - Attach all standard output. • **Standard Error** - Attach standard error output. • **File** - Attach the file specified in the Output File field.
Output File-Auto Retrieve	Required if Automatic Output Retrieval=File. This field specifies the path and filename containing the output that you want automatically retrieved and attached to the task instance.
Start Line	Optional. Allows you to instruct the system to fetch data beginning at the line indicated.
Number of Lines	Optional. Allows you to limit the retrieved data to the number of lines specified. The default is 100. You can change the default by modifying the record **Configuration > Properties > Retrieve Output Default Maximum Lines**.
Scan Text-Automatic Output Retrieval	Optional. Instructs the system to scan the data for the text specified and retrieve only that. Opswise will process this field as a regular expression.

Late Start	If enabled, and if the task instance starts after the time specified in the Late Start Time field (see below), the task instance is flagged as late. To determine whether a task instance started late, open the task instance and locate the Started Late field; the field is checked if the instance started late.
Late Start Type	Required if Late Start is enabled. Options are: • **Time** - Flag the task if it starts after the specified time. • **Duration** - Flag the task if it starts a certain amount of time after the programmed start time. The task must have a specific start time.
Late Start Time	Time after which the task start time is considered late. Use hh:mm, 24-hour time
Late Start Duration	Duration (amount of relative time) after which the task is considered to have started late. For a task within a workflow, the duration is the period between the time the workflow starts and the time the task itself starts. For example, a task might have a Late Start Duration of 60 minutes. If the workflow starts at 9:00 a.m. but the task itself does not start until 10:30, the task has started late. For a task that is not within a workflow, Late Start Duration has meaning only if the task has been held upon starting. For example, if a task has a Late Start Duration of 60 minutes and the Hold on Start field is enabled, if the task is not released from hold within the amount of time specified in the Late Start Duration field, the task has started late.
Started Late	Task instance only; system-supplied. This field is flagged if the task started later than the time specified in the Late Start fields.
Late Finish	If enabled, and if the task instance finishes after the time or period specified, the task instance is flagged as late. You can specify a time or duration to determine a late finish (see Late Finish Type below). To determine whether a task instance finished late, open the task instance and locate the Finished Late field; the field is checked if the instance finished after the specified time or lasted longer than expected. This field only appears on the task instance if the user specified a Late Finish in the task definition.
Late Finish Type	If Late Finish is enabled, use this field to specify whether you want to use time (see Late Finish Time below) or duration (see Late Finish Duration below) to determine whether the task instance finished late.
Late Finish Time	If Late Finish Type is Time, use this to specify the time after which the task finish time is considered late. Use hh:mm, 24-hour time.
Late Finish Duration	If Late Finish Type is Duration, use this to specify the longest amount of time this task instance should take to run. You can specify any combination of hours, minutes, and seconds.
Finished Late	Task instance only; system-supplied. This field is flagged if the task finished later than the time or duration specified in the Late Finish fields.
Early Finish	If enabled, and if the task instance finishes before the time or period specified, the task instance is flagged as early. You can specify a time or duration to determine an early finish (see Early Finish Type below). To determine whether a task instance finished early, open the task instance and locate the Finished Early field; the field is checked if the instance finished before the specified time or did not last as long as expected. This field only appears on the task instance if the user added Early Finish specifications to the task definition.
Early Finish Type	If Early Finish is enabled, use this field to specify whether you want to use time (see Early Finish Time below) or duration (see Early Finish Duration below) to determine whether the task instance finished early.

Early Finish Time	If Early Finish Type is Time, use this to specify the time before which the task finish time is considered early. That is, enter a time at which the task should still be running. Use hh:mm, 24-hour time.
Early Finish Duration	If Early Finish Type is Duration, use this to specify the shortest amount of time this task instance should take to run. You can specify any combination of hours, minutes, and seconds.
Finished Early	Task instance only; system-supplied. This field is flagged if the task finished earlier than the time specified in the Early Finish fields.
Maximum Retries	User-defined. The maximum number of times the system should retry this task after it has started and gone to a failed state.
Retry Indefinitely	User-defined. Enabled or disabled. Indicates whether the system should continue trying indefinitely to run this task. If you enable this field, it overrides any value placed in the Maximum Retries field (above).
Retry Interval	User-defined. The number of seconds between each retry.
Current Retry Count	Task instance only; system-supplied. Displays only for a running task instance. Displays the current number of times the system has retried the task after it first went to failure status.
First Time Ran	Task definition only; system-supplied. Displays after the first time the task runs. The date and time this task first ran.
Lowest Instance Time	Task definition only; system-supplied. Displays after the first time the task runs. The shortest amount of time this task has taken to run.
Last Time Ran	Task definition only; system-supplied. Displays after the first time the task runs. The most recent date and time the task ran.
Average Instance Time	Task definition only; system-supplied. Displays after the first time the task runs. Shows the average amount of time this task takes to run.
Number of Instances	Task definition only; system-supplied. Displays after the first time the task runs. Shows the number of times this task has run.
Highest Instance Time	Task definition only; system-supplied. Displays after the first time the task runs. The longest amount of time this task has taken to run.
Last Instance Duration	Task definition only; system-supplied. Displays after the first time the task runs. The amount of time the task took to run the last time it ran.
Submit button	Submits the new record to the database.
Update button	Saves updates to the record.
Launch Task button	Manually launches the task.

Delete button	Deletes the current record.
View Parent button	Task instance only; Displays this task's parent task (workflow), if any.
Show Details button	Task instance only; displays detailed information about this task instance.
Skip button	Task instance only; for tasks loaded into the schedule that haven't run yet. Allows you to tell the system to skip this task. See Skipping a Task.
Hold button	Task instance only; see Putting a Task on Hold.
Force Finish button	Task instance only; see Force Finishing a Task.
Re-run button	Task instance only; see Re-running a Task.
Output tab	Task instance only. Displays output generated from the process, if any, based on specifications provided by the user in the Automatic Output Retrieval fields in the task definition.
Task Virtual Resources tab	Lists Virtual Resources to which this task is assigned. See Creating Virtual Resources.
Variables tab	Displays all variables associated with this record.
Actions tab	Allows you to specify actions that the system will take automatically based on events that occur during the execution of this task. Events are task instance status, exit codes, late start, late finish, and early finish. Actions are: • **Abort Action** - Abort the task if certain events occur. For details, see Setting Up Abort Actions. • **Email Notification** - Send an email if certain events occur. For details, see Creating Email Notifications. • **SNMP Notification** - Send an email if certain events occur. For details, see Creating SNMP Notifications. • **Set Variable** - Used in workflows to set a variable based on the occurrence of certain events. For details, see Setting Variables in a Workflow.
Task Instances tab	System-supplied. Displays a list of all task instances generated by this task definition.
Triggers tab	Displays a list of all triggers that have been defined to launch this task. Also allows you to add new triggers. If you add a new trigger from this location, the system automatically constructs a default trigger name as follows: <current task name>#TRIGGER#. You can change the default name if desired. For instructions on creating triggers, see Creating Triggers.
Notes tab	Displays all notes associated with this task. See Runbook Notes.
Versions Tab	Stores copies of all previous versions of the current record (see Record Versioning).

Specifying When a Task Runs

You can run the task as part of a workflow, specify triggers that run the task automatically based on times or events, or run the task manually.

Monitoring Task Execution

You can monitor all system activity from the Activity display and can view activity history from the Activity History screen.

File Transfer Task

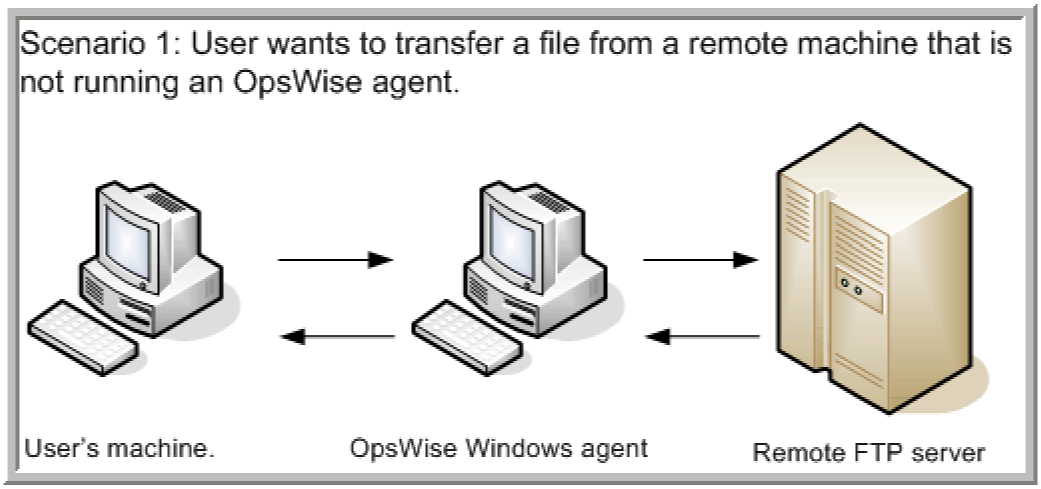
Overview

The File Transfer task allows you to execute file transfers on remote machines using any of the following protocols:

- FTP
- SFTP
- INFITRAN

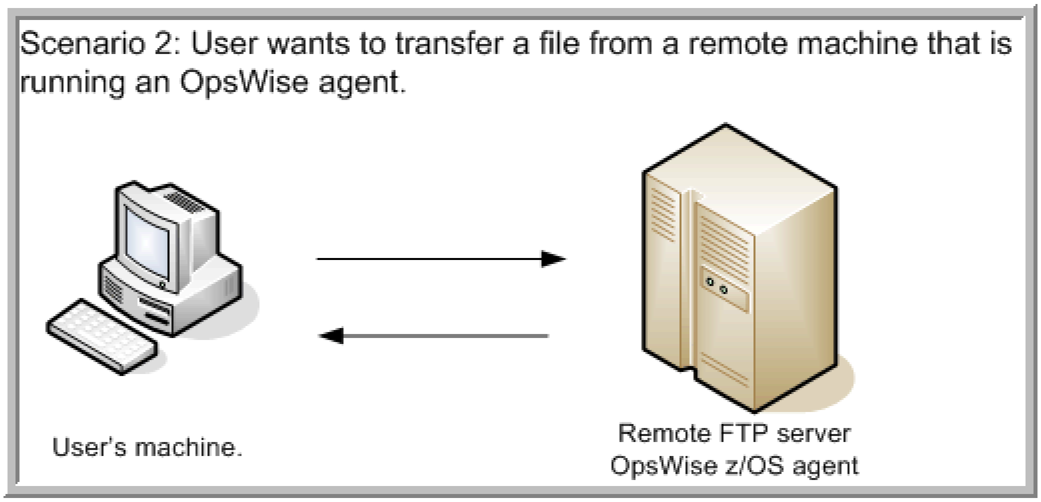
To run a File Transfer task, you need an Opswise Linux/Unix, z/OS, or Windows agent to communicate with the File Transfer server. The agent can but does not have to be running on the same machine as the File Transfer server. The following diagrams provide sample configurations for executing file transfers using an Opswise File Transfer task.

In the example shown below, the user wants to transfer a file from a remote File Transfer Server on a machine that does not have an Opswise agent running on it. In this case, the File Transfer task definition provides an address and login credentials for the machine where the Opswise agent is running as well as address and login credentials for the machine where the File Transfer server is running.



Scenario 1: User wants to transfer a file from a remote machine that is not running an OpsWise agent.
User's machine.
OpsWise Windows agent
Remote FTP server

In the example shown below, the user wants to transfer a file from a remote File Transfer Server on a z/OS machine that does have an Opswise agent running on it. In this case, the login credentials for the Opswise agent machine and the File Transfer server machine are the same.


Scenario 2: User wants to transfer a file from a remote machine that is running an OpsWise agent.
User's machine.
Remote FTP server
OpsWise z/OS agent

FTP and SFTP File Transfer Tasks

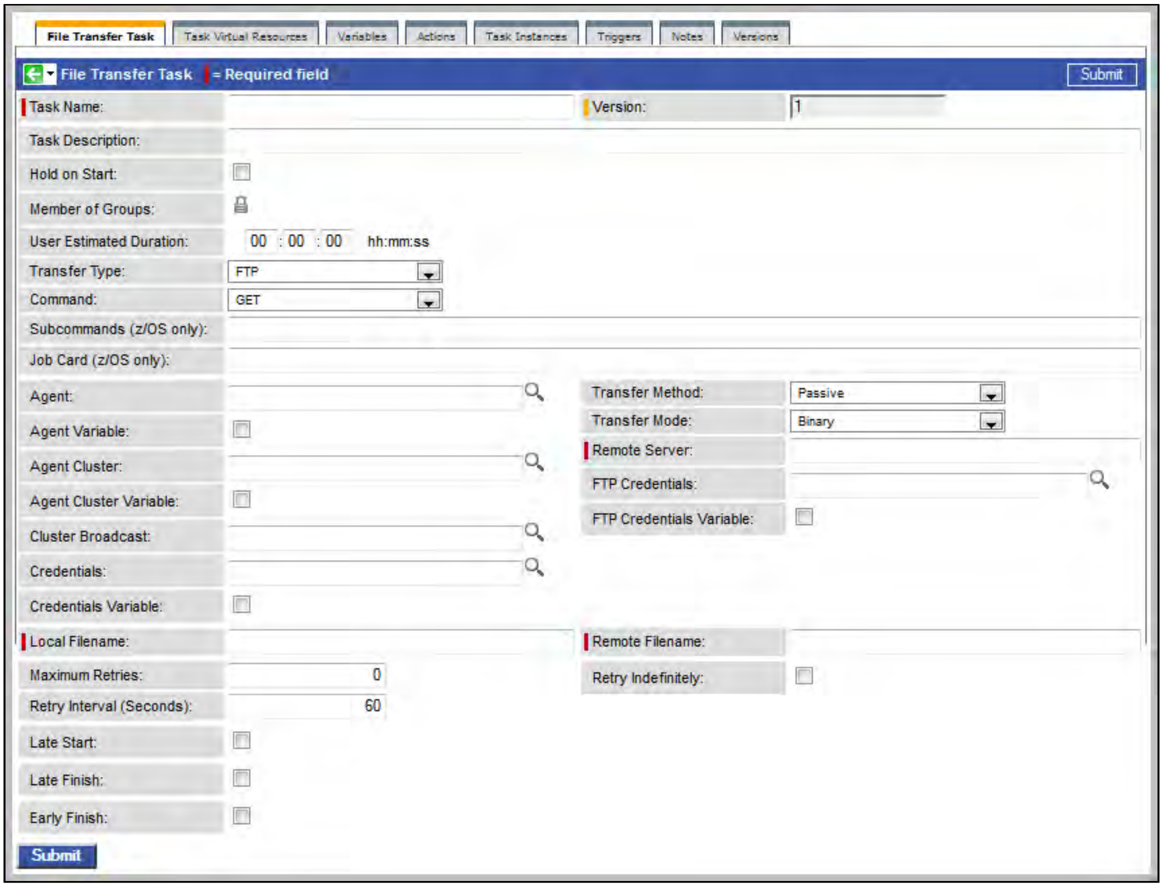
The screens for FTP and SFTP are the same; for information about INFITRAN File Transfer, click here.

Using **SFTP** requires that you supply a valid credential that specifies the location of the SSL Private key on your agent. In the Opswise Credentials form for the user you will use, you supply the location for the private key in the field "Key Location (File Transfer only)". This location must exist on the agent where you intend to run the SFTP task. Automation Center does not at this time support password authentication for SFTP Transfer.

Make sure you have your private/public keys properly set up and working before you configure Opswise to use it. For example, to validate the keys, log into your destination server from your agent server using ssh.

Creating an FTP or SFTP File Transfer Task

1. From the navigation pane, select **Automation Center > Tasks > File Transfer Tasks**. The File Transfer Tasks List screen displays.
2. Click **New**. The File Transfer Task Definition screen displays. (This example shows the FTP file transfer method.)



Task Name:		Version:	1
Task Description:			
Hold on Start:	□		
Member of Groups:			
User Estimated Duration:	00 : 00 : 00 hh:mm:ss		
Transfer Type:	FTP		
Command:	GET		
Subcommands (z/OS only):			
Job Card (z/OS only):			
Agent:		Transfer Method:	Passive
Agent Variable:	□	Transfer Mode:	Binary
Agent Cluster:		Remote Server:	
Agent Cluster Variable:	□	FTP Credentials:	
Cluster Broadcast:		FTP Credentials Variable:	□
Credentials:			
Credentials Variable:	□		
Local Filename:		Remote Filename:	
Maximum Retries:	0	Retry Indefinitely:	□
Retry Interval (Seconds):	60		
Late Start:	□		
Late Finish:	□		
Early Finish:	□		

3. In the Transfer Type field, select FTP or SFTP. Using the field descriptions provided below as a guide, complete the fields as needed.
4. Click the **Submit** button to save the record and return to the menu, or right-click on the title bar and select **Save** to save the record and remain on the current display.
5. If appropriate, repeat these steps for any additional tasks you want to add.

Task run statistics appear after the first time this task has been launched.

FTP and SFTP File Transfer Task Field Descriptions

The table below describes the fields, buttons, and tabs on the task definition and task instance screens. Color coding is provided that differentiates the following three types of fields:

- Fields that display on the task definition and task instance screens are shown in black.
- Fields that display only on the task definition screen are shown in green.
- Fields that display only on the task instance screen are shown in maroon.

Field Name	Description
Task/Instance Name	Required. Name used within Opswise to identify this task. Up to 40 alphanumerics; variables supported. It is the user's responsibility to develop a workable naming scheme for tasks.
Version	Task definition only; system-supplied. The version number of the current record, which is incremented by the system every time a user updates a record. Click on the Versions tab to view previous versions. For details, see Record Versioning.

Invoked by	Task instance only; system-supplied. How the task instance was launched. One of the following: - Trigger: (Trigger Name) - The instance was launched by the named trigger. - Workflow: (Workflow Name) - The instance was launched by the named workflow. - Manually Launched - The instance was launched by a user. To determine the name of the user: 1. From the Activity or All Task Instances screen, click the task instance name to open the record. 2. The Execution User field identifies the user who launched the task instance.
Task	Task instance only; system-supplied. Hover over the paper icon to display more information about the task instance. Click the paper icon to display the task definition record.
Execution User	Task instance only; system-supplied. If the task was launched manually, the ID of the user who launched it.
Instance Reference Id	Task instance only; system-supplied. The system increments this number each time the task is run.
z/OS ID	Task instance only; z/OS only. The z/OS execID, used internally by the z/OS agent to identify each z/OS task.
Task Description	User-supplied description of this record.
Hold on Start	If enabled, when the task is launched it appears in the Activity display with a status of Held. The task runs when the user releases it.
Hold Reason	Information about why the task will be put on hold when it starts.
Member of Groups	User-defined. Allows you to select one or more Opswise groups that this record definition belongs to. Click on the lock icon to unlock the field and select groups.
User Estimated Duration	Task definition only; optional. The estimated amount of time it should normally take to run this task. The system uses this information to calculate the User Estimated End Time on a task instance record.
Status	Task instance only; system-supplied. See Task Statuses.
Exit Code	Task instance only; system-supplied. The exit code captured by the agent when executing the task (for example, a command or script).
Status Description	Task instance only; system-supplied. Provides additional information, if any, about the status of the task.
Start Time	Task instance only; system-supplied. The date and time the task started.
Duration	Task instance only; system-supplied. The amount of time the task took to run.
End Time	Task instance only; system-supplied. The date and time the task instance completed.

Transfer Type	Type of File Transfer server. Options: - FTP - SFTP - INFITRAN
Command-FTP/SFTP	File Transfer command being executed. Options: - **GET** - Copies a remote file to the local computer. - **PUT** - Copies a local file to the remote computer. - **MGET** - Copies multiple remote files to the local computer. - **MPUT** - Copies multiple local files to the remote computer. - **DELETE** - Deletes the specified file from the remote computer. - **MDELETE** - Deletes the specified file(s) from the remote computer - **MKDIR** - Creates the specified directory on the remote computer. - **RMDIR** - Removes the specified directory from the remote computer.
Subcommands (z/OS only)	For z/OS, any subcommands used in the JCL statement.

Job Card (z/OS only)	For z/OS, the job card information for the JCL statement. Example:
	`//File TransferJOB01 JOB (File`
	`Transfer,001),FANNY,MSGCLASS=X,MSGLEVEL=(1,1),NOTIFY=&SYSUID,CLASS=A`

Agent	Optional. The name of the agent resource definition that identifies the machine where the operation will run. If you do not specify an agent, you must specify an agent cluster (see below).
Transfer Method	Transfer method. Options: - Active - Passive - Extended Passive
Agent Variable	Optional. If enabled, the Agent field (see above) converts from a reference field (where you browse and select a record) into a text field that allows you to enter a variable. Use the format: ${variable name}. The variable must be a supported type as described in Using Variables and Functions.
Transfer Mode	Mode of data transfer. Options: - Binary - ASCII
Agent Cluster	Optional. You can specify an agent cluster in addition to or in place of a specific agent. An agent cluster is a group of agents, one of which the system will choose to run this task. If you specify an agent and an agent cluster, Opswise Automation Center first tries to run the task on the specific agent. If the agent is not available, the system reverts to the agent cluster. See Configuring Agent Clusters for more information.
Remote Server	Name or IP address of the File Transfer server. This machine may or may not be the same as the Opswise agent machine.

Agent Cluster Variable	Optional. If enabled, the Agent Cluster field (see above) converts from a reference field (where you browse and select a record) into a text field that allows you to enter a variable. Use the format: ${variable name}. The variable must be a supported type as described in Using Variables and Functions.
FTP Credentials	Login credentials the Opswise agent will use to access the FTP or SFTP server machine. If the File Transfer server and Opswise agent are running on the same machine, enter the same credentials as those you entered in the Credentials field.
Cluster Broadcast	Task definition only; optional. You can specify a Cluster Broadcast in addition to or in place of a specific agent and/or Agent Cluster. When you specify an agent cluster in the Cluster Broadcast field, Opswise Automation Center runs the task on all the agents in the cluster. Each instance of the task running on its own agent becomes a separate task instance record in the database and displays separately in the Activity monitor. See Configuring Agent Clusters for more information about defining agent clusters.
FTP Credentials Variable	Optional. If enabled, the FTP Credentials field (see above) converts from a reference field (where you browse and select a record) into a text field that allows you to enter a variable. Use the format: ${variable name}. The variable must be a supported type as described in Using Variables and Functions.
Credentials	Login credentials Opswise Automation Center will use to access the remote machine. They must be provided either in the resource definition or task definition, or both. Credentials provided in the task definition override any credentials provided in the agent resource definition. See Credentials.
Credentials Variable	Optional. If enabled, the Credentials field (see above) converts from a reference field (where you browse and select a record) into a text field that allows you to enter a variable. Use the format: ${variable name}. The variable must be a supported type as described in Using Variables and Functions .
Local Filename	The path and file name on the local server. That is, the "transfer from" filename.
Remote Filename	Path and file name on the remote server. That is, the "transfer to" filename.
Maximum Retries	User-defined. The maximum number of times the system should retry this task after it has started and gone to a failed state.
Retry Indefinitely	User-defined. Enabled or disabled. Indicates whether the system should continue trying indefinitely to run this task. If you enable this field, it overrides any value placed in the Maximum Retries field (above).
Retry Interval	User-defined. The number of seconds between each retry.
Current Retry Count	Task instance only; system-supplied. Displays only for a running task instance. Displays the current number of times the system has retried the task after it first went to failure status.
User Estimated End Time	Task instance only; system-supplied. If the user entered information into the User Estimated Duration field in the task definition, the system uses this information to calculate an end time for the task instance, based on the date/time the task instance started.
Shortest Estimated End Time	Task instance only; system-supplied.

Average Estimated End Time	Task instance only; system-supplied.
Longest Estimated End Time	Task instance only; system-supplied.
Late Start	If enabled, and if the task instance starts after the time specified in the Late Start Time field (see below), the task instance is flagged as late. To determine whether a task instance started late, open the task instance and locate the Started Late field; the field is checked if the instance started late.
Late Start Type	Required if Late Start is enabled. Options are: - **Time** - Flag the task if it starts after the specified time. - **Duration** - Flag the task if it starts a certain amount of time after the programmed start time. The task must have a specific start time.
Late Start Time	Time after which the task start time is considered late. Use hh:mm, 24-hour time
Late Start Duration	Duration (amount of relative time) after which the task is considered to have started late. For a task within a workflow, the duration is the period between the time the workflow starts and the time the task itself starts. For example, a task might have a Late Start Duration of 60 minutes. If the workflow starts at 9:00 a.m. but the task itself does not start until 10:30, the task has started late. For a task that is not within a workflow, Late Start Duration has meaning only if the task has been held upon starting. For example, if a task has a Late Start Duration of 60 minutes and the Hold on Start field is enabled, if the task is not released from hold within the amount of time specified in the Late Start Duration field, the task has started late.
Started Late	Task instance only; system-supplied. This field is flagged if the task started later than the time specified in the Late Start fields.
Late Finish	If enabled, and if the task instance finishes after the time or period specified, the task instance is flagged as late. You can specify a time or duration to determine a late finish (see Late Finish Type below). To determine whether a task instance finished late, open the task instance and locate the Finished Late field; the field is checked if the instance finished after the specified time or lasted longer than expected. This field only appears on the task instance if the user specified a Late Finish in the task definition.
Late Finish Type	If Late Finish is enabled, use this field to specify whether you want to use time (see Late Finish Time below) or duration (see Late Finish Duration below) to determine whether the task instance finished late.
Late Finish Time	If Late Finish Type is Time, use this to specify the time after which the task finish time is considered late. Use hh:mm, 24-hour time.
Late Finish Duration	If Late Finish Type is Duration, use this to specify the longest amount of time this task instance should take to run. You can specify any combination of hours, minutes, and seconds.
Finished Late	Task instance only; system-supplied. This field is flagged if the task finished later than the time or duration specified in the Late Finish fields.

Early Finish	If enabled, and if the task instance finishes before the time or period specified, the task instance is flagged as early. You can specify a time or duration to determine an early finish (see Early Finish Type below). To determine whether a task instance finished early, open the task instance and locate the Finished Early field; the field is checked if the instance finished before the specified time or did not last as long as expected. This field only appears on the task instance if the user added Early Finish specifications to the task definition.
Early Finish Type	If Early Finish is enabled, use this field to specify whether you want to use time (see Early Finish Time below) or duration (see Early Finish Duration below) to determine whether the task instance finished early.
Early Finish Time	If Early Finish Type is Time, use this to specify the time before which the task finish time is considered early. That is, enter a time at which the task should still be running. Use hh:mm, 24-hour time.
Early Finish Duration	If Early Finish Type is Duration, use this to specify the shortest amount of time this task instance should take to run. You can specify any combination of hours, minutes, and seconds.
Finished Early	Task instance only; system-supplied. This field is flagged if the task finished earlier than the time specified in the Early Finish fields.
Submit button	Submits the new record to the database.
Update button	Saves updates to the record.
Launch Task button	Manually launches the task.
Delete button	Deletes the current record.
View Parent button	Task instance only; Displays this task's parent task (workflow), if any.
Show Details button	Task instance only; displays detailed information about this task instance.
Skip button	Task instance only; for tasks loaded into the schedule that haven't run yet. Allows you to tell the system to skip this task. See Skipping a Task.
Hold button	Task instance only; see Putting a Task on Hold.
Force Finish button	Task instance only; see Force Finishing a Task.
Output tab	Task instance only. Displays output generated from the process, if any, based on specifications provided by the user in the Automatic Output Retrieval fields in the task definition.
Task Virtual Resources tab	Lists Virtual Resources to which this task is assigned. See Creating Virtual Resources.
Variables tab	Displays all variables associated with this record.

Actions tab	Allows you to specify actions that the system will take automatically based on events that occur during the execution of this task. Events are task instance status, exit codes, late start, late finish, and early finish. Actions are: • **Abort Action** - Abort the task if certain events occur. For details, see Setting Up Abort Actions. • **Email Notification** - Send an email if certain events occur. For details, see Creating Email Notifications. • **SNMP Notification** - Send an email if certain events occur. For details, see Creating SNMP Notifications. • **Set Variable** - Used in workflows to set a variable based on the occurrence of certain events. For details, see Setting Variables in a Workflow.
Task Instances tab	System-supplied. Displays a list of all task instances generated by this task definition.
Triggers tab	Displays a list of all triggers that have been defined to launch this task. Also allows you to add new triggers. If you add a new trigger from this location, the system automatically constructs a default trigger name as follows: <current task name>#TRIGGER#. You can change the default name if desired. For instructions on creating triggers, see Creating Triggers.
Notes tab	Displays all notes associated with this task. See Runbook Notes.
Versions tab	Stores copies of all previous versions of the current record (see Record Versioning).

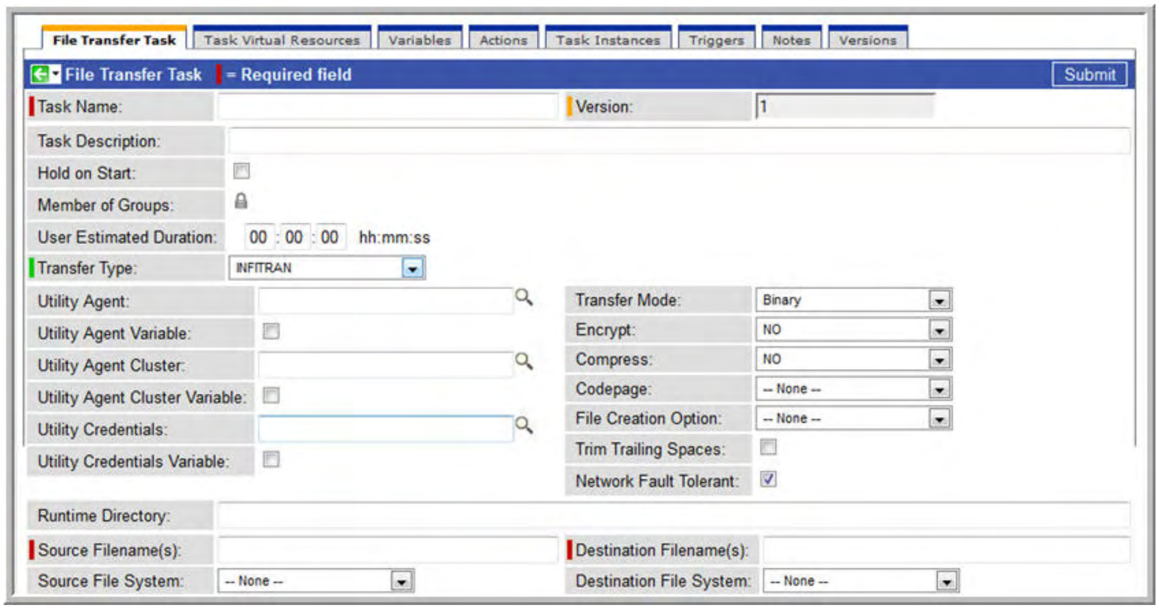
INFITRAN File Transfer Tasks

The screen for INFITRAN file transfers differs considerably from FTP and SFTP.

Creating an INFITRAN File Transfer Task

1. From the navigation nane, select **Automation Center > Tasks > File Transfer Tasks**.
2. Click **New**. Opswise displays the File Transfer Task screen.
3. For Transfer Type, select INFITRAN. Using the field descriptions provided below as a guide, complete the fields as needed.
4. Click the **Submit** button to save the record and return to the menu, or, right-click on the title bar and select **Save** to save the record and remain on the current display.
5. If appropriate, repeat these steps for any additional tasks you want to add.

Shown below is a sample INFITRAN File Transfer task screen. Task run statistics appear after the first time this task has been launched.



Version: 1

User Estimated Duration: 00 : 00 : 00 hh:mm:ss

Transfer Type: INFITRAN

Transfer Mode: Binary

Encrypt: NO

Compress: NO

Codepage: -- None --

File Creation Option: -- None --

Network Fault Tolerant: ☑

Source File System: -- None --

Destination File System: -- None --

INFITRAN File Transfer Task Field Descriptions

The table below describes the fields, buttons, and tabs on the task definition and task instance screens. Color coding is provided that differentiates the following three types of fields:

- Fields that display on the task definition and task instance screens are shown in black.
- Fields that display only on the task definition screen are shown in green.
- Fields that display only on the task instance screen are shown in maroon.

Field Name	Description
Task/Instance Name	Required. Name used within Opswise to identify this task. Up to 40 alphanumerics; variables supported. It is the user's responsibility to develop a workable naming scheme for tasks.
Version	Task definition only; system-supplied. The version number of the current record, which is incremented by the system every time a user updates a record. Click on the Versions tab to view previous versions. For details, see Record Versioning.
Description	User-supplied description of this record.
Invoked by	Task instance only; system-supplied. How the task instance was launched. One of the following: • Trigger: (Trigger Name) - The instance was launched by the named trigger. • Workflow: (Workflow Name) - The instance was launched by the named workflow. • Manually Launched - The instance was launched by a user. To determine the name of the user: 1. From the Activity or All Task Instances screen, click the task instance name to open the record. 2. The Execution User field identifies the user who launched the task instance.
Task	Task instance only; system-supplied. Hover over the paper icon to display more information about the task instance. Click the paper icon to display the task definition record.
Execution User	Task instance only; system-supplied. If the task was launched manually, the ID of the user who launched it.
Instance Reference Id	Task instance only; system-supplied. The system increments this number each time the task is run.
Hold on Start	If enabled, when the task is launched it appears in the Activity display with a status of Held. The task runs when the user releases it.
Hold Reason	Information about why the task will be put on hold when it starts.
Member of Groups	User-defined. Allows you to select one or more Opswise groups that this record definition belongs to. Click on the lock icon to unlock the field and select groups.
User Estimated Duration	Task definition only; optional. The estimated amount of time it should normally take to run this task. The system uses this information to calculate the User Estimated End Time on a task instance record.
Status	Task instance only; system-supplied. See Task Statuses.
Exit Code-Task Instance	Task instance only; system-supplied. The exit code captured by the agent when executing the task (for example, a command or script).

Status Description	Task instance only; system-supplied. Provides additional information, if any, about the status of the task.
Start Time	Task instance only; system-supplied. The date and time the task started.
Duration	Task instance only; system-supplied. The amount of time the task took to run.
End Time	Task instance only; system-supplied. The date and time the task instance completed.
Transfer Type	Type of File Transfer server. Options: • FTP • SFTP • INFITRAN
Utility Agent	Required. The name of the Windows, Linux/Unix, or z/OS agent resource that will communicate with the Indesca agent. If you do not specify an agent, you must specify an agent cluster (see below).
Transfer Mode	Mode of data transfer. Options: • Binary • ASCII
Utility Agent Variable	Optional. If enabled, the Utility Agent field (see above) converts from a reference field (where you browse and select a record) into a text field that allows you to enter a variable. Use the format: ${variable name}. The variable must be a supported type as described in Using Variables and Functions.
Encrypt	The method of encryption the system will use in the transfer. Options: • YES • NO - None. • RC4-SHA • RC4-MD5 • AES256-SHA • AES128-SHA • DES-CBC3-SHA • DES-CBC-SHA • NULL-SHA • NULL-MD5 • NULL-NULL
Utility Agent Cluster	Optional. You can specify an agent cluster in addition to or in place of a specific agent. An agent cluster is a group of agents, one of which the system will choose to run this task. If you specify an agent and an agent cluster, Opswise Automation Center first tries to run the task on the specific agent. If the agent is not available, the system reverts to the agent cluster. See Configuring Agent Clusters for more information
Compress	The type of data compression used in the transfer, if any. Options: • YES • NO • ZLIB • HASP

Utility Agent Cluster Variable	Optional. If enabled, the Agent Cluster field (see above) converts from a reference field (where you browse and select a record) into a text field that allows you to enter a variable. Use the format: ${variable name}. The variable must be a supported type as described in Using Variables and Functions.
Codepage	Options: • None • ISO 8859-1 • IBM Latin-1 1047
Utility Credentials	Login credentials the Opswise agent will use to access the Indesca server machine.
File Creation Option	Specifies whether the transferred file should be created (new), appended, or replace any existing file. Options: • None • APPEND • NEW • REPLACE
Utility Credentials Variable	Optional. If enabled, the Utility Credentials field (see above) converts from a reference field (where you browse and select a record) into a text field that allows you to enter a variable. Use the format: ${variable name}. . The variable must be a supported type as described in Using Variables and Functions.
Trim Trailing Spaces	Enabled or not. Specifies whether the system should trim trailing spaces from lines on an ASCII transfer.
Network Fault Tolerant	Enable if the session is network fault tolerant.
Runtime Directory	Optional. The directory from which the application should be executed. Variables supported.
Source Filename	The path and file name on the source Infitran server.
Destination Filename	The path and file name on the destination INFITRAN server.
Source File System	The type of file system on the source server. Options: • None • DSN • HFS • LIB
Destination File System	The type of file system on the destination server. Options: • None • DSN • HFS • LIB
Source Infitran Agent	Required. The name of the agent resource defined in Opswise that describes the source Indesca agent machine (primary transfer server).

Destination Infitran Agent	Required. The name of the agent resource defined in Opswise that provides details about the destination Indesca agent machine (secondary transfer server).
Source Infitran Agent Option	Defines how you will specify the Source Infitran Agent. Options: • Infitran Agent - The source agent is an Indesca/Infitran agent defined in Opswise. • Infitran Agent Variable - The source agent will be defined by setting the variable in the Source Infitran Agent field. • Infitran Agent Hostname - The source agent runs on the hostname specified in the Source Infitran Agent field.
Destination Infitran Agent Option	Defines how you will specify the Destination Infitran Agent. Options: • **Infitran Agent** - The destination agent is an Indesca/Infitran agent defined in Opswise. • **Infitran Agent Variable** - The destination agent will be defined by setting the variable in the Destination Infitran Agent field. • **Infitran Agent Hostname** - The destination agent runs on the hostname specified in the Destination Infitran Agent field.
Source Credentials	Specifies the source user ID and password (local to the host on which the server is running) under which the transfer operation is being carried out.
Destination Credentials	Specifies the destination user ID and password (local to the host on which the server is running) under which the transfer operation is being carried out.
Source Credentials Variable	Optional. If enabled, the Source Credentials field (see above) converts from a reference field (where you browse and select a credentials record from the OpsWise credentials table) into a text field that allows you to enter a variable. Use the format: ${variable name}. The variable must be a supported type as described in Using Variables and Functions.
Destination Credentials Variable	Optional. If enabled, the Destination Credentials field (see above) converts from a reference field (where you browse and select a credentials record from the OpsWise credentials table) into a text field that allows you to enter a variable. Use the format: ${variable name}. The variable must be a supported type as described in Using Variables and Functions.
Append Source Open Options	Optional. Any additional free form open command options for the source (primary) transfer server.
Append Destination Open Option	Optional. Any additional free form open command options for the destination (secondary) transfer server.
Append Infitran Options	Optional. Any additional free form Universal Data Mover command options.
Exit Code Processing	Required. Specifies how Opswise Automation Center should determine whether the executed command failed or completed successfully. Options: • **Success Exitcode Range** - The command is considered completed successfully if its exit code falls within the range specified in the Exit Codes field (see below). • **Failure Exitcode Range** - The command is considered failed if its exit code falls within the range specified in the Exit Codes field (see below). • **Success Output Contains** - The command is considered completed successfully if its output contains the text specified in the Scan Output For field (see below). • **Failure Output Contains** - The command is considered failed if its output contains the text specified in the Scan Output For field (see below). • **Step Conditions** - The command is considered completed successfully/failed if any of its specified condition codes falls within the range specified under the Step Conditions tab (see Creating Step Conditions).

Output Type-Exit Code	Required if Exit Code Processing = Success Output Contains or Failure Output Contains. This field specifies the type of output. Options: • Standard Output (STDOUT) • Standard Error (STDERR) • File
Exit Codes	Required if Exit Code Processing = Success Exitcode Range or Failure Exit Code Range. This field specifies the range. Format: Numeric. Use commas to list a series of exitcodes; use hyphens to specify a range. Example: 1,5, 22-30.
Scan Output For	Required if Exit Code Processing = Success Output Contains or Failure Output Contains. This field specifies the text for which Opswise should scan the output file. Opswise will process this field as a regular expression.
Output File-Exit Codes	Required if Output Type = File. This field specifies the path and file name of the output file that should be scanned for the text in the Scan Output For field.
Maximum Retries	User-defined. The maximum number of times the system should retry this task after it has started and gone to a failed state.
Retry Indefinitely	User-defined. Enabled or disabled. Indicates whether the system should continue trying indefinitely to run this task. If you enable this field, it overrides any value placed in the Maximum Retries field (above).
Retry Interval	User-defined. The number of seconds between each retry.
Current Retry Count	Task instance only; system-supplied. Displays only for a running task instance. Displays the current number of times the system has retried the task after it first went to failure status.
User Estimated End Time	Task instance only; system-supplied. If the user entered information into the User Estimated Duration field in the task definition, the system uses this information to calculate an end time for the task instance, based on the date/time the task instance started.
Shortest Estimated End Time	Task instance only; system-supplied.
Average Estimated End Time	Task instance only; system-supplied.
Longest Estimated End Time	Task instance only; system-supplied.
Late Start	If enabled, and if the task instance starts after the time specified in the Late Start Time field (see below), the task instance is flagged as late. To determine whether a task instance started late, open the task instance and locate the Started Late field; the field is checked if the instance started late.
Late Start Type	Required if Late Start is enabled. Options are: • **Time** - Flag the task if it starts after the specified time. • **Duration** - Flag the task if it starts a certain amount of time after the programmed start time. The task must have a specific start time.

Late Start Time	Time after which the task start time is considered late. Use hh:mm, 24-hour time
Late Start Duration	Duration (amount of relative time) after which the task is considered to have started late. For a task within a workflow, the duration is the period between the time the workflow starts and the time the task itself starts. For example, a task might have a Late Start Duration of 60 minutes. If the workflow starts at 9:00 a.m. but the task itself does not start until 10:30, the task has started late. For a task that is not within a workflow, Late Start Duration has meaning only if the task has been held upon starting. For example, if a task has a Late Start Duration of 60 minutes and the Hold on Start field is enabled, if the task is not released from hold within the amount of time specified in the Late Start Duration field, the task has started late.
Started Late	Task instance only; system-supplied. This field is flagged if the task started later than the time specified in the Late Start fields.
Late Finish	If enabled, and if the task instance finishes after the time or period specified, the task instance is flagged as late. You can specify a time or duration to determine a late finish (see Late Finish Type below). To determine whether a task instance finished late, open the task instance and locate the Finished Late field; the field is checked if the instance finished after the specified time or lasted longer than expected. This field only appears on the task instance if the user specified a Late Finish in the task definition.
Late Finish Type	If Late Finish is enabled, use this field to specify whether you want to use time (see Late Finish Time below) or duration (see Late Finish Duration below) to determine whether the task instance finished late.
Late Finish Time	If Late Finish Type is Time, use this to specify the time after which the task finish time is considered late. Use hh:mm, 24-hour time.
Late Finish Duration	If Late Finish Type is Duration, use this to specify the longest amount of time this task instance should take to run. You can specify any combination of hours, minutes, and seconds.
Finished Late	Task instance only; system-supplied. This field is flagged if the task finished later than the time or duration specified in the Late Finish fields.
Early Finish	If enabled, and if the task instance finishes before the time or period specified, the task instance is flagged as early. You can specify a time or duration to determine an early finish (see Early Finish Type below). To determine whether a task instance finished early, open the task instance and locate the Finished Early field; the field is checked if the instance finished before the specified time or did not last as long as expected. This field only appears on the task instance if the user added Early Finish specifications to the task definition.
Early Finish Type	If Early Finish is enabled, use this field to specify whether you want to use time (see Early Finish Time below) or duration (see Early Finish Duration below) to determine whether the task instance finished early.
Early Finish Time	If Early Finish Type is Time, use this to specify the time before which the task finish time is considered early. That is, enter a time at which the task should still be running. Use hh:mm, 24-hour time.
Early Finish Duration	If Early Finish Type is Duration, use this to specify the shortest amount of time this task instance should take to run. You can specify any combination of hours, minutes, and seconds.
Finished Early	Task instance only; system-supplied. This field is flagged if the task finished earlier than the time specified in the Early Finish fields.

Submit button	Submits the new record to the database.
Update button	Saves updates to the record.
Launch Task button	Manually launches the task.
Delete button	Deletes the current record.
View Parent button	Task instance only; Displays this task's parent task (workflow), if any.
Show Details button	Task instance only; displays detailed information about this task instance.
Skip button	Task instance only; for tasks loaded into the schedule that haven't run yet. Allows you to tell the system to skip this task. See Skipping a Task.
Hold button	Task instance only; see Putting a Task on Hold.
Force Finish button	Task instance only; see Force Finishing a Task.
Re-run button	Task instance only; see Re-running a Task.
Output tab	Task instance only. Displays output generated from the process, if any, based on specifications provided by the user in the Automatic Output Retrieval fields in the task definition.
Task Virtual Resources tab	Lists Virtual Resources to which this task is assigned. See Creating Virtual Resources.
Variables tab	Displays all variables associated with this record.
Actions tab	Allows you to specify actions that the system will take automatically based on events that occur during the execution of this task. Events are task instance status, exit codes, late start, late finish, and early finish. Actions are: • **Abort Action** - Abort the task if certain events occur. For details, see Setting Up Abort Actions. • **Email Notification** - Send an email if certain events occur. For details, see Creating Email Notifications. • **SNMP Notification** - Send an email if certain events occur. For details, see Creating SNMP Notifications. • **Set Variable** - Used in workflows to set a variable based on the occurrence of certain events. For details, see Setting Variables in a Workflow.
Task Instances tab	System-supplied. Displays a list of all task instances generated by this task definition.
Triggers tab	Displays a list of all triggers that have been defined to launch this task. Also allows you to add new triggers. If you add a new trigger from this location, the system automatically constructs a default trigger name as follows: <current task name>#TRIGGER#. You can change the default name if desired. For instructions on creating triggers, see Creating Triggers.

Notes tab	Displays all notes associated with this task. See Runbook Notes.
Versions tab	Stores copies of all previous versions of the current record (see Record Versioning).

Specifying When a Task Runs

You can run the task as part of a workflow, specify triggers that run the task automatically based on times or events, or run the task manually.

Monitoring Task Execution

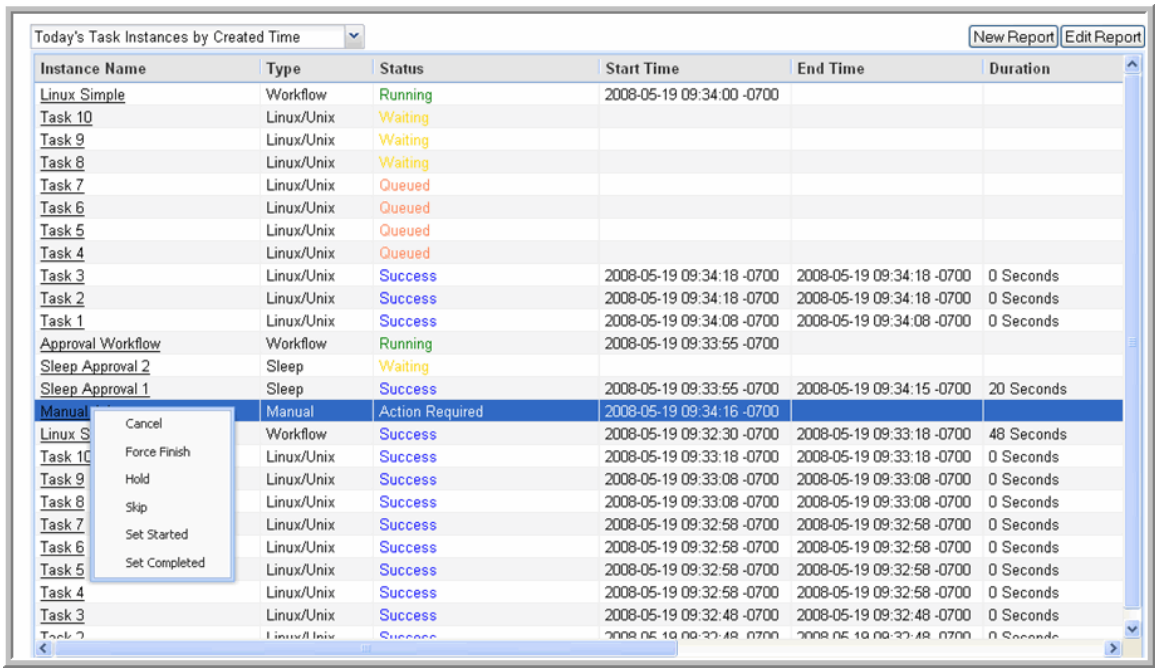
You can monitor all system activity from the Activity display and can view activity history from the Activity History screen.

Manual Task

Overview

Manual tasks are used to create a pause in the workflow during which the user must take some action. The processing on a Manual task within a workflow is described below.

1. While its upstream tasks are running, the Manual task remains in WAITING status. When the Manual task launches, it goes immediately into ACTION REQUIRED status, meaning a user must perform some manual activity. The system also sets the **Started Time** in the Manual task instance to the time the task goes into the ACTION REQUIRED status.
2. Optionally, the user can re-set the Started Time on the Manual task by issuing the **Set Started** command as follows:
   a. From the Activity display or the Workflow monitor, right click on the Manual task.
   b. Select **Set Started** or, if you are using the Workflow monitor, select **Commands > Set Started**.



Instance Name	Type	Status	Start Time	End Time	Duration
Linux Simple	Workflow	Running	2008-05-19 09:34:00 -0700		
Task 10	Linux/Unix	Waiting			
Task 9	Linux/Unix	Waiting			
Task 8	Linux/Unix	Waiting			
Task 7	Linux/Unix	Queued			
Task 6	Linux/Unix	Queued			
Task 5	Linux/Unix	Queued			
Task 4	Linux/Unix	Queued			
Task 3	Linux/Unix	Success	2008-05-19 09:34:18 -0700	2008-05-19 09:34:18 -0700	0 Seconds
Task 2	Linux/Unix	Success	2008-05-19 09:34:18 -0700	2008-05-19 09:34:18 -0700	0 Seconds
Task 1	Linux/Unix	Success	2008-05-19 09:34:08 -0700	2008-05-19 09:34:08 -0700	0 Seconds
Approval Workflow	Workflow	Running	2008-05-19 09:33:55 -0700		
Sleep Approval 2	Sleep	Waiting			
Sleep Approval 1	Sleep	Success	2008-05-19 09:33:55 -0700	2008-05-19 09:34:15 -0700	20 Seconds
Manual	Manual	Action Required	2008-05-19 09:34:16 -0700		
Linux S	Workflow	Success	2008-05-19 09:32:30 -0700	2008-05-19 09:33:18 -0700	48 Seconds
Task 10	Linux/Unix	Success	2008-05-19 09:33:18 -0700	2008-05-19 09:33:18 -0700	0 Seconds
Task 9	Linux/Unix	Success	2008-05-19 09:33:08 -0700	2008-05-19 09:33:08 -0700	0 Seconds
Task 8	Linux/Unix	Success	2008-05-19 09:33:08 -0700	2008-05-19 09:33:08 -0700	0 Seconds
Task 7	Linux/Unix	Success	2008-05-19 09:32:58 -0700	2008-05-19 09:32:58 -0700	0 Seconds
Task 6	Linux/Unix	Success	2008-05-19 09:32:58 -0700	2008-05-19 09:32:58 -0700	0 Seconds
Task 5	Linux/Unix	Success	2008-05-19 09:32:58 -0700	2008-05-19 09:32:58 -0700	0 Seconds
Task 4	Linux/Unix	Success	2008-05-19 09:32:58 -0700	2008-05-19 09:32:58 -0700	0 Seconds
Task 3	Linux/Unix	Success	2008-05-19 09:32:48 -0700	2008-05-19 09:32:48 -0700	0 Seconds
Task 2	Linux/Unix	Success	2008-05-19 09:32:48 -0700	2008-05-19 09:32:48 -0700	0 Seconds

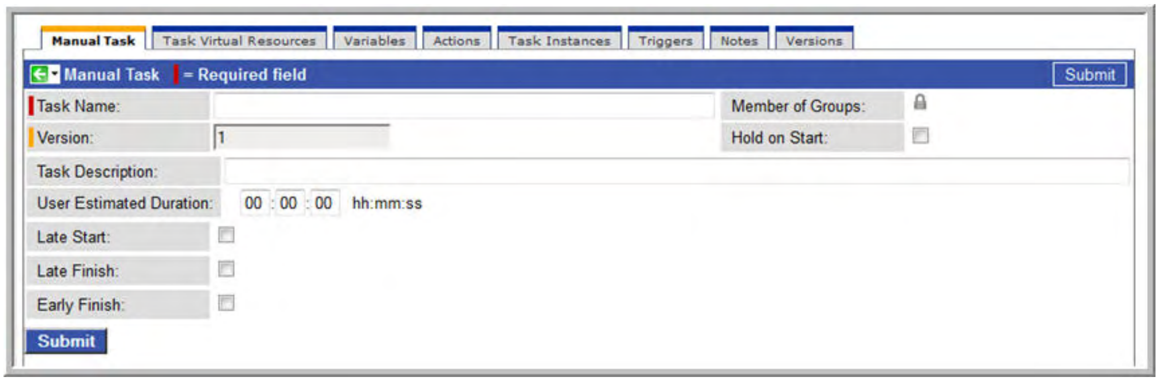
The system re-sets the Started Time.

3. When you have completed the activities called for in the Manual task, you need to indicate that the task is completed and that the workflow can continue, as follows:
   a. From the Activity display or the Workflow monitor, right click on the Manual task.
   b. Select **Set Completed** or select **Commands > Set Completed** if you are using the Workflow monitor. The Manual task goes into SUCCESS status, the End Time is set, and the workflow continues.
   Or, if the Manual task is not completed but you want the workflow to continue anyway, select **Force Finish**.

You can also set a Manual task to STARTED or COMPLETED status from the command line interface. See Command Line Reference for instructions.

Creating a New Manual Task

1. From the navigation pane, select **Automation Center > Tasks > Manual Tasks**. The Manual Tasks List screen displays.
2. Click **New**. The Manual Task Definition screen displays.



Manual Task = Required field

Task Name:

Member of Groups:

Version: 1

Hold on Start: □

Task Description:

User Estimated Duration: 00 : 00 : 00 hh:mm:ss

Late Start: □

Late Finish: □

Early Finish: □

3. Using the field descriptions provided below as a guide, complete the fields as needed.
4. Click the **Submit** button to save the record and return to the menu, or right-click on the title bar and select **Save** to save the record and remain on the current display.
5. If appropriate, repeat these steps for any additional tasks you want to add.

Task run statistics will show at the bottom after the first time this task has been launched.

Manual Task Field Descriptions

The table below describes the fields, buttons, and tabs on the task definition and task instance screens. Color coding is provided that differentiates the following three types of fields:

- Fields that display on the task definition and task instance screens are shown in black.
- Fields that display only on the task definition screen are shown in green.
- Fields that display only on the task instance screen are shown in maroon.

Field Name	Description
Task/Instance Name	Required. Name used within Opswise to identify this task. Up to 40 alphanumerics; variables supported. It is the user's responsibility to develop a workable naming scheme for tasks.
Invoked by	Task instance only; system-supplied. How the task instance was launched. One of the following: • Trigger: (Trigger Name) - The instance was launched by the named trigger. • Workflow: (Workflow Name) - The instance was launched by the named workflow. • Manually Launched - The instance was launched by a user. To determine the name of the user: 1. From the Activity or All Task Instances screen, click the task instance name to open the record. 2. The Execution User field identifies the user who launched the task instance.
Task	Task instance only; system-supplied. Hover over the paper icon to display more information about the task instance. Click the paper icon to display the task definition record.
Instance Reference Id	Task instance only; system-supplied. The system increments this number each time the task is run.
Member of Groups	User-defined. Allows you to select one or more Opswise groups that this record definition belongs to. Click on the lock icon to unlock the field and select groups.
Version	Task definition only; system-supplied. The version number of the current record, which is incremented by the system every time a user updates a record. Click on the Versions tab to view previous versions. For details, see Record Versioning.
Task Description	User-supplied description of this record.

Status	Task instance only; system-supplied. See Task Statuses.
Exit Code	Task instance only; system-supplied. The exit code captured by the agent when executing the task (for example, a command or script).
Status Description	Task instance only; system-supplied. Provides additional information, if any, about the status of the task.
Start Time	Task instance only; system-supplied. The date and time the task started.
Duration	Task instance only; system-supplied. The amount of time the task took to run.
End Time	Task instance only; system-supplied. The date and time the task instance completed.
User Estimated End Time	Task instance only; system-supplied. If the user entered information into the User Estimated Duration field in the task definition, the system uses this information to calculate an end time for the task instance, based on the date/time the task instance started.
Shortest Estimated End Time	Task instance only; system-supplied.
Average Estimated End Time	Task instance only; system-supplied.
Longest Estimated End Time	Task instance only; system-supplied.
Hold on Start	If enabled, when the task is launched it appears in the Activity display with a status of Held. The task runs when the user releases it.
Hold Reason	Information about why the task will be put on hold when it starts.
User Estimated Duration	Task definition only; optional. The estimated amount of time it should normally take to run this task. The system uses this information to calculate the User Estimated End Time on a task instance record.
Late Start	If enabled, and if the task instance starts after the time specified in the Late Start Time field (see below), the task instance is flagged as late. To determine whether a task instance started late, open the task instance and locate the Started Late field; the field is checked if the instance started late.
Late Start Type	Required if Late Start is enabled. Options are: • **Time** - Flag the task if it starts after the specified time. • **Duration** - Flag the task if it starts a certain amount of time after the programmed start time. The task must have a specific start time.
Late Start Time	Time after which the task start time is considered late. Use hh:mm, 24-hour time

Late Start Duration	Duration (amount of relative time) after which the task is considered to have started late. For a task within a workflow, the duration is the period between the time the workflow starts and the time the task itself starts. For example, a task might have a Late Start Duration of 60 minutes. If the workflow starts at 9:00 a.m. but the task itself does not start until 10:30, the task has started late. For a task that is not within a workflow, Late Start Duration has meaning only if the task has been held upon starting. For example, if a task has a Late Start Duration of 60 minutes and the Hold on Start field is enabled, if the task is not released from hold within the amount of time specified in the Late Start Duration field, the task has started late.
Started Late	Task instance only; system-supplied. This field is flagged if the task started later than the time specified in the Late Start fields.
Late Finish	If enabled, and if the task instance finishes after the time or period specified, the task instance is flagged as late. You can specify a time or duration to determine a late finish (see Late Finish Type below). To determine whether a task instance finished late, open the task instance and locate the Finished Late field; the field is checked if the instance finished after the specified time or lasted longer than expected. This field only appears on the task instance if the user specified a Late Finish in the task definition.
Late Finish Type	If Late Finish is enabled, use this field to specify whether you want to use time (see Late Finish Time below) or duration (see Late Finish Duration below) to determine whether the task instance finished late.
Late Finish Time	If Late Finish Type is Time, use this to specify the time after which the task finish time is considered late. Use hh:mm, 24-hour time.
Late Finish Duration	If Late Finish Type is Duration, use this to specify the longest amount of time this task instance should take to run. You can specify any combination of hours, minutes, and seconds.
Finished Late	Task instance only; system-supplied. This field is flagged if the task finished later than the time or duration specified in the Late Finish fields.
Early Finish	If enabled, and if the task instance finishes before the time or period specified, the task instance is flagged as early. You can specify a time or duration to determine an early finish (see Early Finish Type below). To determine whether a task instance finished early, open the task instance and locate the Finished Early field; the field is checked if the instance finished before the specified time or did not last as long as expected. This field only appears on the task instance if the user added Early Finish specifications to the task definition.
Early Finish Type	If Early Finish is enabled, use this field to specify whether you want to use time (see Early Finish Time below) or duration (see Early Finish Duration below) to determine whether the task instance finished early.
Early Finish Time	If Early Finish Type is Time, use this to specify the time before which the task finish time is considered early. That is, enter a time at which the task should still be running. Use hh:mm, 24-hour time.
Early Finish Duration	If Early Finish Type is Duration, use this to specify the shortest amount of time this task instance should take to run. You can specify any combination of hours, minutes, and seconds.
Finished Early	Task instance only; system-supplied. This field is flagged if the task finished earlier than the time specified in the Early Finish fields.
First Time Ran	Task definition only; system-supplied. Displays after the first time the task runs. The date and time this task first ran.

Lowest Instance Time	Task definition only; system-supplied. Displays after the first time the task runs. The shortest amount of time this task has taken to run.
Last Time Ran	Task definition only; system-supplied. Displays after the first time the task runs. The most recent date and time the task ran.
Average Instance Time	Task definition only; system-supplied. Displays after the first time the task runs. Shows the average amount of time this task takes to run.
Number of Instances	Task definition only; system-supplied. Displays after the first time the task runs. Shows the number of times this task has run.
Highest Instance Time	Task definition only; system-supplied. Displays after the first time the task runs. The longest amount of time this task has taken to run.
Last Instance Duration	Task definition only; system-supplied. Displays after the first time the task runs. The amount of time the task took to run the last time it ran.
Submit button	Submits the new record to the database.
Update button	Saves updates to the record.
Launch Task button	Manually launches the task.
Delete button	Deletes the current record.
View Parent button	Task instance only; Displays this task's parent task (workflow), if any.
Show Details button	Task instance only; displays detailed information about this task instance.
Skip button	Task instance only; for tasks loaded into the schedule that haven't run yet. Allows you to tell the system to skip this task. See Skipping a Task.
Hold button	Task instance only; see Putting a Task on Hold.
Force Finish button	Task instance only; see Force Finishing a Task.
Re-run button	Task instance only; see Re-running a Task.
Output tab	Task instance only. Displays output generated from the process, if any, based on specifications provided by the user in the Automatic Output Retrieval fields in the task definition.
Task Virtual Resources tab	Lists Virtual Resources to which this task is assigned. See Creating Virtual Resources.

Variables tab	Displays all variables associated with this record.
Actions tab	Allows you to specify actions that the system will take automatically based on events that occur during the execution of this task. Events are task instance status, exit codes, late start, late finish, and early finish. Actions are: • **Abort Action** - Abort the task if certain events occur. For details, see Setting Up Abort Actions. • **Email Notification** - Send an email if certain events occur. For details, see Creating Email Notifications. • **SNMP Notification** - Send an email if certain events occur. For details, see Creating SNMP Notifications. • **Set Variable** - Used in workflows to set a variable based on the occurrence of certain events. For details, see Setting Variables in a Workflow.
Task Instances tab	System-supplied. Displays a list of all task instances generated by this task definition.
Triggers tab	Displays a list of all triggers that have been defined to launch this task. Also allows you to add new triggers. If you add a new trigger from this location, the system automatically constructs a default trigger name as follows: <current task name>#TRIGGER#. You can change the default name if desired. For instructions on creating triggers, see Creating Triggers.
Notes tab	Displays all notes associated with this task. See Runbook Notes.
Versions Tab	Stores copies of all previous versions of the current record (see Record Versioning).

Specifying When a Task Runs

You can run the task as part of a workflow, specify triggers that run the task automatically based on times or events, or run the task manually.

Monitoring Task Execution

You can monitor all system activity from the Activity display and can view activity history from the Activity History screen.

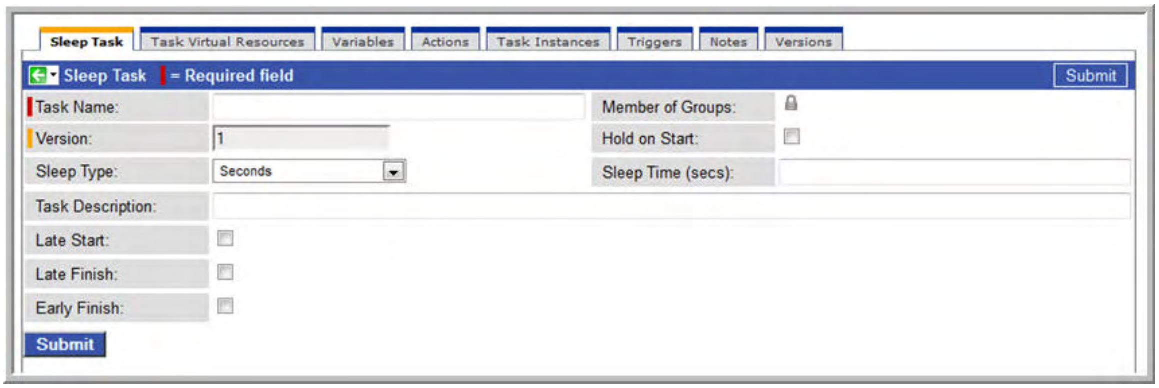
Sleep Task

Overview

The Sleep task allows you to execute a sleep command for a specified period of time or until a specific time. This task is helpful, for example, if you need to impose a pause in the processing of a workflow.

Creating a New Sleep Task

1. From the navigation pane, select **Automation Center > Tasks > Sleep Tasks**. The Sleep Tasks List screen displays.
2. Click **New**. The Sleep Task Definition screen displays.



Task Name:		Member of Groups:	
Version:	1	Hold on Start:	□
Sleep Type:	Seconds	Sleep Time (secs):	
Task Description:			
Late Start:	□		
Late Finish:	□		
Early Finish:	□		

3. Using the field descriptions provided below as a guide, complete the fields as needed.
4. Click the **Submit** button to save the record and return to the menu, or right-click on the title bar and select **Save** to save the record and remain on the current display.
5. If appropriate, repeat these steps for any additional tasks you want to add.

Task run statistics appear after the first time this task has been launched.

Sleep Task Field Descriptions

The table below describes the fields, buttons, and tabs on the task definition and task instance screens. Color coding is provided that differentiates the following three types of fields:

- Fields that display on the task definition and task instance screens are shown in black.
- Fields that display only on the task definition screen are shown in green.
- Fields that display only on the task instance screen are shown in maroon.

Field Name	Description
Task/Instance Name	Required. Name used within Opswise to identify this task. Up to 40 alphanumerics; variables supported. It is the user's responsibility to develop a workable naming scheme for tasks.

Invoked by	Task instance only; system-supplied. How the task instance was launched. One of the following: • Trigger: (Trigger Name) - The instance was launched by the named trigger. • Workflow: (Workflow Name) - The instance was launched by the named workflow. • Manually Launched - The instance was launched by a user. To determine the name of the user: 1. From the Activity or All Task Instances screen, click the task instance name to open the record. 2. The Execution User field identifies the user who launched the task instance.
Task	Task instance only; system-supplied. Hover over the paper icon to display more information about the task instance. Click the paper icon to display the task definition record.
Instance Reference Id	Task instance only; system-supplied. The system increments this number each time the task is run.
Member of Groups	User-defined. Allows you to select one or more Opswise groups that this record definition belongs to. Click on the lock icon to unlock the field and select groups.
Version	Task definition only; system-supplied. The version number of the current record, which is incremented by the system every time a user updates a record. Click on the Versions tab to view previous versions. For details, see Record Versioning.
Sleep Type	User-supplied. The type of sleep command you want to execute. Options: • Seconds - Use the Sleep Time (secs) field to specify the number of seconds. • Duration - Use the Sleep Duration field to specify the number of days, hours, minutes, and/or seconds. • Time - Use the Sleep Until Time (hh:mm) field to specify the time you want the sleep command to complete.
Sleep Time	User-supplied. If Sleep Type = Seconds, the number of seconds the Sleep should last.
Sleep Duration	User-supplied. If Sleep Type = Duration, the number of days, hours, minutes and/or seconds the Sleep should last.
Sleep Until Time	User-supplied. If Sleep Type = Time, the time the Sleep command should go to complete status. Use 24-hour time
Task Description	User-supplied description of this record.
Status	Task instance only; system-supplied. See Task Statuses.
Exit Code	Task instance only; system-supplied. The exit code captured by the agent when executing the task (for example, a command or script).
Status Description	Task instance only; system-supplied. Provides additional information, if any, about the status of the task.
Start Time	Task instance only; system-supplied. The date and time the task started.
End Time	Task instance only; system-supplied. The date and time the task instance completed.
Duration	Task instance only; system-supplied. The amount of time the task took to run.

Hold on Start	If enabled, when the task is launched it appears in the Activity display with a status of Held. The task runs when the user releases it.
Hold Reason	Information about why the task will be put on hold when it starts.
Late Start	If enabled, and if the task instance starts after the time specified in the Late Start Time field (see below), the task instance is flagged as late. To determine whether a task instance started late, open the task instance and locate the Started Late field; the field is checked if the instance started late.
Late Start Type	Required if Late Start is enabled. Options are: - **Time** - Flag the task if it starts after the specified time. - **Duration** - Flag the task if it starts a certain amount of time after the programmed start time. The task must have a specific start time.
Late Start Time	Time after which the task start time is considered late. Use hh:mm, 24-hour time
Late Start Duration	Duration (amount of relative time) after which the task is considered to have started late. For a task within a workflow, the duration is the period between the time the workflow starts and the time the task itself starts. For example, a task might have a Late Start Duration of 60 minutes. If the workflow starts at 9:00 a.m. but the task itself does not start until 10:30, the task has started late. For a task that is not within a workflow, Late Start Duration has meaning only if the task has been held upon starting. For example, if a task has a Late Start Duration of 60 minutes and the Hold on Start field is enabled, if the task is not released from hold within the amount of time specified in the Late Start Duration field, the task has started late.
Started Late	Task instance only; system-supplied. This field is flagged if the task started later than the time specified in the Late Start fields.
Late Finish	If enabled, and if the task instance finishes after the time or period specified, the task instance is flagged as late. You can specify a time or duration to determine a late finish (see Late Finish Type below). To determine whether a task instance finished late, open the task instance and locate the Finished Late field; the field is checked if the instance finished after the specified time or lasted longer than expected. This field only appears on the task instance if the user specified a Late Finish in the task definition.
Late Finish Type	If Late Finish is enabled, use this field to specify whether you want to use time (see Late Finish Time below) or duration (see Late Finish Duration below) to determine whether the task instance finished late.
Late Finish Time	If Late Finish Type is Time, use this to specify the time after which the task finish time is considered late. Use hh:mm, 24-hour time.
Late Finish Duration	If Late Finish Type is Duration, use this to specify the longest amount of time this task instance should take to run. You can specify any combination of hours, minutes, and seconds.
Finished Late	Task instance only; system-supplied. This field is flagged if the task finished later than the time or duration specified in the Late Finish fields.

Early Finish	If enabled, and if the task instance finishes before the time or period specified, the task instance is flagged as early. You can specify a time or duration to determine an early finish (see Early Finish Type below). To determine whether a task instance finished early, open the task instance and locate the Finished Early field; the field is checked if the instance finished before the specified time or did not last as long as expected. This field only appears on the task instance if the user added Early Finish specifications to the task definition.
Early Finish Type	If Early Finish is enabled, use this field to specify whether you want to use time (see Early Finish Time below) or duration (see Early Finish Duration below) to determine whether the task instance finished early.
Early Finish Time	If Early Finish Type is Time, use this to specify the time before which the task finish time is considered early. That is, enter a time at which the task should still be running. Use hh:mm, 24-hour time.
Early Finish Duration	If Early Finish Type is Duration, use this to specify the shortest amount of time this task instance should take to run. You can specify any combination of hours, minutes, and seconds.
Finished Early	Task instance only; system-supplied. This field is flagged if the task finished earlier than the time specified in the Early Finish fields.
First Time Ran	Task definition only; system-supplied. Displays after the first time the task runs. The date and time this task first ran.
Last Time Ran	Task definition only; system-supplied. Displays after the first time the task runs. The most recent date and time the task ran.
Number of Instances	Task definition only; system-supplied. Displays after the first time the task runs. Shows the number of times this task has run.
Submit button	Submits the new record to the database.
Update button	Saves updates to the record.
Launch Task button	Manually launches the task.
Delete button	Deletes the current record.
View Parent button	Task instance only; Displays this task's parent task (workflow), if any.
Show Details button	Task instance only; displays detailed information about this task instance.
Skip button	Task instance only; for tasks loaded into the schedule that haven't run yet. Allows you to tell the system to skip this task. See Skipping a Task.
Hold button	Task instance only; see Putting a Task on Hold.
Force Finish button	Task instance only; see Force Finishing a Task.

Re-run button	Task instance only; see Re-running a Task.
Output tab	Task instance only. Displays output generated from the process, if any, based on specifications provided by the user in the Automatic Output Retrieval fields in the task definition.
Task Virtual Resources tab	Lists Virtual Resources to which this task is assigned. See Creating Virtual Resources.
Variables tab	Displays all variables associated with this record.
Actions tab	Allows you to specify actions that the system will take automatically based on events that occur during the execution of this task. Events are task instance status, exit codes, late start, late finish, and early finish. Actions are: • **Abort Action** - Abort the task if certain events occur. For details, see Setting Up Abort Actions. • **Email Notification** - Send an email if certain events occur. For details, see Creating Email Notifications. • **SNMP Notification** - Send an email if certain events occur. For details, see Creating SNMP Notifications. • **Set Variable** - Used in workflows to set a variable based on the occurrence of certain events. For details, see Setting Variables in a Workflow.
Task Instances tab	System-supplied. Displays a list of all task instances generated by this task definition.
Triggers tab	Displays a list of all triggers that have been defined to launch this task. Also allows you to add new triggers. If you add a new trigger from this location, the system automatically constructs a default trigger name as follows: <current task name>#TRIGGER#. You can change the default name if desired. For instructions on creating triggers, see Creating Triggers.
Notes tab	Displays all notes associated with this task. See Runbook Notes.
Versions Tab	Stores copies of all previous versions of the current record (see Record Versioning).

Specifying When a Task Runs

You can run the task as part of a workflow, specify triggers that run the task automatically based on times or events, or run the task manually.

Monitoring Task Execution

You can monitor all system activity from the Activity display and can view activity history from the Activity History screen.

SQL Task

Before You Begin

The SQL task allows you to execute an SQL statement against a database. To run an SQL task, you first need to create a Database Connection, which defines the information needed to locate and access the database.

Overview

The SQL task allows you to execute one or a series of SQL statements against the database specified in the task.

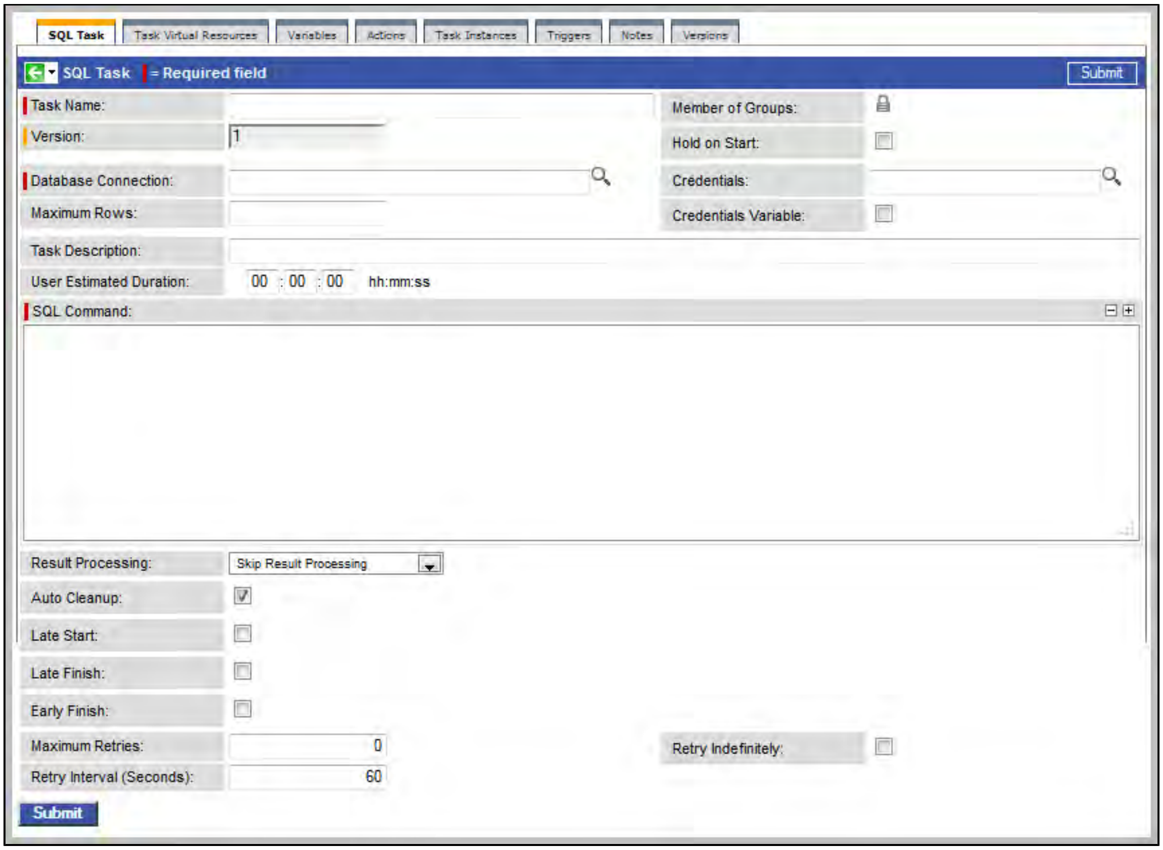
Built-In Variables

The built-in variables outlined below can be used in a SQL task to pass data where appropriate:

- Task and Task Instance Variables
- SQL and Stored Procedure Variables

Creating a New SQL Task

1. From the navigation pane, select **Automation Center > Tasks > SQL Tasks**. The SQL Tasks List screen displays.
2. Click **New**. The SQL Task Definition screen displays.



Task Name:		Member of Groups:	
Version:	1	Hold on Start:	□
Database Connection:		Credentials:	
Maximum Rows:		Credentials Variable:	□
Task Description:			
User Estimated Duration:	00 : 00 : 00 hh:mm:ss		
SQL Command:			
Result Processing:	Skip Result Processing		
Auto Cleanup:	☑		
Late Start:	□		
Late Finish:	□		
Early Finish:	□		
Maximum Retries:	0	Retry Indefinitely:	□
Retry Interval (Seconds):	60		

3. Using the field descriptions provided below as a guide, complete the fields as needed.
4. Click the **Submit** button to save the record and return to the menu, or right-click on the title bar and select **Save** to save the record and remain on the current display.
5. If appropriate, repeat these steps for any additional tasks you want to add.

Task run statistics appear after the first time this task has been launched.

SQL Task Field Descriptions

The table below describes the fields, buttons, and tabs on the task definition and task instance screens. Color coding is provided that differentiates the following three types of fields:

- Fields that display on the task definition and task instance screens are shown in black.
- Fields that display only on the task definition screen are shown in green.
- Fields that display only on the task instance screen are shown in maroon.

Field Name	Description
Task/Instance Name	Required. Name used within Opswise to identify this task. Up to 40 alphanumerics; variables supported. It is the user's responsibility to develop a workable naming scheme for tasks.
Member of Groups	User-defined. Allows you to select one or more Opswise groups that this record definition belongs to. Click on the lock icon to unlock the field and select groups.
Version	Task definition only; system-supplied. The version number of the current record, which is incremented by the system every time a user updates a record. Click on the Versions tab to view previous versions. For details, see Record Versioning.

Invoked by	Task instance only; system-supplied. How the task instance was launched. One of the following: • Trigger: (Trigger Name) - The instance was launched by the named trigger. • Workflow: (Workflow Name) - The instance was launched by the named workflow. • Manually Launched - The instance was launched by a user. To determine the name of the user: 1. From the Activity or All Task Instances screen, click the task instance name to open the record. 2. The Execution User field identifies the user who launched the task instance.
Task	Task instance only; system-supplied. Hover over the paper icon to display more information about the task instance. Click the paper icon to display the task definition record.
Execution User	Task instance only; system-supplied. If the task was launched manually, the ID of the user who launched it.
Instance Reference Id	Task instance only; system-supplied. The system increments this number each time the task is run.
Hold on Start	If enabled, when the task is launched it appears in the Activity display with a status of Held. The task runs when the user releases it.
Hold Reason	Information about why the task will be put on hold when it starts.
Database Connection field	Optional. The name of the Opswise Automation Center database connection that defines the database. Click the magnifying glass to browse for an existing database connection or add a new one.
Credentials	Login credentials Opswise Automation Center will use to access the remote machine. They must be provided either in the resource definition or task definition, or both. Credentials provided in the task definition override any credentials provided in the agent resource definition. See Credentials.
Maximum Rows	Optional. If necessary, specify a limit to the number of rows you want returned by the SQL statement. This value overrides any value you specify in the database connection.
Credentials Variable	Optional. If enabled, the Credentials field (see above) converts from a reference field (where you browse and select a record) into a text field that allows you to enter a variable. Use the format: ${variable name}. The variable must be a supported type as described in Using Variables and Functions .
Task Description	User-supplied description of this record.
Status	Task instance only; system-supplied. See Task Statuses.
Exit Code	Task instance only; system-supplied. The vendor-specific exception code for the SQL exception.
Status Description	Task instance only; system-supplied. Provides additional information, if any, about the status of the task.
Start Time	Task instance only; system-supplied. The date and time the task started.
Duration	Task instance only; system-supplied. The amount of time the task took to run.

End Time	Task instance only; system-supplied. The date and time the task instance completed.
Rows Retrieved	Task instance only; system-supplied. The number of rows retrieved by the SQL procedure.
SQL State	Task instance only; system-supplied. Resolves to a return code that indicates the outcome of the most recently executed SQL statement.
SQL Error Message	Task instance only; system-supplied. Any error messages returned by the SQL procedure.
User Estimated Duration	Task definition only; optional. The estimated amount of time it should normally take to run this task. The system uses this information to calculate the User Estimated End Time on a task instance record.
SQL Command	SQL command being executed against the database. Variables supported.
Result Processing	Specifies how Opswise Automation Center should determine whether the SQL command failed or completed successfully. Options: • Skip Result Processing • Success Exitcode Range - The SQL command is considered completed successfully if its exitcode falls within the range specified in the Exit Codes field (see below). • Failure Exitcode Range - The SQL command is considered failed if its exitcode falls within the range specified in the Exit Codes field (see below). • Success Result Set Contains - The SQL command is considered completed successfully depending on the value in a specific database column (see Column Name, Operator and Value fields below). • Failure Result Set Contains - The SQL command is considered failed depending on the value in a specific database column (see Column Name, Operator and Value fields below).
Exit Codes (Result Processing)	If Result Processing = Success Exitcode Range or Failure Exit Code Range, this field specifies the range. Format: Numeric. Use commas to list a series of exitcodes; use hyphens to specify a range. Example: 1,5, 22-30.
Column Name	If Result Processing = Success Result Set Contains or Failure Result Set Contains, this field specifies the name of a database column that is being checked for a specific value. Options: • Standard Output (STDOUT) • Standard Error (STDERR) • File
Operator	Operator being used for the comparison. Options: =, !=, >, >=, <, <=, regex.
Value	Value being compared, using the operator specified.
Auto Cleanup	When data is retrieved as the result of a SQL task, the data is written to a temporary table in the database. If Auto Cleanup is enabled, the data is discarded upon completion of the task. This feature is useful for testing your SQL task or in any other situation where you do not want a copy of the retrieved data.
Late Start	If enabled, and if the task instance starts after the time specified in the Late Start Time field (see below), the task instance is flagged as late. To determine whether a task instance started late, open the task instance and locate the Started Late field; the field is checked if the instance started late.

Late Start Type	Required if Late Start is enabled. Options are: • **Time** - Flag the task if it starts after the specified time. • **Duration** - Flag the task if it starts a certain amount of time after the programmed start time. The task must have a specific start time.
Late Start Time	Time after which the task start time is considered late. Use hh:mm, 24-hour time
Late Start Duration	Duration (amount of relative time) after which the task is considered to have started late. For a task within a workflow, the duration is the period between the time the workflow starts and the time the task itself starts. For example, a task might have a Late Start Duration of 60 minutes. If the workflow starts at 9:00 a.m. but the task itself does not start until 10:30, the task has started late. For a task that is not within a workflow, Late Start Duration has meaning only if the task has been held upon starting. For example, if a task has a Late Start Duration of 60 minutes and the Hold on Start field is enabled, if the task is not released from hold within the amount of time specified in the Late Start Duration field, the task has started late.
Started Late	Task instance only; system-supplied. This field is flagged if the task started later than the time specified in the Late Start fields.
Late Finish	If enabled, and if the task instance finishes after the time or period specified, the task instance is flagged as late. You can specify a time or duration to determine a late finish (see Late Finish Type below). To determine whether a task instance finished late, open the task instance and locate the Finished Late field; the field is checked if the instance finished after the specified time or lasted longer than expected. This field only appears on the task instance if the user specified a Late Finish in the task definition.
Late Finish Type	If Late Finish is enabled, use this field to specify whether you want to use time (see Late Finish Time below) or duration (see Late Finish Duration below) to determine whether the task instance finished late.
Late Finish Time	If Late Finish Type is Time, use this to specify the time after which the task finish time is considered late. Use hh:mm, 24-hour time.
Late Finish Duration	If Late Finish Type is Duration, use this to specify the longest amount of time this task instance should take to run. You can specify any combination of hours, minutes, and seconds.
Finished Late	Task instance only; system-supplied. This field is flagged if the task finished later than the time or duration specified in the Late Finish fields.
Early Finish	If enabled, and if the task instance finishes before the time or period specified, the task instance is flagged as early. You can specify a time or duration to determine an early finish (see Early Finish Type below). To determine whether a task instance finished early, open the task instance and locate the Finished Early field; the field is checked if the instance finished before the specified time or did not last as long as expected. This field only appears on the task instance if the user added Early Finish specifications to the task definition.
Early Finish Type	If Early Finish is enabled, use this field to specify whether you want to use time (see Early Finish Time below) or duration (see Early Finish Duration below) to determine whether the task instance finished early.
Early Finish Time	If Early Finish Type is Time, use this to specify the time before which the task finish time is considered early. That is, enter a time at which the task should still be running. Use hh:mm, 24-hour time.

Early Finish Duration	If Early Finish Type is Duration, use this to specify the shortest amount of time this task instance should take to run. You can specify any combination of hours, minutes, and seconds.
Finished Early	Task instance only; system-supplied. This field is flagged if the task finished earlier than the time specified in the Early Finish fields.
Maximum Retries	User-defined. The maximum number of times the system should retry this task after it has started and gone to a failed state.
Retry Indefinitely	User-defined. Enabled or disabled. Indicates whether the system should continue trying indefinitely to run this task. If you enable this field, it overrides any value placed in the Maximum Retries field (above).
Retry Interval	User-defined. The number of seconds between each retry.
Current Retry Count	Task instance only; system-supplied. Displays only for a running task instance. Displays the current number of times the system has retried the task after it first went to failure status.
User Estimated End Time	Task instance only; system-supplied. If the user entered information into the User Estimated Duration field in the task definition, the system uses this information to calculate an end time for the task instance, based on the date/time the task instance started.
Shortest Estimated End Time	Task instance only; system-supplied.
Average Estimated End Time	Task instance only; system-supplied.
Longest Estimated End Time	Task instance only; system-supplied.
First Time Ran	Task definition only; system-supplied. Displays after the first time the task runs. The date and time this task first ran.
Lowest Instance Time	Task definition only; system-supplied. Displays after the first time the task runs. The shortest amount of time this task has taken to run.
Last Time Ran	Task definition only; system-supplied. Displays after the first time the task runs. The most recent date and time the task ran.
Average Instance Time	Task definition only; system-supplied. Displays after the first time the task runs. Shows the average amount of time this task takes to run.
Number of Instances	Task definition only; system-supplied. Displays after the first time the task runs. Shows the number of times this task has run.
Highest Instance Time	Task definition only; system-supplied. Displays after the first time the task runs. The longest amount of time this task has taken to run.
Last Instance Duration	Task definition only; system-supplied. Displays after the first time the task runs. The amount of time the task took to run the last time it ran.

Submit button	Submits the new record to the database.
Update button	Saves updates to the record.
Launch Task button	Manually launches the task.
Delete button	Deletes the current record.
View Parent button	Task instance only; Displays this task's parent task (workflow), if any.
Show Details button	Task instance only; displays detailed information about this task instance.
Skip button	Task instance only; for tasks loaded into the schedule that haven't run yet. Allows you to tell the system to skip this task. See Skipping a Task.
Hold button	Task instance only; see Putting a Task on Hold.
Force Finish button	Task instance only; see Force Finishing a Task.
Re-run button	Task instance only; see Re-running a Task.
SQL Result Set tab	Task instance only. Stores results of executed SQL statements, if any.
SQL Warning Set tab	Task instance only. Contains warnings returned by executed SQL statements, if any.
Task Virtual Resources tab	Lists Virtual Resources to which this task is assigned. See Creating Virtual Resources.
Variables tab	Displays all variables associated with this record.
Actions tab	Allows you to specify actions that the system will take automatically based on events that occur during the execution of this task. Events are task instance status, exit codes, late start, late finish, and early finish. Actions are: • **Abort Action** - Abort the task if certain events occur. For details, see Setting Up Abort Actions. • **Email Notification** - Send an email if certain events occur. For details, see Creating Email Notifications. • **SNMP Notification** - Send an email if certain events occur. For details, see Creating SNMP Notifications. • **Set Variable** - Used in workflows to set a variable based on the occurrence of certain events. For details, see Setting Variables in a Workflow.
Task Instances tab	System-supplied. Displays a list of all task instances generated by this task definition.

Triggers tab	Displays a list of all triggers that have been defined to launch this task. Also allows you to add new triggers. If you add a new trigger from this location, the system automatically constructs a default trigger name as follows: <current task name>#TRIGGER#. You can change the default name if desired. For instructions on creating triggers, see Creating Triggers.
Notes tab	Displays all notes associated with this task. See Runbook Notes.
Versions Tab	Stores copies of all previous versions of the current record (see Record Versioning).

Specifying When a Task Runs

You can run the task as part of a workflow, specify triggers that run the task automatically based on times or events, or run the task manually.

Monitoring Task Execution

You can monitor all system activity from the Activity display and can view activity history from the Activity History screen.

Stored Procedure Task

Before You Begin

To run a Stored Procedure task, you first need to create a Database Connection, which defines the information needed to locate and access the database.

Overview

The Stored Procedure task allows you to execute a stored procedure against the database specified in the task.

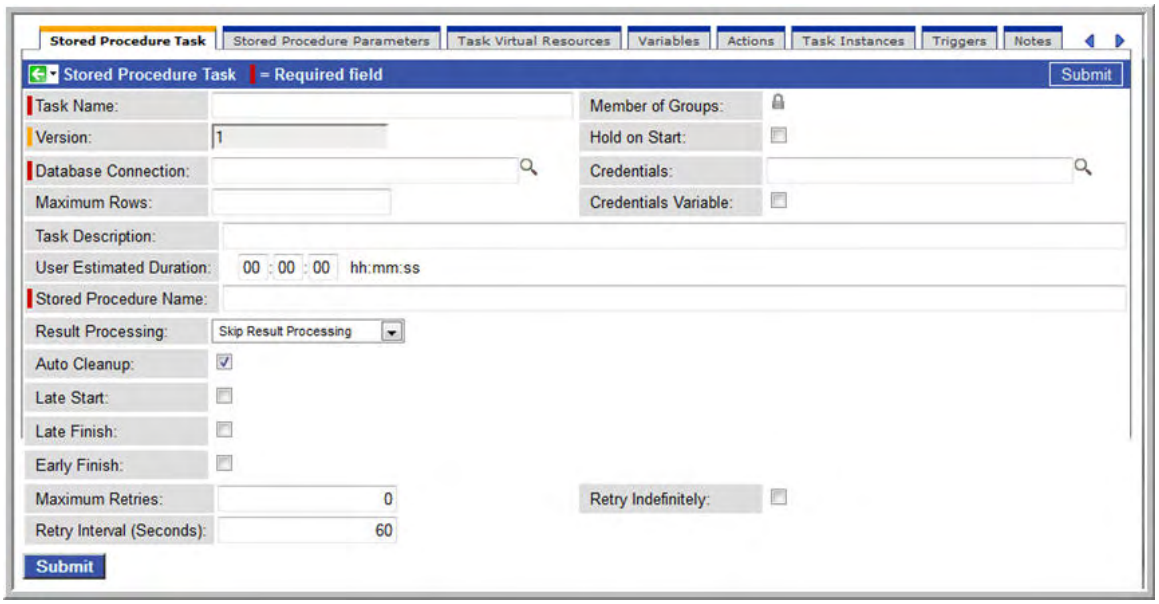
Built-In Variables

The built-in variables outlined below can be used in a Stored Procedure task to pass data where appropriate:

- Task and Task Instance Variables
- SQL and Stored Procedure Variables

Creating a New Stored Procedure Task

1. From the navigation pane, select **Automation Center > Tasks > Stored Procedure Tasks**. The Stored Procedure Tasks List screen displays.
2. Click **New**. The Stored Procedure Task Definition screen displays.



Task Name:		Member of Groups:	
Version:	1	Hold on Start:	□
Database Connection:		Credentials:	
Maximum Rows:		Credentials Variable:	□
Task Description:			
User Estimated Duration:	00 : 00 : 00 hh:mm:ss		
Stored Procedure Name:			
Result Processing:	Skip Result Processing		
Auto Cleanup:	☑		
Late Start:	□		
Late Finish:	□		
Early Finish:	□		
Maximum Retries:	0	Retry Indefinitely:	□
Retry Interval (Seconds):	60		

3. Using the field descriptions provided below as a guide, complete the fields as needed.
4. Click the **Submit** button to save the record and return to the menu, or right-click on the title bar and select **Save** to save the record and remain on the current display.
5. If appropriate, repeat these steps for any additional tasks you want to add.

Task run statistics appear after the first time this task has been launched.

Stored Procedure Task Field Descriptions

The table below describes the fields, buttons, and tabs on the task definition and task instance screens. Color coding is provided that differentiates the following three types of fields:

- Fields that display on the task definition and task instance screens are shown in black.
- Fields that display only on the task definition screen are shown in green.
- Fields that display only on the task instance screen are shown in maroon.

Field Name	Description
Task/Instance Name	Required. Name used within Opswise to identify this task. Up to 40 alphanumerics; variables supported. It is the user's responsibility to develop a workable naming scheme for tasks.
Invoked by	Task instance only; system-supplied. How the task instance was launched. One of the following: • Trigger: (Trigger Name) - The instance was launched by the named trigger. • Workflow: (Workflow Name) - The instance was launched by the named workflow. • Manually Launched - The instance was launched by a user. To determine the name of the user: 1. From the Activity or All Task Instances screen, click the task instance name to open the record. 2. The Execution User field identifies the user who launched the task instance.
Task	Task instance only; system-supplied. Hover over the paper icon to display more information about the task instance. Click the paper icon to display the task definition record.
Execution User	Task instance only; system-supplied. If the task was launched manually, the ID of the user who launched it.
Instance Reference Id	Task instance only; system-supplied. The system increments this number each time the task is run.
Member of Groups	User-defined. Allows you to select one or more Opswise groups that this record definition belongs to. Click on the lock icon to unlock the field and select groups.
Version	Task definition only; system-supplied. The version number of the current record, which is incremented by the system every time a user updates a record. Click on the Versions tab to view previous versions. For details, see Record Versioning.
Hold on Start	If enabled, when the task is launched it appears in the Activity display with a status of Held. The task runs when the user releases it.
Hold Reason	Information about why the task will be put on hold when it starts.
Database Connection field	Optional. The name of the Opswise Automation Center database connection that defines the database. Click the magnifying glass to browse for an existing database connection or add a new one.
Credentials	Login credentials Opswise Automation Center will use to access the remote machine. They must be provided either in the resource definition or task definition, or both. Credentials provided in the task definition override any credentials provided in the agent resource definition. See Credentials.

Maximum Rows	Optional. If necessary, specify a limit to the number of rows you want returned by the SQL statement. This value overrides any value you specify in the database connection.
Credentials Variable	Optional. If enabled, the Credentials field (see above) converts from a reference field (where you browse and select a record) into a text field that allows you to enter a variable. Use the format: ${variable name}. The variable must be a supported type as described in Using Variables and Functions .
Task Description	User-supplied description of this record.
Status	Task instance only; system-supplied. See Task Statuses.
Exit Code	Task instance only; system-supplied. The vendor-specific exception code for the SQL exception.
Status Description	Task instance only; system-supplied. Provides additional information, if any, about the status of the task.
Start Time	Task instance only; system-supplied. The date and time the task started.
Duration	Task instance only; system-supplied. The amount of time the task took to run.
End Time	Task instance only; system-supplied. The date and time the task instance completed.
Rows Retrieved	Task instance only; system-supplied. The number of rows retrieved by the SQL procedure.
SQL State	Task instance only; system-supplied. Resolves to a return code that indicates the outcome of the most recently executed SQL statement.
SQL Error Message	Task instance only; system-supplied. Any error messages returned by the SQL procedure.
User Estimated Duration	Task definition only; optional. The estimated amount of time it should normally take to run this task. The system uses this information to calculate the User Estimated End Time on a task instance record.
Stored Procedure Name	Name of the file containing the stored procedure being executed against the database. Variables supported.

Result Processing	Specifies how Opswise should determine whether the Stored Procedure failed or completed successfully. Options: • Skip Result Processing • Success Exitcode Range - The Stored Procedure is considered completed successfully if its exitcode falls within the range specified in the Exit Codes field (see below). • Failure Exitcode Range - The Stored Procedure is considered failed if its exitcode falls within the range specified in the Exit Codes field (see below). • Success Result Set Contains - The Stored Procedure is considered completed successfully depending on the value in a specific database column (see Column Name, Operator and Value fields below). • Failure Result Set Contains - The Stored Procedure is considered failed depending on the value in a specific database column (see Column Name, Operator and Value fields below). • Success Output Parameter - The Stored Procedure is considered completed successfully if its output parameter satisfies the condition specified in the associated Parameter Position, Operator, and Value fields. • Failure Output Parameter - The Stored Procedure is considered failed if its output parameter satisfies the condition specified in the associated Parameter Position, Operator, and Value fields.
Parameter Position	Position of this parameter within a list of parameters.
Operator	Operator being used for the comparison. Options: =, !=, >, >=, <, <=, regex.
Value	Value being compared, using the operator specified.
Exit Codes (Result Processing)	If Result Processing = Success Exitcode Range or Failure Exit Code Range, this field specifies the range. Format: Numeric. Use commas to list a series of exitcodes; use hyphens to specify a range. Example: 1,5, 22-30.
Column Name	If Result Processing = Success Result Set Contains or Failure Result Set Contains, this field specifies the name of a database column that is being checked for a specific value. Options: • Standard Output (STDOUT) • Standard Error (STDERR) • File
Operator	Operator being used for the comparison. Options: =, !=, >, >=, <, <=, regex.
Value	Value being compared, using the operator specified.
Auto Cleanup	When data is retrieved as the result of a SQL task, the data is written to a temporary table in the database. If Auto Cleanup is enabled, the data is discarded upon completion of the task. This feature is useful for testing your SQL task or in any other situation where you do not want a copy of the retrieved data.
Late Start	If enabled, and if the task instance starts after the time specified in the Late Start Time field (see below), the task instance is flagged as late. To determine whether a task instance started late, open the task instance and locate the Started Late field; the field is checked if the instance started late.
Late Start Type	Required if Late Start is enabled. Options are: • **Time** - Flag the task if it starts after the specified time. • **Duration** - Flag the task if it starts a certain amount of time after the programmed start time. The task must have a specific start time.
Late Start Time	Time after which the task start time is considered late. Use hh:mm, 24-hour time

Late Start Duration	Duration (amount of relative time) after which the task is considered to have started late. For a task within a workflow, the duration is the period between the time the workflow starts and the time the task itself starts. For example, a task might have a Late Start Duration of 60 minutes. If the workflow starts at 9:00 a.m. but the task itself does not start until 10:30, the task has started late. For a task that is not within a workflow, Late Start Duration has meaning only if the task has been held upon starting. For example, if a task has a Late Start Duration of 60 minutes and the Hold on Start field is enabled, if the task is not released from hold within the amount of time specified in the Late Start Duration field, the task has started late.
Started Late	Task instance only; system-supplied. This field is flagged if the task started later than the time specified in the Late Start fields.
Late Finish	If enabled, and if the task instance finishes after the time or period specified, the task instance is flagged as late. You can specify a time or duration to determine a late finish (see Late Finish Type below). To determine whether a task instance finished late, open the task instance and locate the Finished Late field; the field is checked if the instance finished after the specified time or lasted longer than expected. This field only appears on the task instance if the user specified a Late Finish in the task definition.
Late Finish Type	If Late Finish is enabled, use this field to specify whether you want to use time (see Late Finish Time below) or duration (see Late Finish Duration below) to determine whether the task instance finished late.
Late Finish Time	If Late Finish Type is Time, use this to specify the time after which the task finish time is considered late. Use hh:mm, 24-hour time.
Late Finish Duration	If Late Finish Type is Duration, use this to specify the longest amount of time this task instance should take to run. You can specify any combination of hours, minutes, and seconds.
Finished Late	Task instance only; system-supplied. This field is flagged if the task finished later than the time or duration specified in the Late Finish fields.
Early Finish	If enabled, and if the task instance finishes before the time or period specified, the task instance is flagged as early. You can specify a time or duration to determine an early finish (see Early Finish Type below). To determine whether a task instance finished early, open the task instance and locate the Finished Early field; the field is checked if the instance finished before the specified time or did not last as long as expected. This field only appears on the task instance if the user added Early Finish specifications to the task definition.
Early Finish Type	If Early Finish is enabled, use this field to specify whether you want to use time (see Early Finish Time below) or duration (see Early Finish Duration below) to determine whether the task instance finished early.
Early Finish Time	If Early Finish Type is Time, use this to specify the time before which the task finish time is considered early. That is, enter a time at which the task should still be running. Use hh:mm, 24-hour time.
Early Finish Duration	If Early Finish Type is Duration, use this to specify the shortest amount of time this task instance should take to run. You can specify any combination of hours, minutes, and seconds.
Finished Early	Task instance only; system-supplied. This field is flagged if the task finished earlier than the time specified in the Early Finish fields.
Maximum Retries	User-defined. The maximum number of times the system should retry this task after it has started and gone to a failed state.

Retry Indefinitely	User-defined. Enabled or disabled. Indicates whether the system should continue trying indefinitely to run this task. If you enable this field, it overrides any value placed in the Maximum Retries field (above).
Retry Interval	User-defined. The number of seconds between each retry.
Current Retry Count	Task instance only; system-supplied. Displays only for a running task instance. Displays the current number of times the system has retried the task after it first went to failure status.
User Estimated End Time	Task instance only; system-supplied. If the user entered information into the User Estimated Duration field in the task definition, the system uses this information to calculate an end time for the task instance, based on the date/time the task instance started.
Shortest Estimated End Time	Task instance only; system-supplied.
Average Estimated End Time	Task instance only; system-supplied.
Longest Estimated End Time	Task instance only; system-supplied.
First Time Ran	Task definition only; system-supplied. Displays after the first time the task runs. The date and time this task first ran.
Lowest Instance Time	Task definition only; system-supplied. Displays after the first time the task runs. The shortest amount of time this task has taken to run.
Last Time Ran	Task definition only; system-supplied. Displays after the first time the task runs. The most recent date and time the task ran.
Average Instance Time	Task definition only; system-supplied. Displays after the first time the task runs. Shows the average amount of time this task takes to run.
Number of Instances	Task definition only; system-supplied. Displays after the first time the task runs. Shows the number of times this task has run.
Highest Instance Time	Task definition only; system-supplied. Displays after the first time the task runs. The longest amount of time this task has taken to run.
Last Instance Duration	Task definition only; system-supplied. Displays after the first time the task runs. The amount of time the task took to run the last time it ran.
Submit button	Submits the new record to the database.
Update button	Saves updates to the record.
Launch Task button	Manually launches the task.

Delete button	Deletes the current record.
View Parent button	Task instance only; Displays this task's parent task (workflow), if any.
Show Details button	Task instance only; displays detailed information about this task instance.
Skip button	Task instance only; for tasks loaded into the schedule that haven't run yet. Allows you to tell the system to skip this task. See Skipping a Task.
Hold button	Task instance only; see Putting a Task on Hold.
Force Finish button	Task instance only; see Force Finishing a Task.
Re-run button	Task instance only; see Re-running a Task.
Stored Procedure Parameters tab	See "Adding Stored Procedure Parameters" below.
SQL Result Set tab	Task instance only. Stores results of executed SQL statements, if any.
SQL Warning Set tab	Task instance only. Contains warnings returned by executed SQL statements, if any.
Task Virtual Resources tab	Lists Virtual Resources to which this task is assigned. See Creating Virtual Resources.
Variables tab	Displays all variables associated with this record.
Actions tab	Allows you to specify actions that the system will take automatically based on events that occur during the execution of this task. Events are task instance status, exit codes, late start, late finish, and early finish. Actions are: • **Abort Action** - Abort the task if certain events occur. For details, see Setting Up Abort Actions. • **Email Notification** - Send an email if certain events occur. For details, see Creating Email Notifications. • **SNMP Notification** - Send an email if certain events occur. For details, see Creating SNMP Notifications. • **Set Variable** - Used in workflows to set a variable based on the occurrence of certain events. For details, see Setting Variables in a Workflow.
Task Instances tab	System-supplied. Displays a list of all task instances generated by this task definition.
Triggers tab	Displays a list of all triggers that have been defined to launch this task. Also allows you to add new triggers. If you add a new trigger from this location, the system automatically constructs a default trigger name as follows: <current task name>#TRIGGER#. You can change the default name if desired. For instructions on creating triggers, see Creating Triggers.
Notes tab	Displays all notes associated with this task. See Runbook Notes.

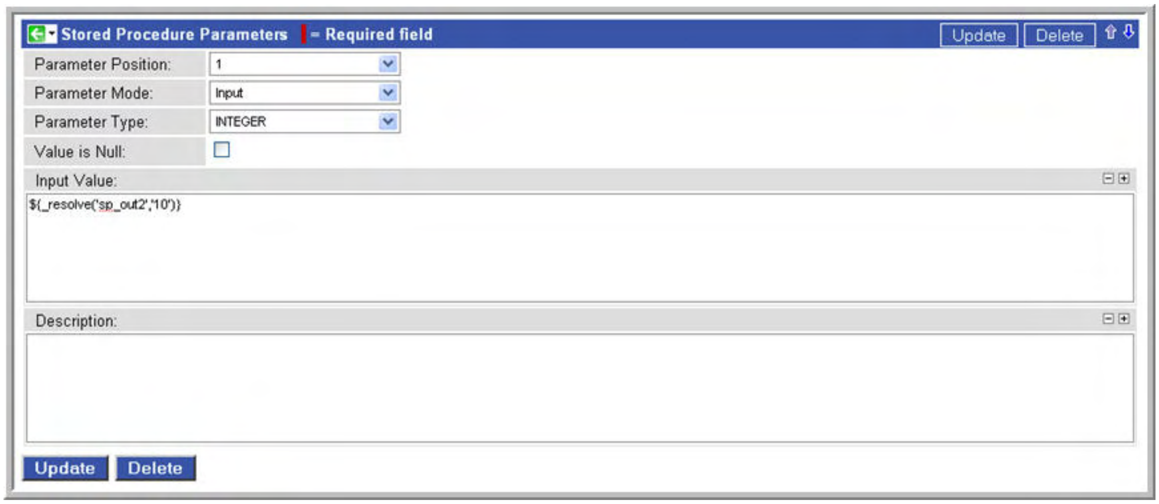
Versions Tab	Stores copies of all previous versions of the current record (see Record Versioning).

Adding Stored Procedure Parameters

You can enter one or more parameters for each stored procedure, as described below.

Adding a Parameter

1. Open the Stored Procedure task to which you want to add the parameter.
2. Click on the **Stored Procedure Parameters** tab or scroll to the parameters section. The system displays a list of parameters, if any.
3. Click the **New** button.
4. Use the field descriptions provided below to fill in the screen. An example parameter entry is shown below:



Parameter Position: 1

Parameter Mode: Input

Parameter Type: INTEGER

Value is Null: □

Input Value:

```
${_resolve('sp_out2','10')}
```

Description:

5. Click the **Submit** button.

Stored Procedure Parameter Field Descriptions

Field Name	Description
Parameter Position	The position of this parameter within a list of parameters.
Parameter Mode	Mode of this parameter. Options: • Input • Output • Input/Output

Parameter Type	Type of parameter. Options: - VARCHAR - SMALLINT - INTEGER - BIGINT - FLOAT - REAL - DOUBLE - NUMERIC - DECIMAL - DATE - TIME - TIMESTAMP - VARBINARY - BOOLEAN
Value is Null	The value for the parameter is a database NULL value; applies to the input part of a stored procedure parameter. That is, if a value in a database is undefined, it is NULL, which means it has no set value. An input value can be NULL and is represented by selecting "Value is Null".
Input Value	Input value of the parameter, if any.
Description	Description of this parameter.

Deleting a Parameter

Display the parameter you want to delete and click the **Delete** button. Or, you can delete one or more parameters as follows:

1. From the parameters list, click on the box associated with the parameter or parameters you want to delete.
2. From the **Actions on selected rows** menu, select **Delete**.

Viewing a Parameter

1. From the parameters list, scroll to the parameter you want to read.
2. Click the underlined field displayed in the leftmost column. The system displays the contents of the parameter.

Specifying When a Task Runs

You can run the task as part of a workflow, specify triggers that run the task automatically based on times or events, or run the task manually.

Monitoring Task Execution

You can monitor all system activity from the Activity display and can view activity history from the Activity History screen.

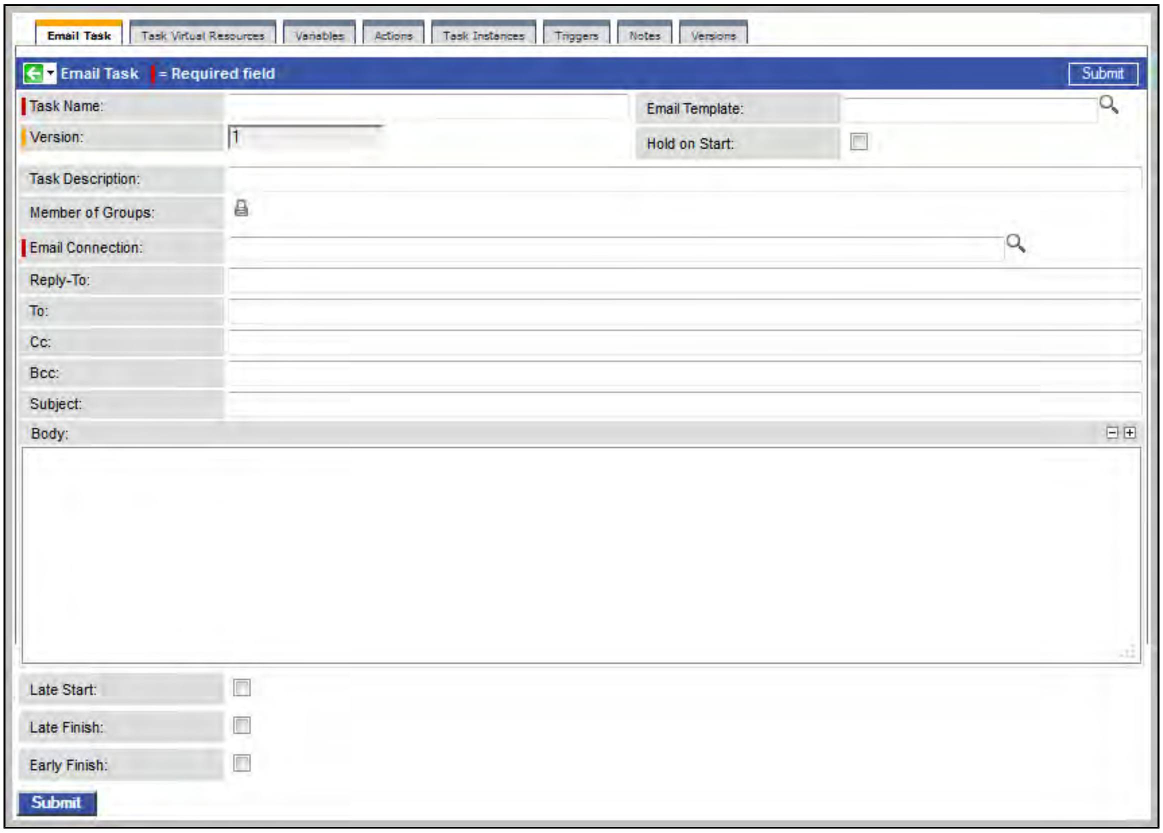
Email Task

Overview

The Email task allows you to create and send emails. In order to execute Email tasks, you first need to define an Email Connection, which defines the server information needed to create and send emails.

Creating a New Email Task

1. From the navigation pane, select **Automation Center > Tasks > Email Tasks**. The Email Tasks List screen displays.
2. Click **New**. The Email Task Definition screen displays.



@nbsp;

3. Using the field descriptions provided below as a guide, complete the fields as needed.
4. Click the **Submit** button to save the record and return to the menu, or right-click on the title bar and select **Save** to save the record and remain on the current display.
5. If appropriate, repeat these steps for any additional tasks you want to add.

Task run statistics show at the bottom appear after the first time this task has been launched.

Email Task Field Descriptions

The table below describes the fields, buttons, and tabs on the task definition and task instance screens. Color coding is provided that differentiates the following three types of fields:

- Fields that display on the task definition and task instance screens are shown in black.
- Fields that display only on the task definition screen are shown in green.
- Fields that display only on the task instance screen are shown in maroon.

Field Name	Description
Task/Instance Name	Required. Name used within Opswise to identify this task. Up to 40 alphanumerics; variables supported. It is the user's responsibility to develop a workable naming scheme for tasks.
Email Template	Optional. The name of the Email template defined using the Email template screen. The Email template allows you to specify standard recipients and text for outgoing emails. Type in a name, or click the magnifying glass to browse to an existing Email template or create a new one. You must specify either an Email template or Email connection, or both. If you specify both, the Email server specified in the Email Connection record overrides the server in the template.
Version	Task definition only; system-supplied. The version number of the current record, which is incremented by the system every time a user updates a record. Click on the Versions tab to view previous versions. For details, see Record Versioning.
Invoked by	Task instance only; system-supplied. How the task instance was launched. One of the following: • Trigger: (Trigger Name) - The instance was launched by the named trigger. • Workflow: (Workflow Name) - The instance was launched by the named workflow. • Manually Launched - The instance was launched by a user. To determine the name of the user: 1. From the Activity or All Task Instances screen, click the task instance name to open the record. 2. The Execution User field identifies the user who launched the task instance.
Task	Task instance only; system-supplied. Hover over the paper icon to display more information about the task instance. Click the paper icon to display the task definition record.
Instance Reference Id	Task instance only; system-supplied. The system increments this number each time the task is run.
Description	User-supplied description of this record.
Status	Task instance only; system-supplied. See Task Statuses.
Exit Code	Task instance only; system-supplied. The exit code captured by the agent when executing the task (for example, a command or script).
Status Description	Task instance only; system-supplied. Provides additional information, if any, about the status of the task.
Start Time	Task instance only; system-supplied. The date and time the task started.
Duration	Task instance only; system-supplied. The amount of time the task took to run.
End Time	Task instance only; system-supplied. The date and time the task instance completed.
Hold on Start	If enabled, when the task is launched it appears in the Activity display with a status of Held. The task runs when the user releases it.

Hold Reason	Information about why the task will be put on hold when it starts.
Member of Groups	User-defined. Allows you to select one or more Opswise groups that this record definition belongs to. Click on the lock icon to unlock the field and select groups.
Email Connection	Required. The name of the Email connection defined using the Email connection screen. The email connection specifies information about the email server. You can also specify the Email connection in the Email template (see above). You must specify either an Email template or Email connection, or both. If you specify an Email template and an Email connection, the server selected in the Email connection overrides the server selected in the Email template. Type in a name, or click the magnifying glass to browse to an existing Email server definition or create a new one.
Reply-To	Required. Specifies the email address of the sender. Use commas to separate multiple recipients. Variables supported.
To-Email Task	Required. Specifies the email address of the recipient. Use commas to separate multiple recipients. Variables supported.
CC-Email Task	Optional. Specifies the email address of the party being sent a copy of the email, if any. Use commas to separate multiple recipients. Variables supported.
BCC-Email Task	Optional. Specifies the email address of the party being sent a blind (hidden) copy of the email, if any. Use commas to separate multiple recipients. Variables supported.
Subject-Email	Optional. Specifies the subject line of the email. Variables supported.
Body-Email	Optional. Contains the text of the email message. Variables supported. If both the email template and the email task contain text in the body, the text is appended.
Late Start	If enabled, and if the task instance starts after the time specified in the Late Start Time field (see below), the task instance is flagged as late. To determine whether a task instance started late, open the task instance and locate the Started Late field; the field is checked if the instance started late.
Late Start Type	Required if Late Start is enabled. Options are: • **Time** - Flag the task if it starts after the specified time. • **Duration** - Flag the task if it starts a certain amount of time after the programmed start time. The task must have a specific start time.
Late Start Time	Time after which the task start time is considered late. Use hh:mm, 24-hour time
Late Start Duration	Duration (amount of relative time) after which the task is considered to have started late. For a task within a workflow, the duration is the period between the time the workflow starts and the time the task itself starts. For example, a task might have a Late Start Duration of 60 minutes. If the workflow starts at 9:00 a.m. but the task itself does not start until 10:30, the task has started late. For a task that is not within a workflow, Late Start Duration has meaning only if the task has been held upon starting. For example, if a task has a Late Start Duration of 60 minutes and the Hold on Start field is enabled, if the task is not released from hold within the amount of time specified in the Late Start Duration field, the task has started late.
Started Late	Task instance only; system-supplied. This field is flagged if the task started later than the time specified in the Late Start fields.

Late Finish	If enabled, and if the task instance finishes after the time or period specified, the task instance is flagged as late. You can specify a time or duration to determine a late finish (see Late Finish Type below). To determine whether a task instance finished late, open the task instance and locate the Finished Late field; the field is checked if the instance finished after the specified time or lasted longer than expected. This field only appears on the task instance if the user specified a Late Finish in the task definition.
Late Finish Type	If Late Finish is enabled, use this field to specify whether you want to use time (see Late Finish Time below) or duration (see Late Finish Duration below) to determine whether the task instance finished late.
Late Finish Time	If Late Finish Type is Time, use this to specify the time after which the task finish time is considered late. Use hh:mm, 24-hour time.
Late Finish Duration	If Late Finish Type is Duration, use this to specify the longest amount of time this task instance should take to run. You can specify any combination of hours, minutes, and seconds.
Finished Late	Task instance only; system-supplied. This field is flagged if the task finished later than the time or duration specified in the Late Finish fields.
Early Finish	If enabled, and if the task instance finishes before the time or period specified, the task instance is flagged as early. You can specify a time or duration to determine an early finish (see Early Finish Type below). To determine whether a task instance finished early, open the task instance and locate the Finished Early field; the field is checked if the instance finished before the specified time or did not last as long as expected. This field only appears on the task instance if the user added Early Finish specifications to the task definition.
Early Finish Type	If Early Finish is enabled, use this field to specify whether you want to use time (see Early Finish Time below) or duration (see Early Finish Duration below) to determine whether the task instance finished early.
Early Finish Time	If Early Finish Type is Time, use this to specify the time before which the task finish time is considered early. That is, enter a time at which the task should still be running. Use hh:mm, 24-hour time.
Early Finish Duration	If Early Finish Type is Duration, use this to specify the shortest amount of time this task instance should take to run. You can specify any combination of hours, minutes, and seconds.
Finished Early	Task instance only; system-supplied. This field is flagged if the task finished earlier than the time specified in the Early Finish fields.
First Time Ran	Task definition only; system-supplied. Displays after the first time the task runs. The date and time this task first ran.
Lowest Instance Time	Task definition only; system-supplied. Displays after the first time the task runs. The shortest amount of time this task has taken to run.
Last Time Ran	Task definition only; system-supplied. Displays after the first time the task runs. The most recent date and time the task ran.
Average Instance Time	Task definition only; system-supplied. Displays after the first time the task runs. Shows the average amount of time this task takes to run.

Number of Instances	Task definition only; system-supplied. Displays after the first time the task runs. Shows the number of times this task has run.
Highest Instance Time	Task definition only; system-supplied. Displays after the first time the task runs. The longest amount of time this task has taken to run.
Last Instance Duration	Task definition only; system-supplied. Displays after the first time the task runs. The amount of time the task took to run the last time it ran.
User Estimated End Time	Task instance only; system-supplied. If the user entered information into the User Estimated Duration field in the task definition, the system uses this information to calculate an end time for the task instance, based on the date/time the task instance started.
Shortest Estimated End Time	Task instance only; system-supplied.
Average Estimated End Time	Task instance only; system-supplied.
Longest Estimated End Time	Task instance only; system-supplied.
Submit button	Submits the new record to the database.
Update button	Saves updates to the record.
Launch Task button	Manually launches the task.
Delete button	Deletes the current record.
View Parent button	Task instance only; Displays this task's parent task (workflow), if any.
Show Details button	Task instance only; displays detailed information about this task instance.
Skip button	Task instance only; for tasks loaded into the schedule that haven't run yet. Allows you to tell the system to skip this task. See Skipping a Task.
Hold button	Task instance only; see Putting a Task on Hold.
Force Finish button	Task instance only; see Force Finishing a Task.
Re-run button	Task instance only; see Re-running a Task.
Output tab	Task instance only. Displays output generated from the process, if any, based on specifications provided by the user in the Automatic Output Retrieval fields in the task definition.

Task Virtual Resources tab	Lists Virtual Resources to which this task is assigned. See Creating Virtual Resources.
Variables tab	Displays all variables associated with this record.
Actions tab	Allows you to specify actions that the system will take automatically based on events that occur during the execution of this task. Events are task instance status, exit codes, late start, late finish, and early finish. Actions are: • **Abort Action** - Abort the task if certain events occur. For details, see Setting Up Abort Actions. • **Email Notification** - Send an email if certain events occur. For details, see Creating Email Notifications. • **SNMP Notification** - Send an email if certain events occur. For details, see Creating SNMP Notifications. • **Set Variable** - Used in workflows to set a variable based on the occurrence of certain events. For details, see Setting Variables in a Workflow.
Task Instances tab	System-supplied. Displays a list of all task instances generated by this task definition.
Triggers tab	Displays a list of all triggers that have been defined to launch this task. Also allows you to add new triggers. If you add a new trigger from this location, the system automatically constructs a default trigger name as follows: <current task name>#TRIGGER#. You can change the default name if desired. For instructions on creating triggers, see Creating Triggers.
Notes tab	Displays all notes associated with this task. See Runbook Notes.
Versions Tab	Stores copies of all previous versions of the current record (see Record Versioning).

Specifying When a Task Runs

You can run the task as part of a workflow, specify triggers that run the task automatically based on times or events, or run the task manually.

Monitoring Task Execution

You can monitor all system activity from the Activity display and can view activity history from the Activity History screen.

Task Monitor Task

Overview

The Task Monitor task monitors another task or tasks for one or more specific statuses. When setting up a Task Monitor task, you can monitor all tasks; a specific task; a task type, such as Windows tasks; or a group of tasks based on the name, such as all tasks whose name contains the string "DEV". You can also monitor for any combination of tasks statuses. For example, you can monitor for all tasks with a status of RESOURCE WAIT or UNDELIVERABLE, all Windows tasks in a FAILED status, or all tasks whose name contains "REPORT" that have a status of SUCCESS. For Task Monitors within a workflow, you can also specify a Time Scope, or window of time, during which the event being monitored for must be satisfied.

Built-In Variables

The built-in variables outlined below can be used to pass data where appropriate:

- Task and Task Instance Variables
- Task Monitor Variables

Processing Flow for Task Monitors

The processing on a Task Monitor may differ depending on which of the following methods was used to launch it:

- Launched by a workflow
- Launched by a Task Monitor trigger
- Launched manually or by another trigger

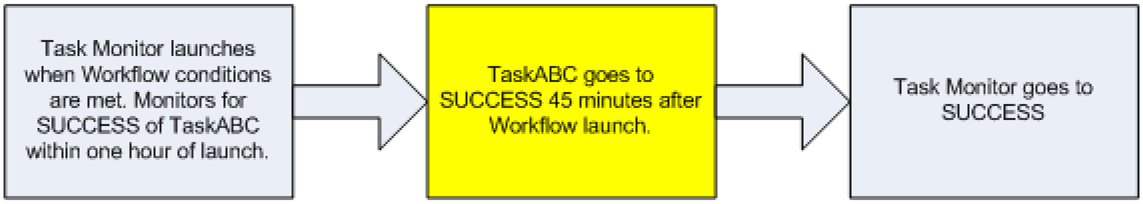
Each method is described in detail below.

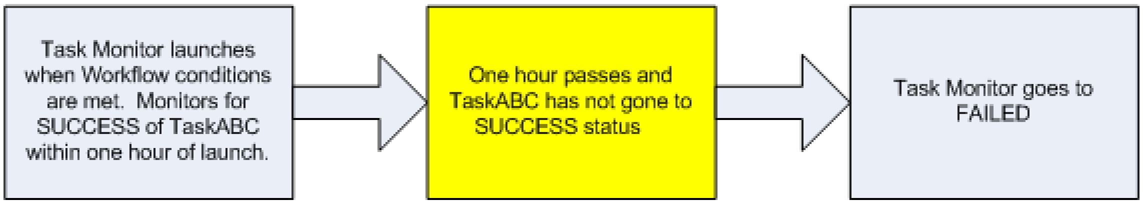
Launching a Task Monitor Task Within a Workflow

Within a workflow, the Task Monitor task launches like any other task in the workflow, that is, whenever the workflow conditions warrant it. The Task Monitor runs until one of the conditions described below occurs:

- When the conditions specified in the Task Monitor are met, the Task Monitor goes to a status of SUCCESS.
- When the time window specified in the Task Monitor passes and the conditions in the Task Monitor are not met, the Task Monitor goes to a status of FAILED. If the time window is entirely in the past and the system does not locate the required event in the Activity table when the Task Monitor launches, the Task Monitor goes immediately to a FAILED status.
- If no time window is specified in the Task Monitor and the Task Monitor conditions are not met, the Task Monitor task continues running.
- A user can manually force finish the Task Monitor task.

The following diagram illustrates how a Task Monitor might go to SUCCESS and FAILED status within a workflow.

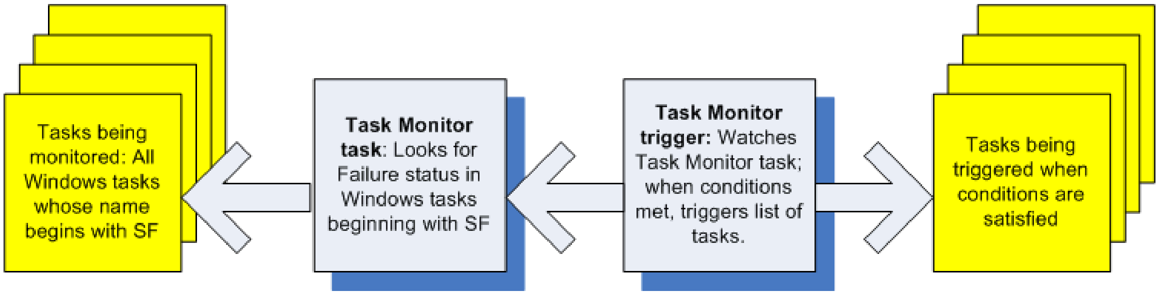

Task Monitor launches when Workflow conditions are met. Monitors for SUCCESS of TaskABC within one hour of launch.
TaskABC goes to SUCCESS 45 minutes after Workflow launch.
Task Monitor goes to SUCCESS


Task Monitor launches when Workflow conditions are met. Monitors for SUCCESS of TaskABC within one hour of launch.
One hour passes and TaskABC has not gone to SUCCESS status
Task Monitor goes to FAILED

Launching a Task Monitor Task Using a Task Monitor Trigger

The Task Monitor task launches when the user enables the Task Monitor trigger. Each time the conditions in the Task Monitor task are satisfied, the tasks specified in the trigger are launched. This process continues until a user disables the associated Task Monitor trigger.

The following diagram shows an example of how you might set up a task monitoring scheme using the Task Monitor task and Task Monitor trigger.


Tasks being monitored: All Windows tasks whose name begins with SF
Task Monitor task: Looks for Failure status in Windows tasks beginning with SF
Task Monitor trigger: Watches Task Monitor task; when conditions met, triggers list of tasks.
Tasks being triggered when conditions are satisfied

Launching a Task Monitor Task Manually or Via Other Trigger

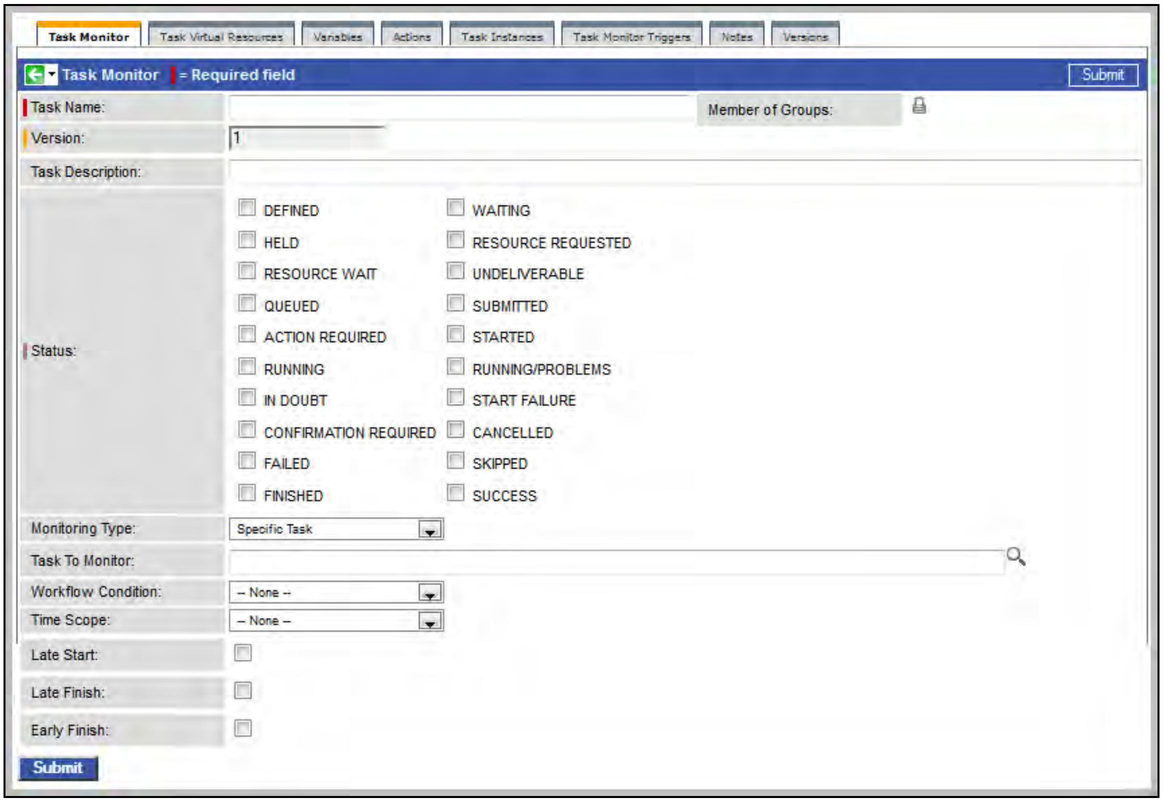
If you manually launch a Task Monitor task or launch it using a trigger other than a Task Monitor trigger, such as a Time trigger, the task continues running until its specified conditions are met. When that occurs, the Task Monitor goes to SUCCESS. No other processing occurs unless you have configured notifications with the task or set up some other task(s) to launch based on the status of this task.

The Task Monitor runs until one of the conditions described below occurs:

- When the time window specified in the Task Monitor passes and the conditions in the Task Monitor are not met, the Task Monitor goes to a status of FAILED. If the time window is entirely in the past and the system does not locate the required event in the Activity table when the Task Monitor launches, the Task Monitor goes immediately to a FAILED status.
- If no time window is specified in the Task Monitor and the Task Monitor conditions are not met, the Task Monitor task continues running.

Creating a New Task Monitor Task

1. From the navigation pane, select **Automation Center > Tasks > Task Monitors**. The Task Monitors List screen displays.
2. Click **New**. The Task Monitor Definition screen displays.



Task Name:

Member of Groups:

Version: 1

Task Description:

Status:

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

Monitoring Type: Specific Task

Task To Monitor:

Workflow Condition: -- None --

Time Scope: -- None --

Late Start: □

Late Finish: □

Early Finish: □

3. Using the field descriptions provided below as a guide, complete the fields as needed.
4. Click the **Submit** button to save the record and return to the menu, or right-click on the title bar and select **Save** to save the record and remain on the current display.
5. If appropriate, repeat these steps for any additional tasks you want to add.

Task run statistics appear after the first time this task has been launched.

Task Monitor Field Descriptions

The table below describes the fields, buttons, and tabs on the task definition and task instance screens. Color coding is provided that differentiates the following three types of fields:

- Fields that display on the task definition and task instance screens are shown in black.
- Fields that display only on the task definition screen are shown in green.
- Fields that display only on the task instance screen are shown in maroon.

Field Name	Description
Task/Instance Name	Required. Name used within Opswise to identify this task. Up to 40 alphanumerics; variables supported. It is the user's responsibility to develop a workable naming scheme for tasks.
Invoked by	Task instance only; system-supplied. How the task instance was launched. One of the following: • Trigger: (Trigger Name) - The instance was launched by the named trigger. • Workflow: (Workflow Name) - The instance was launched by the named workflow. • Manually Launched - The instance was launched by a user. To determine the name of the user: 1. From the Activity or All Task Instances screen, click the task instance name to open the record. 2. The Execution User field identifies the user who launched the task instance.
Task	Task instance only; system-supplied. Hover over the paper icon to display more information about the task instance. Click the paper icon to display the task definition record.
Instance Reference Id	Task instance only; system-supplied. The system increments this number each time the task is run.

Member of Groups	User-defined. Allows you to select one or more Opswise groups that this record definition belongs to. Click on the lock icon to unlock the field and select groups.
Version	Task definition only; system-supplied. The version number of the current record, which is incremented by the system every time a user updates a record. Click on the Versions tab to view previous versions. For details, see Record Versioning.
Description	User-supplied description of this record.
Status	Status being monitored for. When the task being monitored goes to a status specified in this field, the associated trigger is satisfied and the tasks specified in the trigger launch. You can specify status only, or status and exit code. You can specify as many statuses as needed (see Task Statuses).
Monitoring Type	Required. Specifies which task or tasks are being monitored. Options: • Specific Task - One or more specific tasks are being monitored. Use the Task to Monitor field to specify the task names. • General Tasks - Allows you to specify selection parameters that determine which task or tasks to be monitored. Use the Task Name To Monitor Condition and Task Type To Monitor field to create your selection parameters.
Task to Monitor	If Monitoring Type = Specific Task, this field specifies one or more tasks to monitor. Type in a task name or click the magnifying glass to browse for an existing task or to add a new task. To display details about the task specified in this field, hover over the paper icon.
Task Name to Monitor Condition	If Monitoring Type = General Task(s), this field allows you to specify selection parameters for which tasks to monitor. Options: • ALL TASKS - Specifies that the Task Monitor should monitor all tasks. • Starts With - Allows you to specify a string that one or more task names start with. Use the Task Name To Monitor Value field to enter the string. • Contains - Allows you to specify a string that one or more task names contain. Use the Task Name To Monitor Value field to enter the string. • Ends With - Allows you to specify a string that one or more task names end with. Use the Task Name To Monitor Value field to enter the string.
Task Name to Monitor Value	If the Task Name To Monitor Condition field = Starts With, Contains, or Ends With, use this field to specify the selection string. Up to 40 alphanumerics.
Task Type to Monitor	If you selected a Monitoring Type of General Tasks, this field allows you to define specific task types to monitor for. For example, to monitor all SQL tasks, you would select Monitoring Type = General Tasks, then select Task Type to Monitor = SQL Tasks.
Workflow Condition	With Workflow Condition Value, below, allows you to identify a workflow or workflows that contain the task being monitored for. If you specify these parameters, the task monitor conditions will only be considered met if the task appears within the specified workflow. Use this field to specify the condition. Options: • None • Equals • Starts With • Contains • Ends With
Workflow Condition Value	With Workflow Condition, above, allows you to identify a workflow or workflows that contain the task being monitored for. If you specify these parameters, the task monitor conditions will only be considered met if the task appears within the specified workflow. Use this field to specify the name or partial name of the workflow.

Time Scope	Used for Task Monitor tasks not associated with a Trigger. The Time Scope fields are used to create a window during which the Task Monitor conditions must be met in order for the Task Monitor be satisfied. The Time Scope window is always relative to the time that the Task Monitor launched. For example, if you put -01:00 in the **From** time field and 02:00 in the **To** time field, the window's begin time is one hour before the Task Monitor launched and its end time is two hours after it launched. Note that the task being monitored must still be in the Activity screen in order for you to monitor for events that occurred in the past. If you specify a window that begins in the past, when the Task Monitor launches, it searches through the Activity table for the specified event. If it locates the event, the Task Monitor is satisfied immediately.
Time Scope From	Used for Task Monitor tasks not associated with a Trigger. This field, together with the **Time Scope To** field, allows you to specify a window of time, relative to the time the Task Monitor task launched, during which the conditions of the Task Monitor must be met. If the conditions are not met within the specified window, the Task Monitor task goes to a FAILED status. If you specify a past time in the **Time Scope From** field, as soon as the Task Monitor task launches, the system searches the Activity table for past events that match the specified conditions. If the conditions are satisfied already, the Task Monitor task goes immediately to SUCCESS status. Otherwise, the system continues monitoring until the conditions are met or until the **Time Scope To** time has passed.
Time Scope To	Used for Task Monitor tasks not associated with a Trigger. This field, together with the **Time Scope From** field, allows you to specify a window of time, relative to the time the Task Monitor task launched, during which the conditions of the Task Monitor must be met. If the conditions are not met within the specified window, the Task Monitor task goes to a FAILED status. If the conditions in the Task Monitor task are met before the **Time Scope To** time arrives, the Task Monitor task goes to SUCCESS. If the conditions are not met by the **Time Scope To** time, the Task Monitor task goes to FAILED status.
Status	Task instance only; system-supplied. Status of this task instance (see Task Statuses).
Status Description	Task instance only; system-supplied. Provides additional information, if any, about the status of the task.
Start Time	Task instance only; system-supplied. The date and time the task started.
End Time	Task instance only; system-supplied. The date and time the task instance completed.
Hold on Start	If enabled, when the task is launched it appears in the Activity display with a status of Held. The task runs when the user releases it.
Hold Reason	Information about why the task will be put on hold when it starts.
Late Start	If enabled, and if the task instance starts after the time specified in the Late Start Time field (see below), the task instance is flagged as late. To determine whether a task instance started late, open the task instance and locate the Started Late field; the field is checked if the instance started late.
Late Start Type	Required if Late Start is enabled. Options are: • **Time** - Flag the task if it starts after the specified time. • **Duration** - Flag the task if it starts a certain amount of time after the programmed start time. The task must have a specific start time.
Late Start Time	Time after which the task start time is considered late. Use hh:mm, 24-hour time

Late Start Duration	Duration (amount of relative time) after which the task is considered to have started late. For a task within a workflow, the duration is the period between the time the workflow starts and the time the task itself starts. For example, a task might have a Late Start Duration of 60 minutes. If the workflow starts at 9:00 a.m. but the task itself does not start until 10:30, the task has started late. For a task that is not within a workflow, Late Start Duration has meaning only if the task has been held upon starting. For example, if a task has a Late Start Duration of 60 minutes and the Hold on Start field is enabled, if the task is not released from hold within the amount of time specified in the Late Start Duration field, the task has started late.
Started Late	Task instance only; system-supplied. This field is flagged if the task started later than the time specified in the Late Start fields.
Late Finish	If enabled, and if the task instance finishes after the time or period specified, the task instance is flagged as late. You can specify a time or duration to determine a late finish (see Late Finish Type below). To determine whether a task instance finished late, open the task instance and locate the Finished Late field; the field is checked if the instance finished after the specified time or lasted longer than expected. This field only appears on the task instance if the user specified a Late Finish in the task definition.
Late Finish Type	If Late Finish is enabled, use this field to specify whether you want to use time (see Late Finish Time below) or duration (see Late Finish Duration below) to determine whether the task instance finished late.
Late Finish Time	If Late Finish Type is Time, use this to specify the time after which the task finish time is considered late. Use hh:mm, 24-hour time.
Late Finish Duration	If Late Finish Type is Duration, use this to specify the longest amount of time this task instance should take to run. You can specify any combination of hours, minutes, and seconds.
Finished Late	Task instance only; system-supplied. This field is flagged if the task finished later than the time or duration specified in the Late Finish fields.
Early Finish	If enabled, and if the task instance finishes before the time or period specified, the task instance is flagged as early. You can specify a time or duration to determine an early finish (see Early Finish Type below). To determine whether a task instance finished early, open the task instance and locate the Finished Early field; the field is checked if the instance finished before the specified time or did not last as long as expected. This field only appears on the task instance if the user added Early Finish specifications to the task definition.
Early Finish Type	If Early Finish is enabled, use this field to specify whether you want to use time (see Early Finish Time below) or duration (see Early Finish Duration below) to determine whether the task instance finished early.
Early Finish Time	If Early Finish Type is Time, use this to specify the time before which the task finish time is considered early. That is, enter a time at which the task should still be running. Use hh:mm, 24-hour time.
Early Finish Duration	If Early Finish Type is Duration, use this to specify the shortest amount of time this task instance should take to run. You can specify any combination of hours, minutes, and seconds.
Finished Early	Task instance only; system-supplied. This field is flagged if the task finished earlier than the time specified in the Early Finish fields.
First Time Ran	Task definition only; system-supplied. Displays after the first time the task runs. The date and time this task first ran.

Lowest Instance Time	Task definition only; system-supplied. Displays after the first time the task runs. The shortest amount of time this task has taken to run.
Last Time Ran	Task definition only; system-supplied. Displays after the first time the task runs. The most recent date and time the task ran.
Average Instance Time	Task definition only; system-supplied. Displays after the first time the task runs. Shows the average amount of time this task takes to run.
Number of Instances	Task definition only; system-supplied. Displays after the first time the task runs. Shows the number of times this task has run.
Highest Instance Time	Task definition only; system-supplied. Displays after the first time the task runs. The longest amount of time this task has taken to run.
Last Instance Duration	Task definition only; system-supplied. Displays after the first time the task runs. The amount of time the task took to run the last time it ran.
Submit button	Submits the new record to the database.
Update button	Saves updates to the record.
Launch Task button	Manually launches the task.
Delete button	Deletes the current record.
View Parent button	Task instance only; Displays this task's parent task (workflow), if any.
Show Details button	Task instance only; displays detailed information about this task instance.
Skip button	Task instance only; for tasks loaded into the schedule that haven't run yet. Allows you to tell the system to skip this task. See Skipping a Task.
Hold button	Task instance only; see Putting a Task on Hold.
Force Finish button	Task instance only; see Force Finishing a Task.
Re-run button	Task instance only; see Re-running a Task.
Output tab	Task instance only. Displays output generated from the process, if any, based on specifications provided by the user in the Automatic Output Retrieval fields in the task definition.
Task Virtual Resources tab	Lists Virtual Resources to which this task is assigned. See Creating Virtual Resources.

Variables tab	Displays all variables associated with this record.
Actions tab	Allows you to specify actions that the system will take automatically based on events that occur during the execution of this task. Events are task instance status, exit codes, late start, late finish, and early finish. Actions are: • **Abort Action** - Abort the task if certain events occur. For details, see Setting Up Abort Actions. • **Email Notification** - Send an email if certain events occur. For details, see Creating Email Notifications. • **SNMP Notification** - Send an email if certain events occur. For details, see Creating SNMP Notifications. • **Set Variable** - Used in workflows to set a variable based on the occurrence of certain events. For details, see Setting Variables in a Workflow.
Task Instances tab	System-supplied. Displays a list of all task instances generated by this task definition.
Triggers tab	Displays a list of all triggers that have been defined to launch this task. Also allows you to add new triggers. If you add a new trigger from this location, the system automatically constructs a default trigger name as follows: <current task name>#TRIGGER#. You can change the default name if desired. For instructions on creating triggers, see Creating Triggers.
Notes tab	Displays all notes associated with this task. See Runbook Notes.
Versions Tab	Stores copies of all previous versions of the current record (see Record Versioning).

Monitoring Task Execution

You can monitor all system activity from the Activity display and can view activity history from the Activity History screen.

File Monitor Task

Overview

The File Monitor task allows you to monitor a specific remote machine for the creation, deletion, change, existence, or non-existence of one or more files at a specific location. In order to run a File Monitor task, you need an Opswise agent (Windows, Linux, Unix, or z/OS) running on the machine where you are monitoring for the file.

File Monitor tasks are meant to be launched using a File (Monitor) trigger or within a workflow. However, there are no technical restrictions on how a File Monitor task can be launched. The processing may differ depending on which of the following methods was used to launch it:

- Launched by a workflow
- Launched by a File Monitor trigger
- Launched manually or by another trigger type

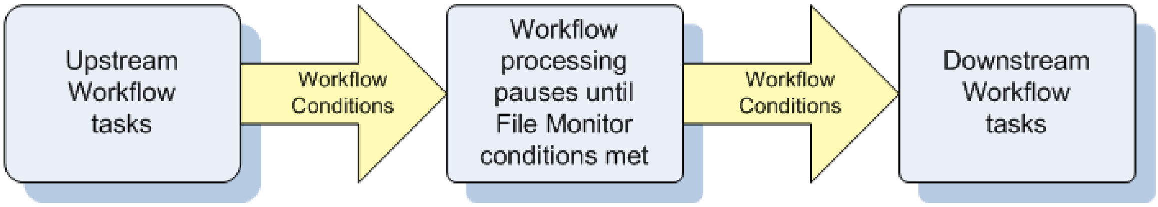
The processing on a File Monitor task for each launching method is described below.

Launching a File Monitor Task Within a Workflow

The File Monitor task can be launched within a Workflow.

In this scenario, the task launches when the upstream workflow conditions are satisfied. Workflow processing then pauses until the conditions in the File Monitor task are satisfied. If the File Monitor is watching for the creation, change, or deletion of a file, the task goes to SUCCESS when the event occurs. If the File Monitor is watching for the existence or non-existence of a file, the task immediately goes to SUCCESS or FAILURE. Subsequent processing depends on the conditions built into the workflow.

The diagram below illustrates the processing for this scenario.


Upstream Workflow tasks
Workflow Conditions
Workflow processing pauses until File Monitor conditions met
Workflow Conditions
Downstream Workflow tasks

Launching a File Monitor Task Using a File (Monitor) Trigger

A common use for the File Monitor task is to launch it using a File (Monitor) trigger, which specifies one or more tasks that are launched when the condition(s) is satisfied.

In this scenario, the File Monitor task launches when its associated File (Monitor) trigger is enabled.

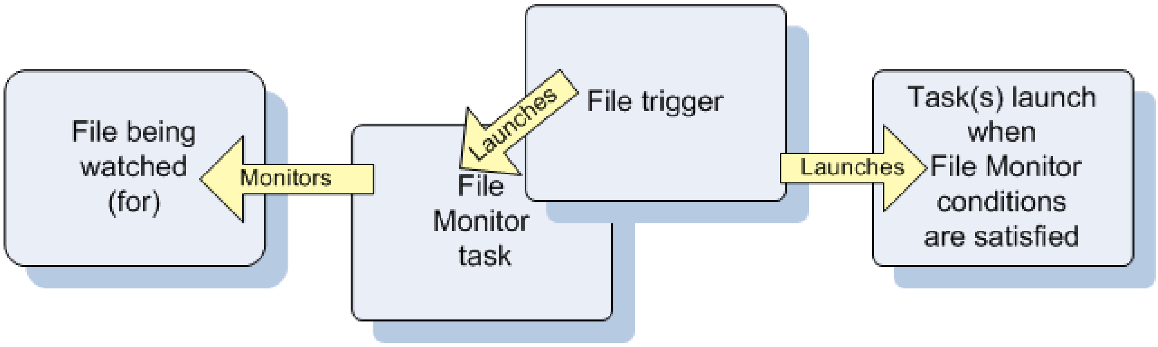
This method is best geared toward watching for the creation, deletion, or change in files. When the conditions in the File Monitor are satisfied, the File Monitor task goes to SUCCESS and the tasks listed in the associated trigger are launched. The File Monitor task continues running until its conditions are satisfied or until the user disables the trigger.

If you use this method to check for the existence or non-existence of a file, as soon as the task is launched it goes to SUCCESS or FINISHED status. If it goes to SUCCESS, the tasks specified in the trigger are launched. A FINISHED status indicates that it found a file that shouldn't be there or didn't find a file that should be there. Both of these cases constitute a "failure" of the conditions and therefore the tasks in the trigger are not launched.

When the File Monitor task goes to FINISHED or SUCCESS, the associated File (Monitor) trigger is automatically disabled.

When you launch a File Monitor task from a File trigger, you cannot manually cancel or force finish the task. You can only stop the task by disabling the trigger. If you manually disable the trigger while the task is still running, the task goes to FINISHED status.

The diagram below illustrates the processing flow for this scenario.



Launching a File Monitor Task Manually or Via Other Trigger

If you manually launch a File Monitor task or launch it using a non-File trigger, such as a Time trigger, the task continues running until its specified conditions are met, at which time the task goes to SUCCESS. No other processing occurs unless you have configured notifications with the task or set up some other task(s) to launch based on the status of this task.

If the conditions are not met, the task runs perpetually or until a user issues a Cancel or Force Finish command against it.

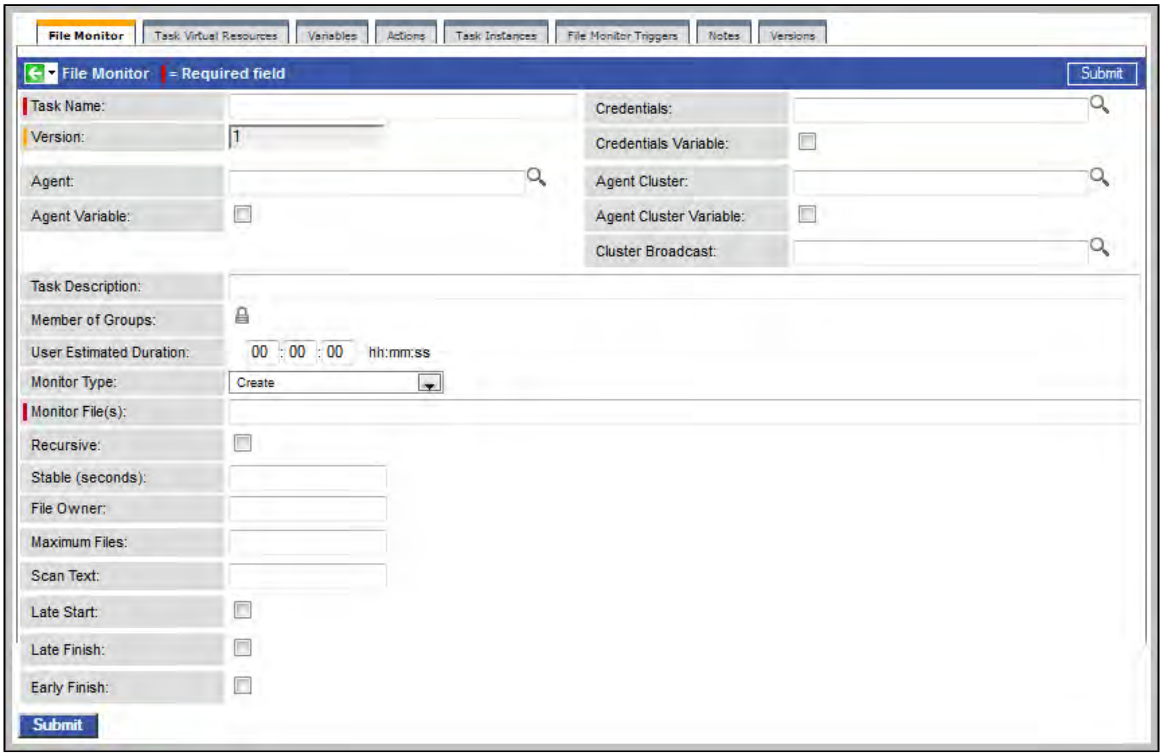
Built-In Variables

The built-in variables outlined below can be used to pass data where appropriate:

- Task and Task Instance Variables
- File Monitor Variables.

Creating a New File Monitor Task

1. From the navigation pane, select **Automation Center > Tasks > File Monitors**. The File Monitors List screen displays.
2. Click **New**. The File Monitor Definition screen displays.



Task Name:
Version: 1
Agent:
Agent Variable: □
Credentials:
Credentials Variable: □
Agent Cluster:
Agent Cluster Variable: □
Cluster Broadcast:
Task Description:
Member of Groups:
User Estimated Duration: 00 : 00 : 00 hh:mm:ss
Monitor Type: Create
Monitor File(s):
Recursive: □
Stable (seconds):
File Owner:
Maximum Files:
Scan Text:
Late Start: □
Late Finish: □
Early Finish: □

3. Using the field descriptions provided below as a guide, complete the fields as needed.
4. Click the **Submit** button to save the record and return to the menu, or right-click on the title bar and select **Save** to save the record and remain on the current display.
5. If appropriate, repeat these steps for any additional tasks you want to add.

Task run statistics appear after the first time this task has been launched.

File Monitor Task Field Descriptions

The table below describes the fields, buttons, and tabs on the task definition and task instance screens. Color coding is provided that differentiates the following three types of fields:

- Fields that display on the task definition and task instance screens are shown in black.
- Fields that display only on the task definition screen are shown in green.
- Fields that display only on the task instance screen are shown in maroon.

Field Name	Description
Task/Instance Name	Required. Name used within Opswise to identify this task. Up to 40 alphanumerics; variables supported. It is the user's responsibility to develop a workable naming scheme for tasks.
Invoked by	Task instance only; system-supplied. How the task instance was launched. One of the following: • Trigger: (Trigger Name) - The instance was launched by the named trigger. • Workflow: (Workflow Name) - The instance was launched by the named workflow. • Manually Launched - The instance was launched by a user. To determine the name of the user: 1. From the Activity or All Task Instances screen, click the task instance name to open the record. 2. The Execution User field identifies the user who launched the task instance.
Task	Task instance only; system-supplied. Hover over the paper icon to display more information about the task instance. Click the paper icon to display the task definition record.
Execution User	Task instance only; system-supplied. If the task was launched manually, the ID of the user who launched it.
Instance Reference Id	Task instance only; system-supplied. The system increments this number each time the task is run.
Credentials	Login credentials Opswise Automation Center will use to access the remote machine. They must be provided either in the resource definition or task definition, or both. Credentials provided in the task definition override any credentials provided in the agent resource definition. See Credentials.
Version	Task definition only; system-supplied. The version number of the current record, which is incremented by the system every time a user updates a record. Click on the Versions tab to view previous versions. For details, see Record Versioning.
Credentials Variable	Optional. If enabled, the Credentials field (see above) converts from a reference field (where you browse and select a record) into a text field that allows you to enter a variable. Use the format: ${variable name}. The variable must be a supported type as described in Using Variables and Functions .
Agent	Optional. The name of the agent resource definition that identifies the machine where the operation will run. If you do not specify an agent, you must specify an agent cluster (see below).
Agent Cluster	Optional. You can specify an agent cluster in addition to or in place of a specific agent. An agent cluster is a group of agents, one of which the system will choose to run this task. If you specify an agent and an agent cluster, Opswise Automation Center first tries to run the task on the specific agent. If the agent is not available, the system reverts to the agent cluster. See Configuring Agent Clusters for more information.

Agent Variable	Optional. If enabled, the Agent field (see above) converts from a reference field (where you browse and select a record) into a text field that allows you to enter a variable. Use the format: ${variable name}. The variable must be a supported type as described in Using Variables and Functions.
Agent Cluster Variable	Optional. If enabled, the Agent Cluster field (see above) converts from a reference field (where you browse and select a record) into a text field that allows you to enter a variable. Use the format: ${variable name}. The variable must be a supported type as described in Using Variables and Functions.
Cluster Broadcast	Task definition only; optional. You can specify a Cluster Broadcast in addition to or in place of a specific agent and/or Agent Cluster. When you specify an agent cluster in the Cluster Broadcast field, Opswise Automation Center runs the task on all the agents in the cluster. Each instance of the task running on its own agent becomes a separate task instance record in the database and displays separately in the Activity monitor. See Configuring Agent Clusters for more information about defining agent clusters.
Task Description	User-supplied description of this record.
Status	Task instance only; system-supplied. See Task Statuses.
Exit Code	Task instance only; system-supplied. The exit code captured by the agent when executing the task (for example, a command or script).
Status Description	Task instance only; system-supplied. Provides additional information, if any, about the status of the task.
Start Time	Task instance only; system-supplied. The date and time the task started.
Duration	Task instance only; system-supplied. The amount of time the task took to run.
End Time	Task instance only; system-supplied. The date and time the task instance completed.
Member of Groups	User-defined. Allows you to select one or more Opswise groups that this record definition belongs to. Click on the lock icon to unlock the field and select groups.
User Estimated Duration	Task definition only; optional. The estimated amount of time it should normally take to run this task. The system uses this information to calculate the User Estimated End Time on a task instance record.

Monitor Type	Type of file event being monitored for. Options: - Create - Wait for the creation of one or more files. - Delete - Wait for the deletion of one or more files. - Change - Monitor for a change in one or more files. [**NOTE:** not supported for z/OS.] - Exists - Check to see if one or more files already exists. - Missing - Check to see if one or more files is missing. When triggered, Exists and Missing file monitors exit with a success or failure code immediately. Change file monitors exit with a success code as soon as they detect a change. Create file monitors are re-armed automatically when they succeeds so it is not necessary to re-trigger them. Exists and Missing file monitors are not continuous, and are only intended to be used as run-once checks; for example, within a workflow to determine a workflow-path to take based on the existence of or non-existence of a file. While they can be hooked into a trigger, it does not typically make sense to do so. When you enable a File Monitor trigger using an Exists or Missing file monitor type, the Exists or Missing file monitor instance will run and immediately report the status of the exists or missing check. If type is Exists and the file exists, or the type is Missing and the file does not exist, the trigger will fire and become disabled. If the type is Exists and the file does not exist or type is Missing and the file does exist, the trigger will not fire, but will still become disabled.
Monitor File(s)	Location and file name of the file or files being monitored for. Variables supported. Wildcards supported.
Recursive	If enabled, the monitor searches the specified directory and all subdirectories.
File Owner	User ID of owner of the file on the operating system. Specifying a file owner limits the search to files with that owner.
Maximum Files	For searches that use wildcards, limits the number of files to be searched.
Stable (seconds)	If Type = Change, a period of time, in seconds, during which the file has not changed.
By percentage (+/-)	If Type = Change, the amount that the file size has changed, expressed as a percentage of the original file size. For example, enter 10 to monitor for a change in file size of 10 percent (larger or smaller).
By Size (+/-)	If Type = Change, used in conjunction with the By scale field, specifies an actual change in file size. For example, to monitor for a change in file size of 10 MB, enter 10 in this field and select MB in the By scale field.
By scale	If Type = Change, used in conjunction with the By Size field, specifies Bytes, KB (kilobytes), or MB (megabytes). For example, to monitor for a change in file size of 10 MB, enter 10 in the By Size field and select MB in this field.
To Size	If Type = Change, used in conjunction with the To scale field, specifies an actual file size that you want to monitor for. For example, to monitor for a file size of 5 KB, enter 5 in this field and select KB in the To scale field.
To scale	If Type = Change, used in conjunction with the To Size field, specifies an actual file size that you want to monitor for. For example, to monitor for a file size of 5 KB, enter 5 in the To Size field and select KB in this field.
Scan Text	Optional. If Type = Create, Change or Exists, this field specifies a string that the monitor will search for in the file or files. Specifying a string means that only files containing the string constitute a match. Opswise will process this field as a regular expression.

Scan Forward	Optional. If Type = Change. If enabled, this field specifies that once the File Monitor has been satisfied, it should continue from where it left off. If it is scanning within a file, it should resume from the point in the file that it last scanned. If it is monitoring for files, it should resume monitoring for the next file. If you are scanning a file that is being overwritten each time and you want to start from the beginning each time, you should disable Scan Forward.
User Estimated End Time	Task instance only; system-supplied. If the user entered information into the User Estimated Duration field in the task definition, the system uses this information to calculate an end time for the task instance, based on the date/time the task instance started.
Shortest Estimated End Time	Task instance only; system-supplied.
Average Estimated End Time	Task instance only; system-supplied.
Longest Estimated End Time	Task instance only; system-supplied.
First Time Ran	Task definition only; system-supplied. Displays after the first time the task runs. The date and time this task first ran.
Lowest Instance Time	Task definition only; system-supplied. Displays after the first time the task runs. The shortest amount of time this task has taken to run.
Last Time Ran	Task definition only; system-supplied. Displays after the first time the task runs. The most recent date and time the task ran.
Average Instance Time	Task definition only; system-supplied. Displays after the first time the task runs. Shows the average amount of time this task takes to run.
Number of Instances	Task definition only; system-supplied. Displays after the first time the task runs. Shows the number of times this task has run.
Highest Instance Time	Task definition only; system-supplied. Displays after the first time the task runs. The longest amount of time this task has taken to run.
Last Instance Duration	Task definition only; system-supplied. Displays after the first time the task runs. The amount of time the task took to run the last time it ran.
Submit button	Submits the new record to the database.
Update button	Saves updates to the record.
Launch Task button	Manually launches the task.
Delete button	Deletes the current record.

View Parent button	Task instance only; Displays this task's parent task (workflow), if any.
Show Details button	Task instance only; displays detailed information about this task instance.
Skip button	Task instance only; for tasks loaded into the schedule that haven't run yet. Allows you to tell the system to skip this task. See Skipping a Task.
Hold button	Task instance only; see Putting a Task on Hold.
Force Finish button	Task instance only; see Force Finishing a Task.
Re-run button	Task instance only; see Re-running a Task.
Output tab	Task instance only. Displays output generated from the process, if any, based on specifications provided by the user in the Automatic Output Retrieval fields in the task definition.
Task Virtual Resources tab	Lists Virtual Resources to which this task is assigned. See Creating Virtual Resources.
Variables tab	Displays all variables associated with this record.
Actions tab	Allows you to specify actions that the system will take automatically based on events that occur during the execution of this task. Events are task instance status, exit codes, late start, late finish, and early finish. Actions are: • **Abort Action** - Abort the task if certain events occur. For details, see Setting Up Abort Actions. • **Email Notification** - Send an email if certain events occur. For details, see Creating Email Notifications. • **SNMP Notification** - Send an email if certain events occur. For details, see Creating SNMP Notifications. • **Set Variable** - Used in workflows to set a variable based on the occurrence of certain events. For details, see Setting Variables in a Workflow.
Task Instances tab	System-supplied. Displays a list of all task instances generated by this task definition.
File Monitor Triggers tab	Displays a list of all File Monitor triggers associated with this task. Enabling any of the triggers will launch this task. When the conditions in the task are satisfied, the tasks specified in the trigger will launch. For details, see Launching a File Monitor Task Using a File (Monitor) Trigger.
Notes tab	Displays all notes associated with this task. See Runbook Notes.
Versions Tab	Stores copies of all previous versions of the current record (see Record Versioning).

Monitoring Task Execution

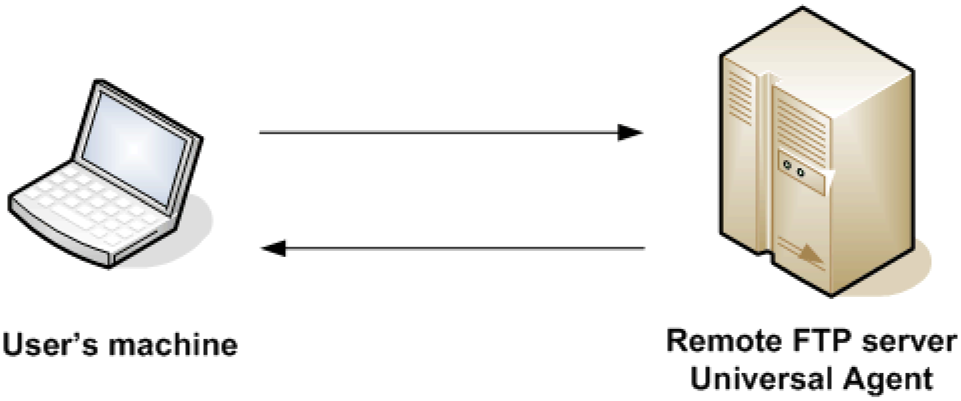
You can monitor all system activity from the Activity display and can view activity history from the Activity History screen.

FTP File Monitor Task

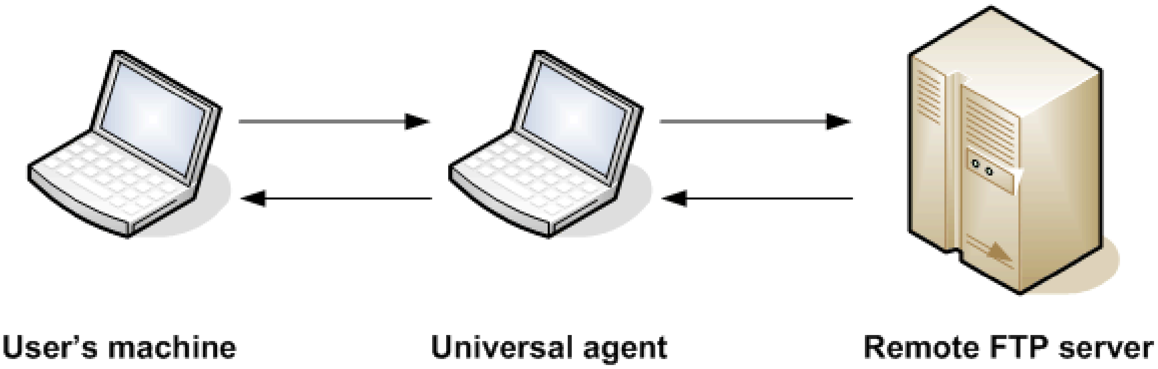
Overview

The FTP File Monitor task allows you to monitor for a file on a remote machine where an FTP server is running. The FTP File Monitor connects to the FTP server rather than the machine's file system to monitor for files. The FTP File Monitor can be used only within a workflow; you cannot run a FTP File Monitor task based on a trigger. To run an FTP File Monitor task, you need an Opswise Universal Automation Center Agent (UAG) to communicate with the FTP server. The agent can but does not have to be running on the same machine as the FTP server.

In the example shown below, the user wants to monitor for a file on a remote FTP Server that has UAG running on it. In this case, the login credentials for the UAG machine and the FTP server machine are the same.



In the example shown below, the user wants to monitor for a file on a remote FTP Server that does not have UAG running on it. In this case, the FTP File Monitor task definition provides an address and login credentials for the machine where UAG is running as well as address and login credentials for the FTP server.



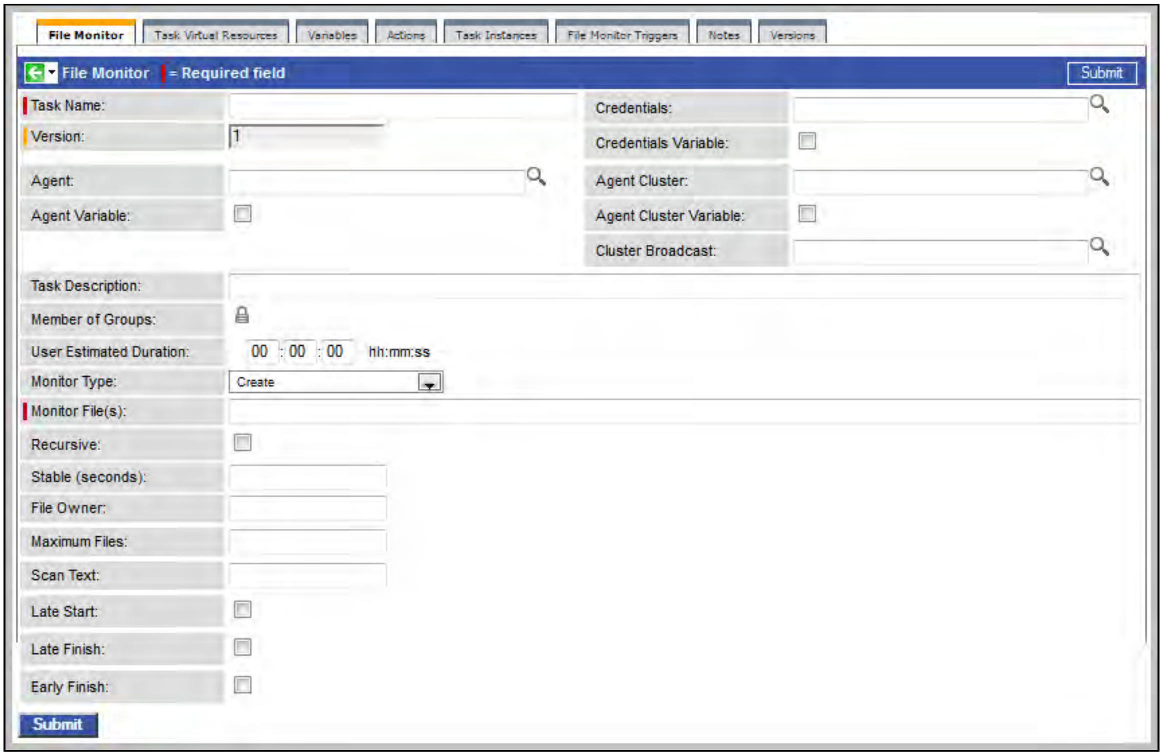
Built-In Variables

The built-in variables outlined below can be used in an FTP File Monitor task to pass data where appropriate:

- Task and Task Instance Variables
- FTP File Monitor Variables.

Creating a New FTP File Monitor Task

1. From the navigation pane, select **Automation Center > Tasks > FTP File Monitors**. The FTP File Monitors List screen displays.
2. Click **New**. The FTP File Monitor Task Definition screen displays.



Task Name:
Version: 1
Agent:
Agent Variable: □
Credentials:
Credentials Variable: □
Agent Cluster:
Agent Cluster Variable: □
Cluster Broadcast:
Task Description:
Member of Groups:
User Estimated Duration: 00 : 00 : 00 hh:mm:ss
Monitor Type: Create
Monitor File(s):
Recursive: □
Stable (seconds):
File Owner:
Maximum Files:
Scan Text:
Late Start: □
Late Finish: □
Early Finish: □

3. Using the field descriptions provided below as a guide, complete the fields as needed.
4. Click the **Submit** button to save the record and return to the menu, or right-click on the title bar and select **Save** to save the record and remain on the current display.
5. If appropriate, repeat these steps for any additional tasks you want to add.

Task run statistics appear after the first time this task has been launched.

FTP File Monitor Task Field Descriptions

The table below describes the fields, buttons, and tabs on the task definition and task instance screens. Color coding is provided that differentiates the following three types of fields:

- Fields that display on the task definition and task instance screens are shown in black.
- Fields that display only on the task definition screen are shown in green.
- Fields that display only on the task instance screen are shown in maroon.

Field Name	Description
Task/Instance Name	Required. Name used within Opswise to identify this task. Up to 40 alphanumerics; variables supported. It is the user's responsibility to develop a workable naming scheme for tasks.
Invoked by	Task instance only; system-supplied. How the task instance was launched. One of the following: • Trigger: (Trigger Name) - The instance was launched by the named trigger. • Workflow: (Workflow Name) - The instance was launched by the named workflow. • Manually Launched - The instance was launched by a user. To determine the name of the user: 1. From the Activity or All Task Instances screen, click the task instance name to open the record. 2. The Execution User field identifies the user who launched the task instance.
Task	Task instance only; system-supplied. Hover over the paper icon to display more information about the task instance. Click the paper icon to display the task definition record.

Execution User	Task instance only; system-supplied. If the task was launched manually, the ID of the user who launched it.
Instance Reference Id	Task instance only; system-supplied. The system increments this number each time the task is run.
Credentials	Login credentials Opswise Automation Center will use to access the remote machine. They must be provided either in the resource definition or task definition, or both. Credentials provided in the task definition override any credentials provided in the agent resource definition. See Credentials.
Version	Task definition only; system-supplied. The version number of the current record, which is incremented by the system every time a user updates a record. Click on the Versions tab to view previous versions. For details, see Record Versioning.
Credentials Variable	Optional. If enabled, the Credentials field (see above) converts from a reference field (where you browse and select a record) into a text field that allows you to enter a variable. Use the format: ${variable name}. The variable must be a supported type as described in Using Variables and Functions .
Agent	Optional. The name of the agent resource definition that identifies the machine where the operation will run. If you do not specify an agent, you must specify an agent cluster (see below).
Agent Cluster	Optional. You can specify an agent cluster in addition to or in place of a specific agent. An agent cluster is a group of agents, one of which the system will choose to run this task. If you specify an agent and an agent cluster, Opswise Automation Center first tries to run the task on the specific agent. If the agent is not available, the system reverts to the agent cluster. See Configuring Agent Clusters for more information.
Agent Variable	Optional. If enabled, the Agent field (see above) converts from a reference field (where you browse and select a record) into a text field that allows you to enter a variable. Use the format: ${variable name}. The variable must be a supported type as described in Using Variables and Functions.
Agent Cluster Variable	Optional. If enabled, the Agent Cluster field (see above) converts from a reference field (where you browse and select a record) into a text field that allows you to enter a variable. Use the format: ${variable name}. The variable must be a supported type as described in Using Variables and Functions.
Cluster Broadcast	Task definition only; optional. You can specify a Cluster Broadcast in addition to or in place of a specific agent and/or Agent Cluster. When you specify an agent cluster in the Cluster Broadcast field, Opswise Automation Center runs the task on all the agents in the cluster. Each instance of the task running on its own agent becomes a separate task instance record in the database and displays separately in the Activity monitor. See Configuring Agent Clusters for more information about defining agent clusters.
Task Description	User-supplied description of this record.
Status	Task instance only; system-supplied. See Task Statuses.
Status Description	Task instance only; system-supplied. Provides additional information, if any, about the status of the task.
Exit Code	Task instance only; system-supplied. The exit code captured by the agent when executing the task (for example, a command or script).

Start Time	Task instance only; system-supplied. The date and time the task started.
Duration	Task instance only; system-supplied. The amount of time the task took to run.
End Time	Task instance only; system-supplied. The date and time the task instance completed.
Hold on Start	If enabled, when the task is launched it appears in the Activity display with a status of Held. The task runs when the user releases it.
Member of Groups	User-defined. Allows you to select one or more Opswise groups that this record definition belongs to. Click on the lock icon to unlock the field and select groups.
User Estimated Duration	Task definition only; optional. The estimated amount of time it should normally take to run this task. The system uses this information to calculate the User Estimated End Time on a task instance record.
Monitor Type	Type of file event being monitored for. Options: - **Exists** - Checks to see if the file exists. - **Missing** - Checks to see if the file does not exist.
Wait until Satisfied	If enabled, the task instance starts and continues to run until one of the following events occurs: - If Monitor Type = Exists and the specified file exists or appears, the task instance completes with a status of SUCCESS. - If Monitor Type = Missing and the specified file does not exist or exists then disappears, the task instance completes with a status of SUCCESS. If not enabled, the task instance: 1. Starts. 2. Checks for the existence of the file. 3. Takes one of the following actions: - If Monitor Type = Exists and if the file exists, the task instance completes with a status of SUCCESS. - If Monitor Type = Exists and if the file does not exist, the task instance completes with a status of FAILURE. - If Monitor Type = Missing and if the file exists, the task instance completes with a status of FAILURE. - If Monitor Type = Missing and if the file does not exist, the task instance completes with a status of SUCCESS.
Server Type	Type of FTP server. Options: - FTP - SFTP
Remote Server	Name or IP address of the File Transfer server. This machine may or may not be the same as the Opswise agent machine.
FTP Credentials	Login credentials the Opswise agent will use to access the FTP or SFTP server machine. If the File Transfer server and Opswise agent are running on the same machine, enter the same credentials as those you entered in the Credentials field.

Transfer Mode	Transfer mode. Options: - Active - Passive - Extended Passive
FTP Credentials Variable	Optional. If enabled, the FTP Credentials field (see above) converts from a reference field (where you browse and select a record) into a text field that allows you to enter a variable. Use the format: ${variable name}. The variable must be a supported type as described in Using Variables and Functions.
Remote Filename	Path and file name on the remote server. That is, the "transfer to" filename.

Job Card (z/OS only)	For z/OS, the job card information for the JCL statement. Example:
	`//File TransferJOB01 JOB (File`
	`Transfer,001),FANNY,MSGCLASS=X,MSGLEVEL=(1,1),NOTIFY=&SYSUID,CLASS=A`

User Estimated End Time	Task instance only; system-supplied. If the user entered information into the User Estimated Duration field in the task definition, the system uses this information to calculate an end time for the task instance, based on the date/time the task instance started.
Shortest Estimated End Time	Task instance only; system-supplied.
Average Estimated End Time	Task instance only; system-supplied.
Longest Estimated End Time	Task instance only; system-supplied.
Late Start	If enabled, and if the task instance starts after the time specified in the Late Start Time field (see below), the task instance is flagged as late. To determine whether a task instance started late, open the task instance and locate the Started Late field; the field is checked if the instance started late.
Late Start Type	Required if Late Start is enabled. Options are: - **Time** - Flag the task if it starts after the specified time. - **Duration** - Flag the task if it starts a certain amount of time after the programmed start time. The task must have a specific start time.
Late Start Time	Time after which the task start time is considered late. Use hh:mm, 24-hour time
Late Start Duration	Duration (amount of relative time) after which the task is considered to have started late. For a task within a workflow, the duration is the period between the time the workflow starts and the time the task itself starts. For example, a task might have a Late Start Duration of 60 minutes. If the workflow starts at 9:00 a.m. but the task itself does not start until 10:30, the task has started late. For a task that is not within a workflow, Late Start Duration has meaning only if the task has been held upon starting. For example, if a task has a Late Start Duration of 60 minutes and the Hold on Start field is enabled, if the task is not released from hold within the amount of time specified in the Late Start Duration field, the task has started late.

Started Late	Task instance only; system-supplied. This field is flagged if the task started later than the time specified in the Late Start fields.
Late Finish	If enabled, and if the task instance finishes after the time or period specified, the task instance is flagged as late. You can specify a time or duration to determine a late finish (see Late Finish Type below). To determine whether a task instance finished late, open the task instance and locate the Finished Late field; the field is checked if the instance finished after the specified time or lasted longer than expected. This field only appears on the task instance if the user specified a Late Finish in the task definition.
Late Finish Type	If Late Finish is enabled, use this field to specify whether you want to use time (see Late Finish Time below) or duration (see Late Finish Duration below) to determine whether the task instance finished late.
Late Finish Time	If Late Finish Type is Time, use this to specify the time after which the task finish time is considered late. Use hh:mm, 24-hour time.
Late Finish Duration	If Late Finish Type is Duration, use this to specify the longest amount of time this task instance should take to run. You can specify any combination of hours, minutes, and seconds.
Finished Late	Task instance only; system-supplied. This field is flagged if the task finished later than the time or duration specified in the Late Finish fields.
Early Finish	If enabled, and if the task instance finishes before the time or period specified, the task instance is flagged as early. You can specify a time or duration to determine an early finish (see Early Finish Type below). To determine whether a task instance finished early, open the task instance and locate the Finished Early field; the field is checked if the instance finished before the specified time or did not last as long as expected. This field only appears on the task instance if the user added Early Finish specifications to the task definition.
Early Finish Type	If Early Finish is enabled, use this field to specify whether you want to use time (see Early Finish Time below) or duration (see Early Finish Duration below) to determine whether the task instance finished early.
Early Finish Time	If Early Finish Type is Time, use this to specify the time before which the task finish time is considered early. That is, enter a time at which the task should still be running. Use hh:mm, 24-hour time.
Early Finish Duration	If Early Finish Type is Duration, use this to specify the shortest amount of time this task instance should take to run. You can specify any combination of hours, minutes, and seconds.
Finished Early	Task instance only; system-supplied. This field is flagged if the task finished earlier than the time specified in the Early Finish fields.
First Time Ran	Task definition only; system-supplied. Displays after the first time the task runs. The date and time this task first ran.
Lowest Instance Time	Task definition only; system-supplied. Displays after the first time the task runs. The shortest amount of time this task has taken to run.
Last Time Ran	Task definition only; system-supplied. Displays after the first time the task runs. The most recent date and time the task ran.

Average Instance Time	Task definition only; system-supplied. Displays after the first time the task runs. Shows the average amount of time this task takes to run.
Number of Instances	Task definition only; system-supplied. Displays after the first time the task runs. Shows the number of times this task has run.
Highest Instance Time	Task definition only; system-supplied. Displays after the first time the task runs. The longest amount of time this task has taken to run.
Last Instance Duration	Task definition only; system-supplied. Displays after the first time the task runs. The amount of time the task took to run the last time it ran.
Submit button	Submits the new record to the database.
Update button	Saves updates to the record.
Launch Task button	Manually launches the task.
Delete button	Deletes the current record.
View Parent button	Task instance only; Displays this task's parent task (workflow), if any.
Show Details button	Task instance only; displays detailed information about this task instance.
Skip button	Task instance only; for tasks loaded into the schedule that haven't run yet. Allows you to tell the system to skip this task. See Skipping a Task.
Hold button	Task instance only; see Putting a Task on Hold.
Force Finish button	Task instance only; see Force Finishing a Task.
Output tab	Task instance only. Displays output generated from the process, if any, based on specifications provided by the user in the Automatic Output Retrieval fields in the task definition.
Task Virtual Resources tab	Lists Virtual Resources to which this task is assigned. See Creating Virtual Resources.
Variables tab	Displays all variables associated with this record.

Actions tab	Allows you to specify actions that the system will take automatically based on events that occur during the execution of this task. Events are task instance status, exit codes, late start, late finish, and early finish. Actions are: • **Abort Action** - Abort the task if certain events occur. For details, see Setting Up Abort Actions. • **Email Notification** - Send an email if certain events occur. For details, see Creating Email Notifications. • **SNMP Notification** - Send an email if certain events occur. For details, see Creating SNMP Notifications. • **Set Variable** - Used in workflows to set a variable based on the occurrence of certain events. For details, see Setting Variables in a Workflow.
Task Instances tab	System-supplied. Displays a list of all task instances generated by this task definition.
Triggers tab	Displays a list of all triggers that have been defined to launch this task. Also allows you to add new triggers. If you add a new trigger from this location, the system automatically constructs a default trigger name as follows: <current task name>#TRIGGER#. You can change the default name if desired. For instructions on creating triggers, see Creating Triggers.
Notes tab	Displays all notes associated with this task. See Runbook Notes.
Versions Tab	Stores copies of all previous versions of the current record (see Record Versioning).

Specifying When a Task Runs

The FTP File Monitor can be used only within a workflow; you cannot run a FTP File Monitor task based on a trigger.

Monitoring Task Execution

You can monitor all system activity from the Activity display and can view activity history from the Activity History screen.

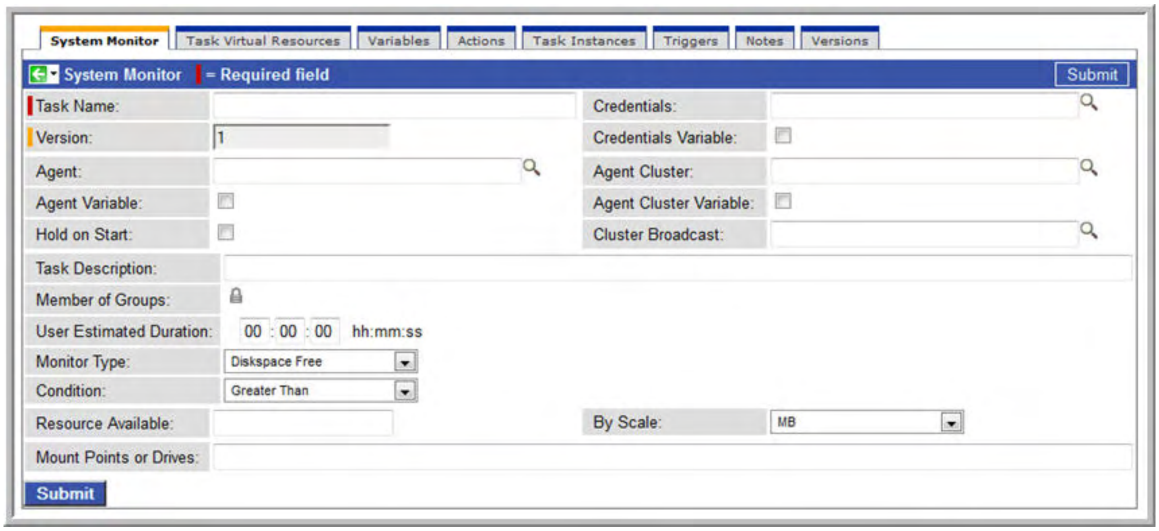
System Monitor Task

Overview

The System Monitor task allows you to monitor a specific remote machine and check for free diskspace. You might use this task to check for sufficient diskspace before running a task on it that requires a specific amount. In order for this task to execute, the remote machine must have Universal Automation Center Agent (UAG) running on it.

Creating a New System Monitor Task

1. From the navigation pane, select **Automation Center > Tasks > System Monitors**. The System Monitors List displays.
2. Click **New**. The System Monitor Definition screen displays.



Version: 1

User Estimated Duration: 00 : 00 : 00 hh:mm:ss

Monitor Type: Diskspace Free

Condition: Greater Than

By Scale: MB

3. Using the field descriptions provided below as a guide, complete the fields as needed.
4. Click the **Submit** button to save the record and return to the menu, or right-click on the title bar and select **Save** to save the record and remain on the current display.
5. If appropriate, repeat these steps for any additional tasks you want to add.

Task run statistics appear after the first time this task has been launched.

System Monitor Field Descriptions

The table below describes the fields, buttons, and tabs on the task definition and task instance screens. Color coding is provided that differentiates the following three types of fields:

- Fields that display on the task definition and task instance screens are shown in black.
- Fields that display only on the task definition screen are shown in green.
- Fields that display only on the task instance screen are shown in maroon.

Field Name	Description
Task/Instance Name	Required. Name used within Opswise to identify this task. Up to 40 alphanumerics; variables supported. It is the user's responsibility to develop a workable naming scheme for tasks.

Invoked by	Task instance only; system-supplied. How the task instance was launched. One of the following: • Trigger: (Trigger Name) - The instance was launched by the named trigger. • Workflow: (Workflow Name) - The instance was launched by the named workflow. • Manually Launched - The instance was launched by a user. To determine the name of the user: 1. From the Activity or All Task Instances screen, click the task instance name to open the record. 2. The Execution User field identifies the user who launched the task instance.
Task	Task instance only; system-supplied. Hover over the paper icon to display more information about the task instance. Click the paper icon to display the task definition record.
Execution User	Task instance only; system-supplied. If the task was launched manually, the ID of the user who launched it.
Instance Reference Id	Task instance only; system-supplied. The system increments this number each time the task is run.
Credentials	Login credentials Opswise Automation Center will use to access the remote machine. They must be provided either in the resource definition or task definition, or both. Credentials provided in the task definition override any credentials provided in the agent resource definition. See Credentials.
Version	Task definition only; system-supplied. The version number of the current record, which is incremented by the system every time a user updates a record. Click on the Versions tab to view previous versions. For details, see Record Versioning.
Credentials Variable	Optional. If enabled, the Credentials field (see above) converts from a reference field (where you browse and select a record) into a text field that allows you to enter a variable. Use the format: ${variable name}. The variable must be a supported type as described in Using Variables and Functions .
Agent	Optional. The name of the agent resource definition that identifies the machine where the operation will run. If you do not specify an agent, you must specify an agent cluster (see below).
Agent Cluster	Optional. You can specify an agent cluster in addition to or in place of a specific agent. An agent cluster is a group of agents, one of which the system will choose to run this task. If you specify an agent and an agent cluster, Opswise Automation Center first tries to run the task on the specific agent. If the agent is not available, the system reverts to the agent cluster. See Configuring Agent Clusters for more information.
Agent Variable	Optional. If enabled, the Agent field (see above) converts from a reference field (where you browse and select a record) into a text field that allows you to enter a variable. Use the format: ${variable name}. The variable must be a supported type as described in Using Variables and Functions.
Agent Cluster Variable	Optional. If enabled, the Agent Cluster field (see above) converts from a reference field (where you browse and select a record) into a text field that allows you to enter a variable. Use the format: ${variable name}. The variable must be a supported type as described in Using Variables and Functions.
Hold on Start	If enabled, when the task is launched it appears in the Activity display with a status of Held. The task runs when the user releases it.
Cluster Broadcast	Task definition only; optional. You can specify a Cluster Broadcast in addition to or in place of a specific agent and/or Agent Cluster. When you specify an agent cluster in the Cluster Broadcast field, Opswise Automation Center runs the task on all the agents in the cluster. Each instance of the task running on its own agent becomes a separate task instance record in the database and displays separately in the Activity monitor. See Configuring Agent Clusters for more information about defining agent clusters.

Task Description	User-supplied description of this record.
Status	Task instance only; system-supplied. See Task Statuses.
Exit Code	Task instance only; system-supplied. The exit code captured by the agent when executing the task (for example, a command or script).
Status Description	Task instance only; system-supplied. Provides additional information, if any, about the status of the task.
Start Time	Task instance only; system-supplied. The date and time the task started.
Duration	Task instance only; system-supplied. The amount of time the task took to run.
End Time	Task instance only; system-supplied. The date and time the task instance completed.
Member of Groups	User-defined. Allows you to select one or more Opswise groups that this record definition belongs to. Click on the lock icon to unlock the field and select groups.
User Estimated Duration	Task definition only; optional. The estimated amount of time it should normally take to run this task. The system uses this information to calculate the User Estimated End Time on a task instance record.
Monitor Type	Type of system status being monitored for. Options: • Diskspace Free - Monitors for the amount of free diskspace.
Condition	Specifies whether you want to check for free diskspace greater than or less than the amount specified in the Resource Available field.
Resource Available	Required. Used in conjunction with the By Scale field. Enter a number indicating the amount of the resource you are checking for. For example, to check to see if the machine has at least one gigabyte of free diskspace, select Greater Than in the Condition field, type 1 in the Resource Available field, and select GB in the By Scale field.
By Scale	Scale of the number you entered in the Resource Available field. Options: KB (kilobyte), MB (megabyte), GB (gigabyte).
Mount Points or Drives	Required. Use this field to limit the check to a specific mount point or drive, such as drive C: for Windows.
User Estimated End Time	Task instance only; system-supplied. If the user entered information into the User Estimated Duration field in the task definition, the system uses this information to calculate an end time for the task instance, based on the date/time the task instance started.
Shortest Estimated End Time	Task instance only; system-supplied.
Average Estimated End Time	Task instance only; system-supplied.

Longest Estimated End Time	Task instance only; system-supplied.
First Time Ran	Task definition only; system-supplied. Displays after the first time the task runs. The date and time this task first ran.
Lowest Instance Time	Task definition only; system-supplied. Displays after the first time the task runs. The shortest amount of time this task has taken to run.
Last Time Ran	Task definition only; system-supplied. Displays after the first time the task runs. The most recent date and time the task ran.
Average Instance Time	Task definition only; system-supplied. Displays after the first time the task runs. Shows the average amount of time this task takes to run.
Number of Instances	Task definition only; system-supplied. Displays after the first time the task runs. Shows the number of times this task has run.
Highest Instance Time	Task definition only; system-supplied. Displays after the first time the task runs. The longest amount of time this task has taken to run.
Last Instance Duration	Task definition only; system-supplied. Displays after the first time the task runs. The amount of time the task took to run the last time it ran.
Submit button	Submits the new record to the database.
Update button	Saves updates to the record.
Launch Task button	Manually launches the task.
Delete button	Deletes the current record.
View Parent button	Task instance only; Displays this task's parent task (workflow), if any.
Show Details button	Task instance only; displays detailed information about this task instance.
Skip button	Task instance only; for tasks loaded into the schedule that haven't run yet. Allows you to tell the system to skip this task. See Skipping a Task.
Hold button	Task instance only; see Putting a Task on Hold.
Force Finish button	Task instance only; see Force Finishing a Task.
Re-run button	Task instance only; see Re-running a Task.

Output tab	Task instance only. Displays output generated from the process, if any, based on specifications provided by the user in the Automatic Output Retrieval fields in the task definition.
Task Virtual Resources tab	Lists Virtual Resources to which this task is assigned. See Creating Virtual Resources.
Variables tab	Displays all variables associated with this record.
Actions tab	Allows you to specify actions that the system will take automatically based on events that occur during the execution of this task. Events are task instance status, exit codes, late start, late finish, and early finish. Actions are: • **Abort Action** - Abort the task if certain events occur. For details, see Setting Up Abort Actions. • **Email Notification** - Send an email if certain events occur. For details, see Creating Email Notifications. • **SNMP Notification** - Send an email if certain events occur. For details, see Creating SNMP Notifications. • **Set Variable** - Used in workflows to set a variable based on the occurrence of certain events. For details, see Setting Variables in a Workflow.
Task Instances tab	System-supplied. Displays a list of all task instances generated by this task definition.
Triggers tab	Displays a list of all triggers that have been defined to launch this task. Also allows you to add new triggers. If you add a new trigger from this location, the system automatically constructs a default trigger name as follows: <current task name>#TRIGGER#. You can change the default name if desired. For instructions on creating triggers, see Creating Triggers.
Notes tab	Displays all notes associated with this task. See Runbook Notes.
Versions Tab	Stores copies of all previous versions of the current record (see Record Versioning).

Specifying When a Task Runs

You can run the task as part of a workflow, specify triggers that run the task automatically based on times or events, or run the task manually.

Monitoring Task Execution

You can monitor all system activity from the Activity display and can view activity history from the Activity History screen.

Copying Tasks

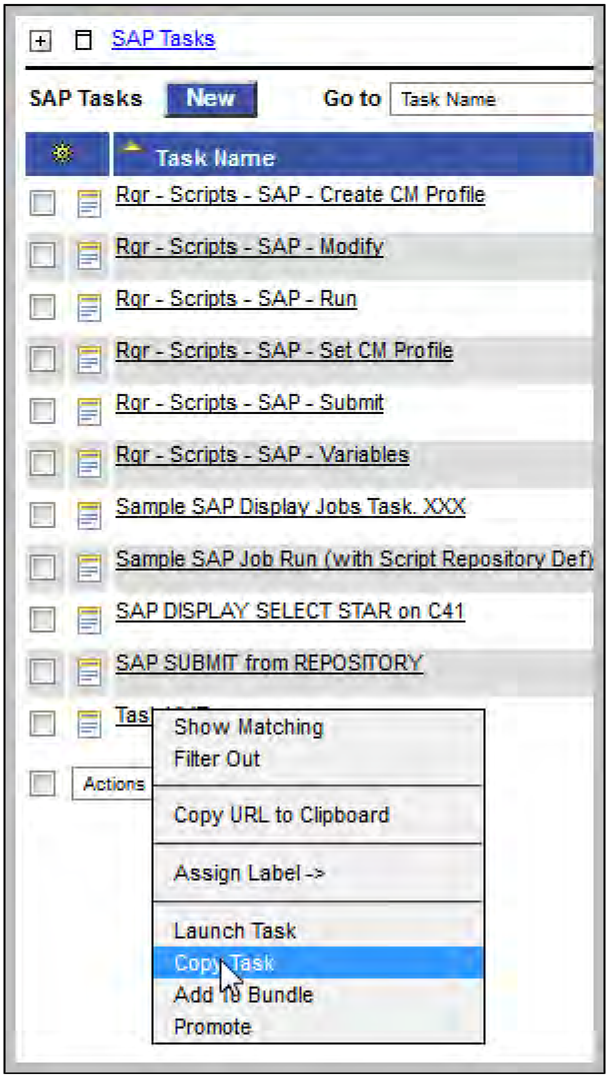
Overview

You can make copies of all Opswise records, including tasks, using the standard methods for Saving, Updating, Deleting, and Copying Records. However, the standard method does not make copies of other records that may be attached to the task, such as Notifications, Actions, Variables, and so on.

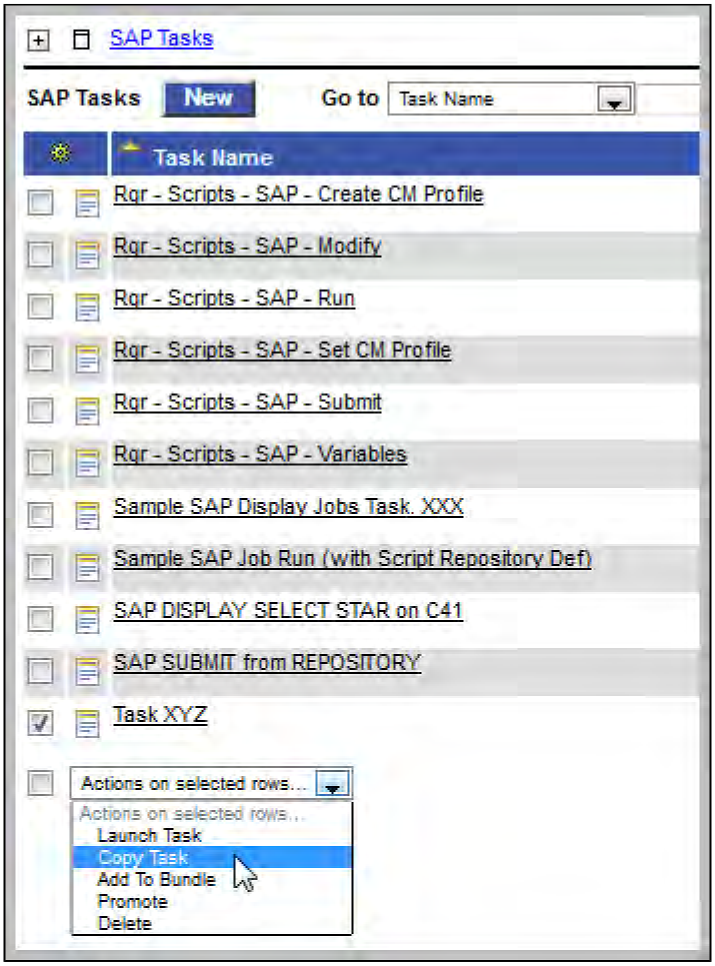
The Copy Task option allows you to make a complete copy of a task, including all of its associated records, such as variables and notes. It does not copy referenced records, such as virtual resources, but retains the relationship to these records for the copied task.

Copying One or More Tasks on the Tasks List

1. From the navigation pane, select a task type from **Automation Center > Tasks**. The Tasks List screen for that task type displays.
2. Locate the task(s) you want to copy (see Searching for Records).
3. Copy the task(s) using either of two methods:
    a. To copy a single task, hover over the task name and right-click. From the pop-up menu, select **Copy Task**.


SAP Tasks
SAP Tasks
New
Go to
Task Name
Task Name
Rgr - Scripts - SAP - Create CM Profile
Rgr - Scripts - SAP - Modify
Rgr - Scripts - SAP - Run
Rgr - Scripts - SAP - Set CM Profile
Rgr - Scripts - SAP - Submit
Rgr - Scripts - SAP - Variables
Sample SAP Display Jobs Task. XXX
Sample SAP Job Run (with Script Repository Def)
SAP DISPLAY SELECT STAR on C41
SAP SUBMIT from REPOSITORY
Tas
Actions
Show Matching
Filter Out
Copy URL to Clipboard
Assign Label ->
Launch Task
Copy Task
Add to Bundle
Promote

b. To copy one or more tasks, click the box to the left of each task name. From the **Action on selected rows...** drop-down list at the bottom of the page, select **Copy Task**.

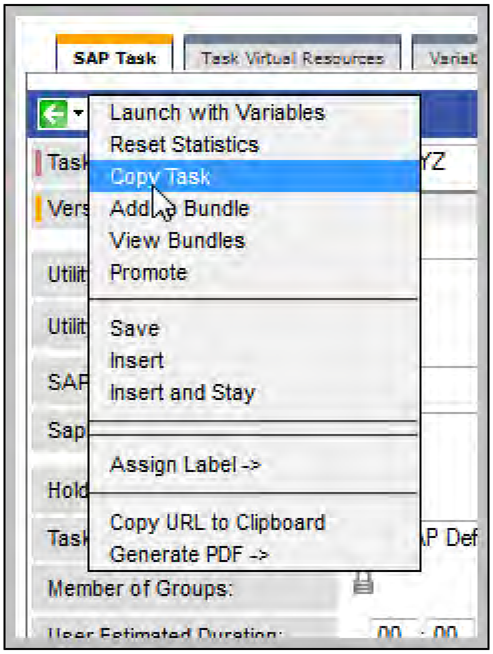


	Task Name
□	Rgr - Scripts - SAP - Create CM Profile
□	Rgr - Scripts - SAP - Modify
□	Rgr - Scripts - SAP - Run
□	Rgr - Scripts - SAP - Set CM Profile
□	Rgr - Scripts - SAP - Submit
□	Rgr - Scripts - SAP - Variables
□	Sample SAP Display Jobs Task. XXX
□	Sample SAP Job Run (with Script Repository Def)
□	SAP DISPLAY SELECT STAR on C41
□	SAP SUBMIT from REPOSITORY
☑	Task XYZ

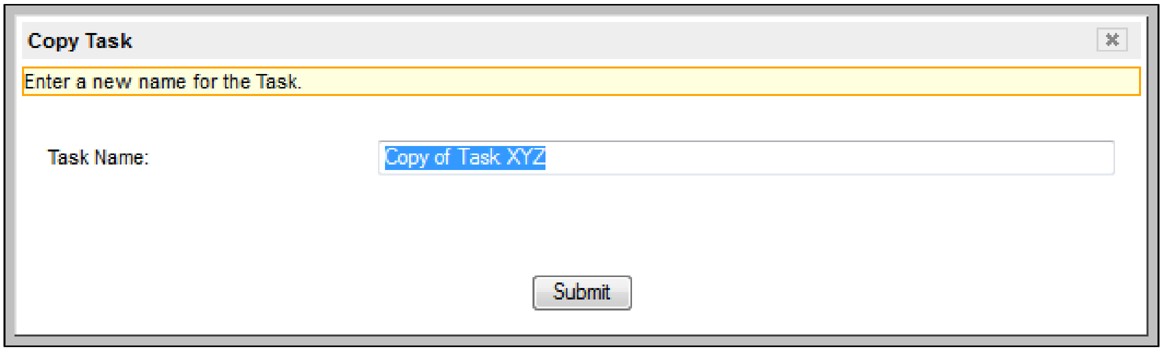
4. The system copies the task(s), automatically creating the new name by prepending the original name with "Copy of" (for example, "Copy of Task XYZ"), and adds it to the list. If the new name already exists, the system appends a counter to the name, such as "Copy of Task XYZ 1", "Copy of Task XYZ 2", and so on, until it finds a name that is available.
5. To modify the name or any other information in the task, open the new task, make your changes, and click **Update**.

Copying a Task on the Task Definition Form

1. Open the task you want to copy.
2. Hover over the down arrow above **Task Name:**. A pop-up menu displays:



3. Click **Copy Task**. Another window appears, prompting for a name for the new task. The default is the original task name, prepended with "Copy of," as shown in the following example:


Copy Task

Enter a new name for the Task.

Task Name: Copy of Task XYZ

4. Enter a new name for the task and click **Submit**. The system copies the task and all its attachments and saves it under the new name. (If the new name already exists, the copy will fail.)

Creating Email Notifications

Overview

You can attach one or more Email notifications to any Opswise task. For Workflow tasks, you can also specify whether you want the email to be triggered by the workflow itself, the workflow and/or its tasks, or by the tasks only.

You can trigger the notification based on one or more of the following events associated with the task instance to which you attach the notification:

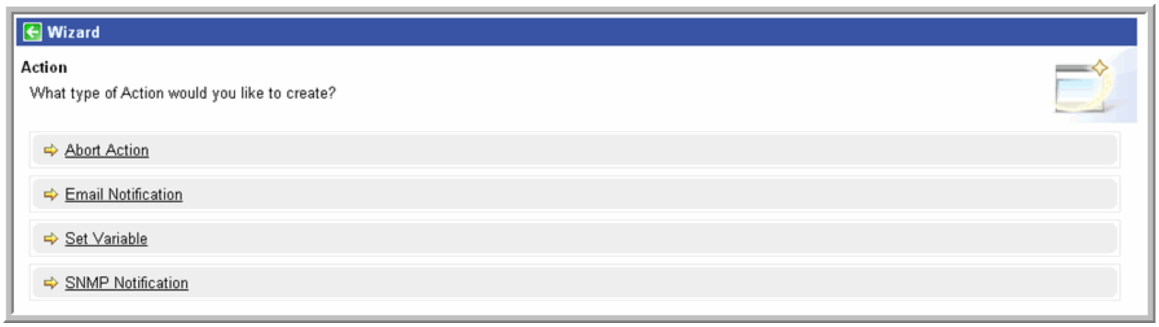
- Status or statuses of the task instance
- Exit code(s) generated by the program (along with at least one status)
- Late start
- Early or late finish

In order to generate Email notifications, you need an Email connection defined, which provides the Email server name and other pertinent information.

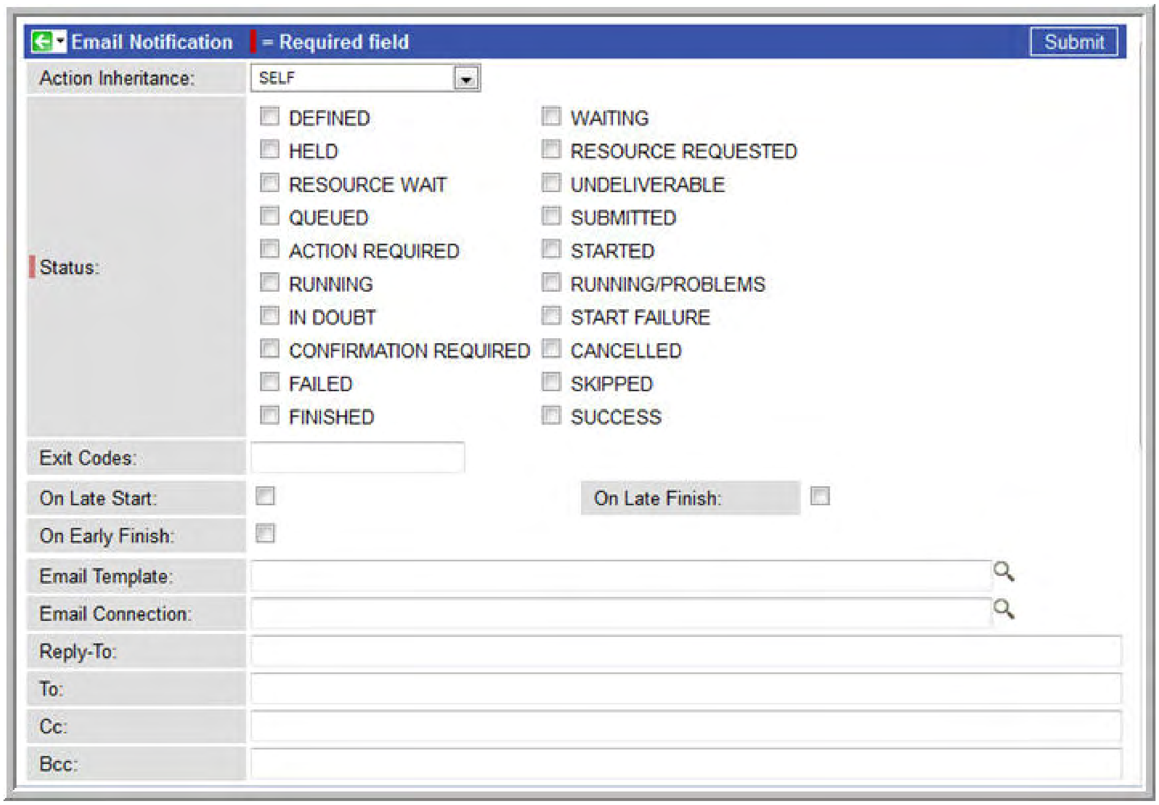
You can also generate notifications based on the status of agents, servers (cluster nodes), and connectors. See Sending Notifications on Opswise Component Status.

Attaching an Email Notification to a Task

1. Display the task that will generate the Email notification.
2. Click on the **Actions** tab, which displays a list of all defined actions, if any.
3. Click **New**. The Actions options displays:



4. Select **Email Notification**.



Action Inheritance: SELF

5. Using the field descriptions provided below as a guide, complete the fields as needed.
6. Click the **Submit** button to save the record and return to the Email notifications list, or, right-click and select **Save** to save the record and remain on the current display.
7. If appropriate, repeat these steps for any additional Email notifications you want to add.

Email Notifications Field Descriptions

The table below describes the fields and buttons on the Email notifications screen.

Field Name	Description
Action Inheritance	Workflow tasks only. Specifies what records these instructions apply to. Options: • SELF - These instructions apply only to the workflow and are not inherited by its children tasks. • SELF/CHILDREN - These instructions apply to the workflow and its contained tasks (children). • CHILDREN - These instructions apply only to the tasks within the workflow (children).

<table>
<tr>
<td>Status</td>
<td>The status of this task. To trigger an abort action, you can specify status only, or status and exit code. You can specify as many statuses as needed. Options:
<table>
<tr><th>Status</th><th>Description</th></tr>
<tr><td><b>Defined</b></td><td>All task types. The new task instance has been created (the task has been launched). Not yet implemented.</td></tr>
<tr><td><b>Waiting</b></td><td>All task types. The task has been loaded by a workflow and is waiting to run.</td></tr>
<tr><td><b>Held</b></td><td>All task types. The task has been put on hold by a user.</td></tr>
<tr><td><b>Resource Requested</b></td><td>All tasks with a virtual resource defined go immediately to a status of Resource Requested. If the resource is available, the task then moves to the next appropriate processing status.</td></tr>
<tr><td><b>Resource Wait</b></td><td>All tasks with a virtual resource defined go immediately to a status of Resource Requested. If the resource is not available, the task goes to a status of Resource Wait. When the resource becomes available, the task moves to the next appropriate processing status</td></tr>
<tr><td><b>Undeliverable</b></td><td>Agent-based tasks. The agent is unavailable.</td></tr>
<tr><td><b>Queued</b></td><td>Agent-based tasks only. The task has been queued on a resource.</td></tr>
<tr><td><b>Submitted</b></td><td>z/OS only. The task has been submitted to the z/OS Job Entry subsystem and scheduled by the z/OS Job Scheduler.</td></tr>
<tr><td><b>Action Required</b></td><td>Manual tasks only. When a manual task launches, it goes into Action Required status, meaning a user must perform some manual activity. For details, see Manual task.</td></tr>
<tr><td><b>Started</b></td><td>Agent-based and Manual tasks only. The task has started. For agent-based tasks, this means the agent has received the task.</td></tr>
<tr><td><b>Running</b></td><td>All task types. The task is running. For agent-based tasks, the agent has started running the program.</td></tr>
<tr><td><b>Running Problems</b></td><td>Workflows only. One or more tasks within the workflow has one of the following statuses:
• Held
• Undeliverable
• Running Problems (for sub-workflows)
• Cancel Pending
• In Doubt
• Start Failure
• Cancelled</td></tr>
<tr><td><b>In Doubt</b></td><td>Agent-based tasks only. The agent is "in doubt" about the current status of the task instance. This may occur if an agent or agent connection goes down. In this case, the agent restarts and reviews its data about tasks in progress. If the agent finds a task still running, it resumes normal monitoring. If the agent cannot find the task, this usually indicates that the task completed, but the agent considers the task status to be "in doubt."</td></tr>
<tr><td><b>Start Failure</b></td><td>All task types. The task was unable to start.</td></tr>
<tr><td><b>Confirmation Required</b></td><td>z/OS only. If you make JCL changes and restart a z/OS task, Opswise will put the task into Confirmation Required status and prompt you for a confirmation. For detailed processing steps, see Rerunning a z/OS Task .</td></tr>
<tr><td><b>Cancelled</b></td><td>All task types. The task was cancelled by a user.</td></tr>
<tr><td><b>Failed</b></td><td>All task types. The task ran to a failure status.</td></tr>
<tr><td><b>Skipped</b></td><td>All task types. The task was skipped by a user.</td></tr>
<tr><td><b>Finished</b></td><td>All task types. The task was forced by the user to finish. The user may do this in cases where the task had "Cancelled" or "Failed" status, and the user needed to release other task instances depending on the successful completion of this task instance in a workflow. For more information, see Force Finishing a Task.</td></tr>
<tr><td><b>Success</b></td><td>All task types. The task has completed successfully.</td></tr>
</table>
</td>
</tr>
<tr>
<td>Exit Codes</td>
<td>Specifies one or more exit codes that will trigger the event. If you specify an exit code, you must also specify at least one status. Use commas to separate multiple exit codes; use a hyphen to specify a range. Example: 1, 5, 22-30.</td>
</tr>
<tr>
<td>On Late Start</td>
<td>Generate the action or notification if the task started late, based on the Late Start Time specified in the task.</td>
</tr>
</table>

On Late Finish	Generate the action or notification if the task finishes late, based on the Late Finish time specified in the task.
On Early Finish	Generate the action or notification if the task finishes early, based on the Early Finish Time specified in the task.
Email Template	Optional. The name of the Email template defined using the Email template screen. The Email template allows you to specify standard recipients and text for outgoing emails. Type in a name, or click the magnifying glass to browse to an existing Email template or create a new one. You must specify either an Email template or Email connection, or both. If you specify both, the Email server specified in the Email Connection record overrides the server in the template.
Email Connection	Required. The name of the Email connection defined using the Email connection screen. The email connection specifies information about the email server. You can also specify the Email connection in the Email template (see above). You must specify either an Email template or Email connection, or both. If you specify an Email template and an Email connection, the server selected in the Email connection overrides the server selected in the Email template. Type in a name, or click the magnifying glass to browse to an existing Email server definition or create a new one.
Reply-To	Required. Specifies the email address of the sender. Use commas to separate multiple recipients. Variables supported.
To	Required. Specifies the email address of the recipient. Use commas to separate multiple recipients. Variables supported.
CC	Optional. Specifies the email address of the party being sent a copy of the email, if any. Use commas to separate multiple recipients. Variables supported.
BCC	Optional. Specifies the email address of the party being sent a blind (hidden) copy of the email, if any. Use commas to separate multiple recipients. Variables supported.
Subject	Optional. Specifies the subject line of the email. Variables supported.
Body	Optional. Contains the text of the email message. Variables supported. If both the email template and the email task contain text in the body, the text is appended.
Attach Standard Output	Attach any standard output generated by the associated task.
Attach Standard Error	Attach standard error data generated by the associated task.
Attach File	Attach any text file that is accessible by the agent.
Start Line	Attach data beginning at the line indicated.
Number of Lines	Optional. Allows you to limit the retrieved data to the number of lines specified. The default is 100. You can change the default by modifying the record **Configuration > Properties > Retrieve Output Default Maximum Lines**.
Scan Text:	Optional. If Type = Create, Change or Exists, this field specifies a string that the monitor will search for in the file or files. Specifying a string means that only files containing the string constitute a match. Opswise will process this field as a regular expression.

File Name	For Attach File only, the path and filename of the file you want to attach to the email notification.
Submit button	Submits the new record to the database.
Update button	Saves updates to the record.
Delete button	Deletes the current record.

Creating SNMP Notifications

Overview

You can attach one or more SNMP notifications to any Opswise task. For Workflow tasks, you can also specify whether you want the SNMP notification to be triggered by the workflow itself, the workflow and/or its tasks, or by the tasks only.

You can trigger the notification based on one or more of the following events associated with the task instance to which you attach the notification:

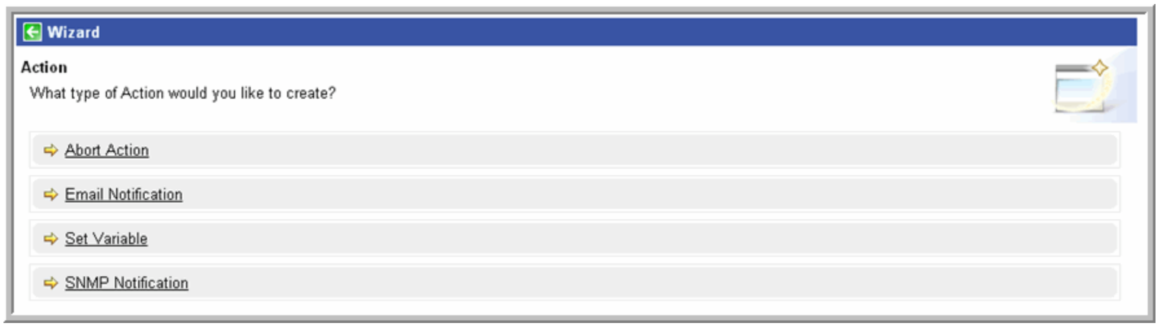
- Status or statuses of the task instance
- Exit code(s) generated by the program (along with at least one status)
- Late start
- Early or late finish

In order to generate SNMP notifications, you need an SNMP Manager definition, which provides the SNMP server name and other pertinent information.

You can also generate notifications based on the status of agents, agent clusters, and connectors. See Sending Notifications on Opswise Component Status.

Attaching an SNMP Notification to a Task

1. Display the task that will generate the SNMP notification.
2. Click on the **Actions** tab, which displays a list of all defined actions, if any.
3. Click **New**. The Actions options displays:

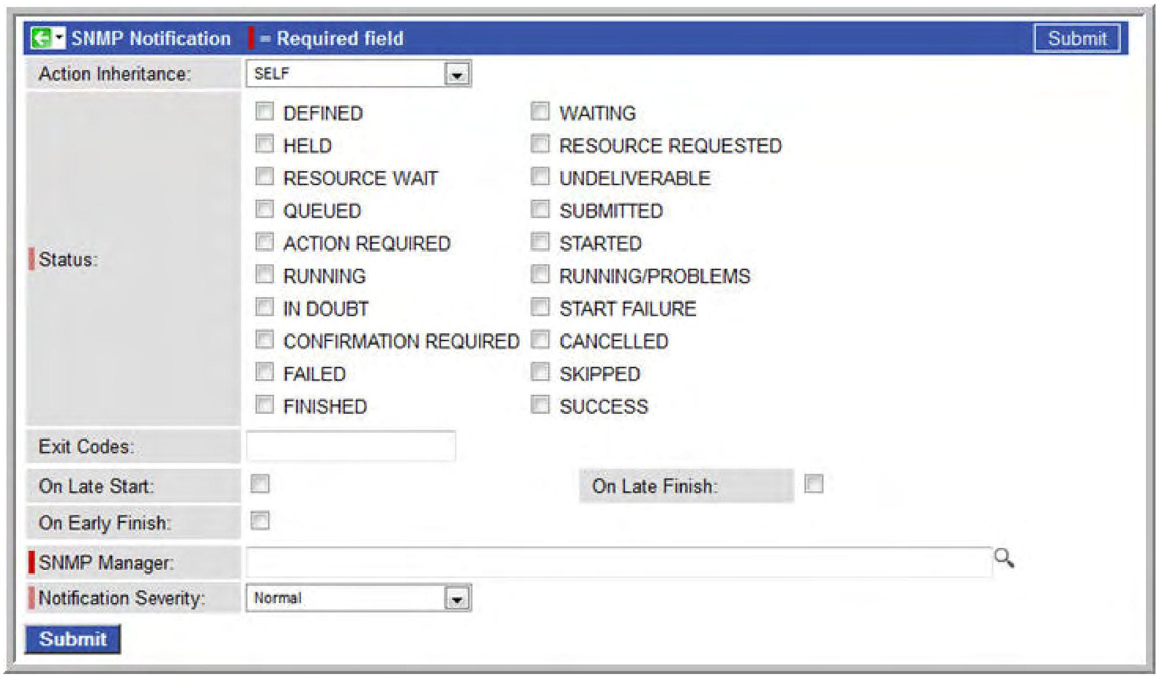


Wizard

Action

What type of Action would you like to create?

4. Select **SNMP Notification**.



SNMP Notification = Required field Submit

Action Inheritance:	SELF	
Status:	□ DEFINED	□ WAITING
	□ HELD	□ RESOURCE REQUESTED
	□ RESOURCE WAIT	□ UNDELIVERABLE
	□ QUEUED	□ SUBMITTED
	□ ACTION REQUIRED	□ STARTED
	□ RUNNING	□ RUNNING/PROBLEMS
	□ IN DOUBT	□ START FAILURE
	□ CONFIRMATION REQUIRED	□ CANCELLED
	□ FAILED	□ SKIPPED
	□ FINISHED	□ SUCCESS
Exit Codes:		
On Late Start:	□	On Late Finish: □
On Early Finish:	□	
SNMP Manager:		
Notification Severity:	Normal	

Submit

5. Using the field descriptions provided below as a guide, complete the fields as needed.
6. Click the **Submit** button to save the record and return to the SNMP notifications list, or, right-click and select **Save** to save the record and remain on the current display.
7. If appropriate, repeat these steps for any additional SNMP notifications you want to add.

SNMP Notifications Field Descriptions

The table below describes the fields and buttons on the SNMP notifications screen.

Field Name	Description
Action Inheritance	Workflow tasks only. Specifies what records these instructions apply to. Options: • SELF - These instructions apply only to the workflow and are not inherited by its children tasks. • SELF/CHILDREN - These instructions apply to the workflow and its contained tasks (children). • CHILDREN - These instructions apply only to the tasks within the workflow (children).

<table>
<tr><td>Status</td><td>The status of this task. To trigger a notification you can specify status only, or status and exit code. You can specify as many statuses as needed. Options:
<table>
<tr><th>Status</th><th>Description</th></tr>
<tr><td><b>Defined</b></td><td>All task types. The new task instance has been created (the task has been launched). Not yet implemented.</td></tr>
<tr><td><b>Waiting</b></td><td>All task types. The task has been loaded by a workflow and is waiting to run.</td></tr>
<tr><td><b>Held</b></td><td>All task types. The task has been put on hold by a user.</td></tr>
<tr><td><b>Resource Requested</b></td><td>All tasks with a virtual resource defined go immediately to a status of Resource Requested. If the resource is available, the task then moves to the next appropriate processing status.</td></tr>
<tr><td><b>Resource Wait</b></td><td>All tasks with a virtual resource defined go immediately to a status of Resource Requested. If the resource is not available, the task goes to a status of Resource Wait. When the resource becomes available, the task moves to the next appropriate processing status</td></tr>
<tr><td><b>Undeliverable</b></td><td>Agent-based tasks. The agent is unavailable.</td></tr>
<tr><td><b>Queued</b></td><td>Agent-based tasks only. The task has been queued on a resource.</td></tr>
<tr><td><b>Submitted</b></td><td>z/OS only. The task has been submitted to the z/OS Job Entry subsystem and scheduled by the z/OS Job Scheduler.</td></tr>
<tr><td><b>Action Required</b></td><td>Manual tasks only. When a manual task launches, it goes into Action Required status, meaning a user must perform some manual activity. For details, see Manual task.</td></tr>
<tr><td><b>Started</b></td><td>Agent-based and Manual tasks only. The task has started. For agent-based tasks, this means the agent has received the task.</td></tr>
<tr><td><b>Running</b></td><td>All task types. The task is running. For agent-based tasks, the agent has started running the program.</td></tr>
<tr><td><b>Running Problems</b></td><td>Workflows only. One or more tasks within the workflow has one of the following statuses:
• Held
• Undeliverable
• Running Problems (for sub-workflows)
• Cancel Pending
• In Doubt
• Start Failure
• Cancelled</td></tr>
<tr><td><b>In Doubt</b></td><td>Agent-based tasks only. The agent is "in doubt" about the current status of the task instance. This may occur if an agent or agent connection goes down. In this case, the agent restarts and reviews its data about tasks in progress. If the agent finds a task still running, it resumes normal monitoring. If the agent cannot find the task, this usually indicates that the task completed, but the agent considers the task status to be "in doubt."</td></tr>
<tr><td><b>Start Failure</b></td><td>All task types. The task was unable to start.</td></tr>
<tr><td><b>Confirmation Required</b></td><td>z/OS only. If you make JCL changes and restart a z/OS task, Opswise will put the task into Confirmation Required status and prompt you for a confirmation. For detailed processing steps, see Rerunning a z/OS Task .</td></tr>
<tr><td><b>Cancelled</b></td><td>All task types. The task was cancelled by a user.</td></tr>
<tr><td><b>Failed</b></td><td>All task types. The task ran to a failure status.</td></tr>
<tr><td><b>Skipped</b></td><td>All task types. The task was skipped by a user.</td></tr>
<tr><td><b>Finished</b></td><td>All task types. The task was forced by the user to finish. The user may do this in cases where the task had "Cancelled" or "Failed" status, and the user needed to release other task instances depending on the successful completion of this task instance in a workflow. For more information, see Force Finishing a Task.</td></tr>
<tr><td><b>Success</b></td><td>All task types. The task has completed successfully.</td></tr>
</table>
</td></tr>
<tr><td>Exit Codes</td><td>Specifies one or more exit codes that will trigger the event. If you specify an exit code, you must also specify at least one status. Use commas to separate multiple exit codes; use a hyphen to specify a range. Example: 1, 5, 22-30.</td></tr>
<tr><td>On Late Start</td><td>Generate the action or notification if the task started late, based on the Late Start Time specified in the task.</td></tr>
</table>

On Late Finish	Generate the action or notification if the task finishes late, based on the Late Finish time specified in the task.
On Early Finish	Generate the action or notification if the task finishes early, based on the Early Finish Time specified in the task.
SNMP Manager	The SNMP Manager that will be used to generate the notification.
Notification Severity	Optional. Informational only. Indicates the severity of this notification. Options: Normal, Warning, Minor, Major Critical.
Submit button	Submits the new record to the database.
Update button	Saves updates to the record.
Delete button	Deletes the current record.

Setting Up Abort Actions

Overview

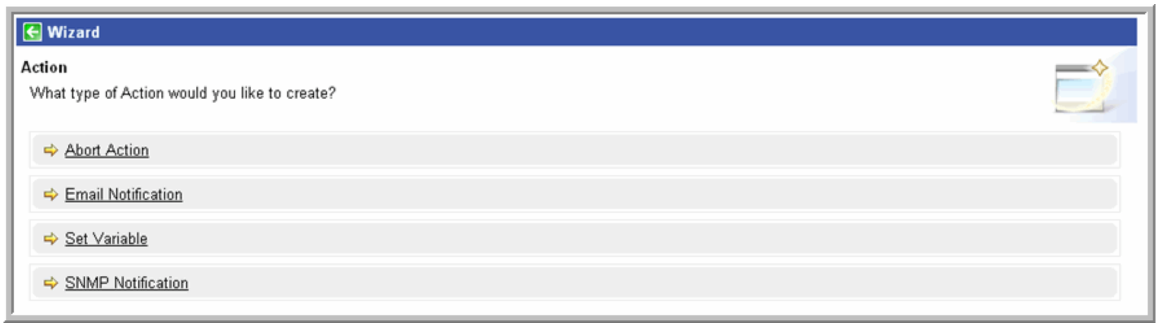
The Abort Action allows you to abort a running process and/or overwrite the exit code generated by a process, based on specified conditions. You can trigger the action based on one or more of the following events associated with the task instance:

- Status or statuses of the task instance
- Exit code(s) generated by the program (along with at least one status)
- Late start
- Early or late finish

You can attach one or more Abort Actions to any Opswise task. For Workflow tasks, you can also specify whether you want the Abort Action instructions to apply to the workflow itself, the workflow an/or its tasks, or to the tasks only.

Attaching an Abort Action to a Task

1. Display the task to which you are attaching the Abort Action.
2. Click on the **Actions** tab. This tab displays a list of all previously entered Actions for this task, including Abort Actions, Notifications, and Set Variables.
3. Click **New**. The Action wizard displays, as shown below.

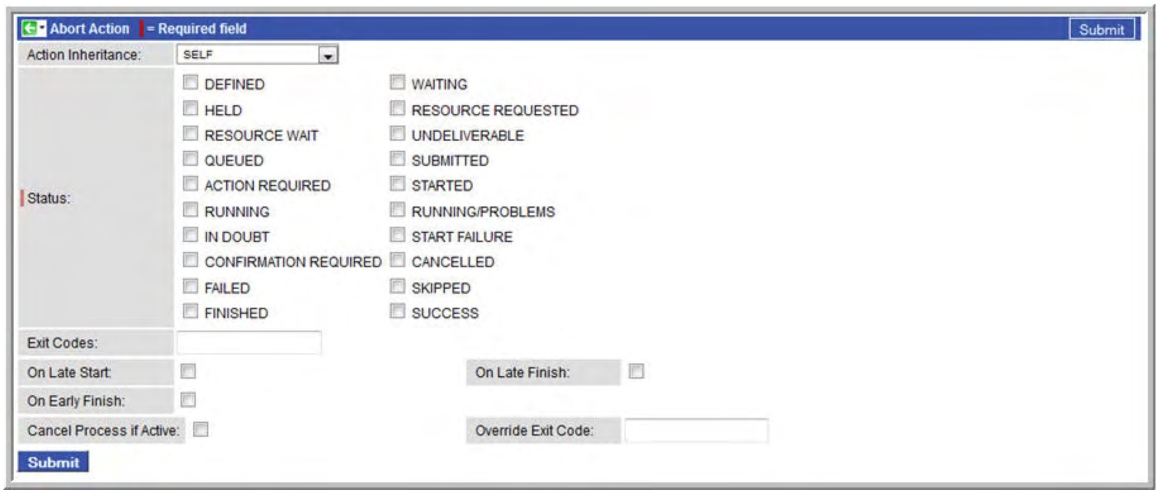


Wizard

Action

What type of Action would you like to create?

4. Click **Abort Action**. The Abort Action screen displays.



Abort Action = Required field

Action Inheritance: SELF

Status:

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

Exit Codes:

On Late Start: □ On Late Finish: □

On Early Finish: □

Cancel Process if Active: □ Override Exit Code:

5. Using the field descriptions provided below as a guide, complete the fields as needed.
6. Click the **Submit** button to save the record and return to the Actions list, or, right-click and select **Save** to save the record and remain on the current display.

7. If appropriate, repeat these steps for any additional Actions you want to add.

Abort Actions Field Descriptions

The table below describes the fields and buttons on the Abort Actions screen.

Field Name	Description
Action Inheritance	Workflow tasks only. Specifies what records these instructions apply to. Options: • SELF - These instructions apply only to the workflow and are not inherited by its children tasks. • SELF/CHILDREN - These instructions apply to the workflow and its contained tasks (children). • CHILDREN - These instructions apply only to the tasks within the workflow (children).

<table>
<tr><td>Status</td><td>The status of this task. To trigger an abort action, you can specify status only, or status and exit code. You can specify as many statuses as needed. Options:
<table>
<tr><th>Status</th><th>Description</th></tr>
<tr><td><b>Defined</b></td><td>All task types. The new task instance has been created (the task has been launched). Not yet implemented.</td></tr>
<tr><td><b>Waiting</b></td><td>All task types. The task has been loaded by a workflow and is waiting to run.</td></tr>
<tr><td><b>Held</b></td><td>All task types. The task has been put on hold by a user.</td></tr>
<tr><td><b>Resource Requested</b></td><td>All tasks with a virtual resource defined go immediately to a status of Resource Requested. If the resource is available, the task then moves to the next appropriate processing status.</td></tr>
<tr><td><b>Resource Wait</b></td><td>All tasks with a virtual resource defined go immediately to a status of Resource Requested. If the resource is not available, the task goes to a status of Resource Wait. When the resource becomes available, the task moves to the next appropriate processing status</td></tr>
<tr><td><b>Undeliverable</b></td><td>Agent-based tasks. The agent is unavailable.</td></tr>
<tr><td><b>Queued</b></td><td>Agent-based tasks only. The task has been queued on a resource.</td></tr>
<tr><td><b>Submitted</b></td><td>z/OS only. The task has been submitted to the z/OS Job Entry subsystem and scheduled by the z/OS Job Scheduler.</td></tr>
<tr><td><b>Action Required</b></td><td>Manual tasks only. When a manual task launches, it goes into Action Required status, meaning a user must perform some manual activity. For details, see Manual task.</td></tr>
<tr><td><b>Started</b></td><td>Agent-based and Manual tasks only. The task has started. For agent-based tasks, this means the agent has received the task.</td></tr>
<tr><td><b>Running</b></td><td>All task types. The task is running. For agent-based tasks, the agent has started running the program.</td></tr>
<tr><td><b>Running Problems</b></td><td>Workflows only. One or more tasks within the workflow has one of the following statuses:
• Held
• Undeliverable
• Running Problems (for sub-workflows)
• Cancel Pending
• In Doubt
• Start Failure
• Cancelled</td></tr>
<tr><td><b>In Doubt</b></td><td>Agent-based tasks only. The agent is "in doubt" about the current status of the task instance. This may occur if an agent or agent connection goes down. In this case, the agent restarts and reviews its data about tasks in progress. If the agent finds a task still running, it resumes normal monitoring. If the agent cannot find the task, this usually indicates that the task completed, but the agent considers the task status to be "in doubt."</td></tr>
<tr><td><b>Start Failure</b></td><td>All task types. The task was unable to start.</td></tr>
<tr><td><b>Confirmation Required</b></td><td>z/OS only. If you make JCL changes and restart a z/OS task, Opswise will put the task into Confirmation Required status and prompt you for a confirmation. For detailed processing steps, see Rerunning a z/OS Task .</td></tr>
<tr><td><b>Cancelled</b></td><td>All task types. The task was cancelled by a user.</td></tr>
<tr><td><b>Failed</b></td><td>All task types. The task ran to a failure status.</td></tr>
<tr><td><b>Skipped</b></td><td>All task types. The task was skipped by a user.</td></tr>
<tr><td><b>Finished</b></td><td>All task types. The task was forced by the user to finish. The user may do this in cases where the task had "Cancelled" or "Failed" status, and the user needed to release other task instances depending on the successful completion of this task instance in a workflow. For more information, see Force Finishing a Task.</td></tr>
<tr><td><b>Success</b></td><td>All task types. The task has completed successfully.</td></tr>
</table>
</td></tr>
<tr><td>Exit Codes</td><td>Specifies one or more exit codes that will trigger the event. If you specify an exit code, you must also specify at least one status. Use commas to separate multiple exit codes; use a hyphen to specify a range. Example: 1, 5, 22-30.</td></tr>
<tr><td>On Late Start</td><td>Generate the action or notification if the task started late, based on the Late Start Time specified in the task.</td></tr>
</table>

On Late Finish	Generate the action or notification if the task finishes late, based on the Late Finish time specified in the task.
On Early Finish	Generate the action or notification if the task finishes early, based on the Early Finish Time specified in the task.
Cancel Process if Active	If enabled, instructs the system to cancel the process that was launched by this task.
Override Exit Code	Overrides the exit code returned by the process with the exit code specified in this field.
Submit button	Submits the new record to the database.
Update button	Saves updates to the record.
Delete button	Deletes the current record.

Set Variable Action

See Setting Variables in a Workflow.

Creating Step Conditions

Overview

Step condition checking provides a dynamic means for a person or system automation to guide the execution of a z/OS job based on individual step condition codes in that job. For example, if a multi-step job has a step that ends with a condition code of 8, you could include a step condition check to decide whether or not to run the following step(s).

Step condition checking can be performed automatically or by manual intervention via communication with an operator.

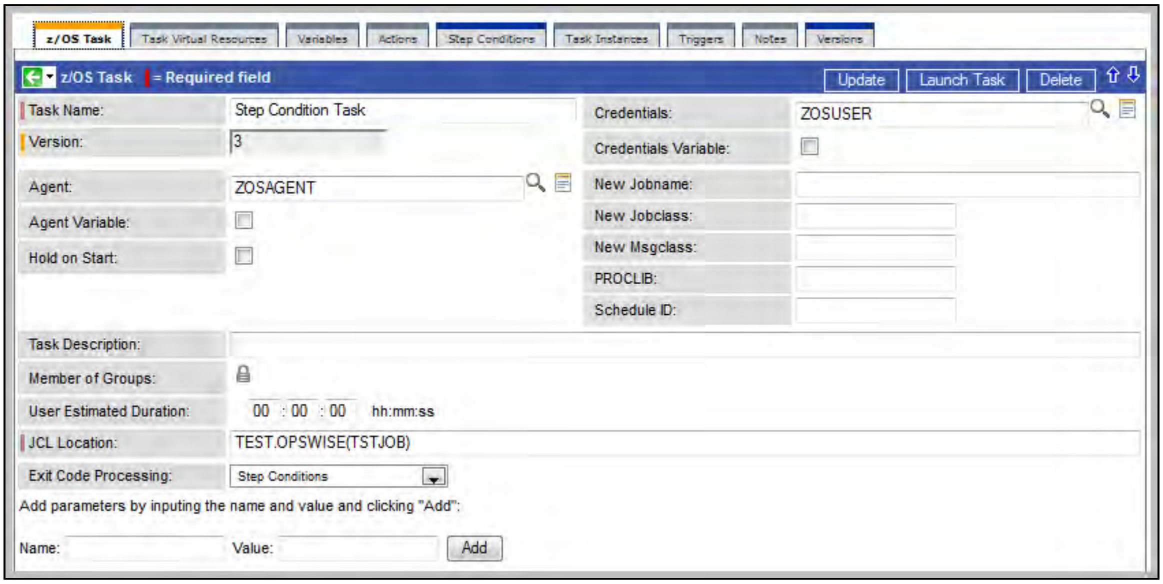
The z/OS Task definition lets you specify any number of steps, condition codes and/or ranges, and actions for a job. You also can specify step conditions at the workflow level that apply to all z/OS tasks in that workflow and child workflows.

Runtime Monitoring

You can monitor step conditions at runtime via the Activity screen, which lets you add or change step conditions for a single task instance and then re-run that job.

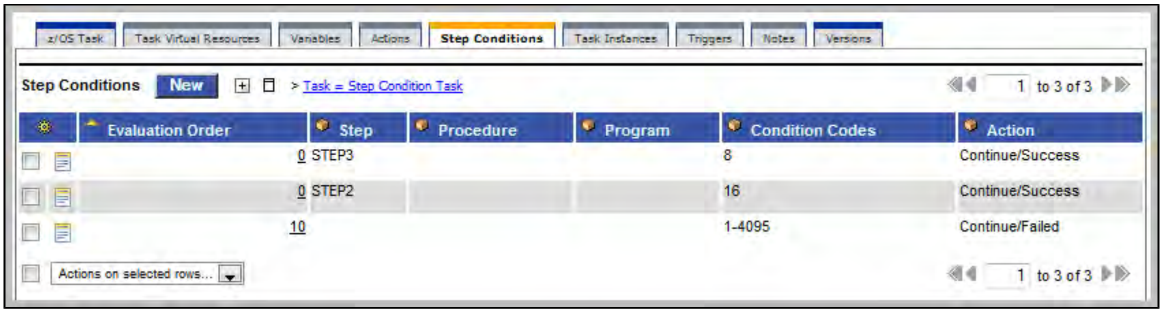
Creating a Step Condition

1. From the Navigation Pane, select **z/OS Tasks**. The z/OS Tasks List screen displays.
2. Select the task for which you want to create one or more step conditions. The z/OS Task screen displays.
3. In the Exit Code Processing field, select Step Conditions.



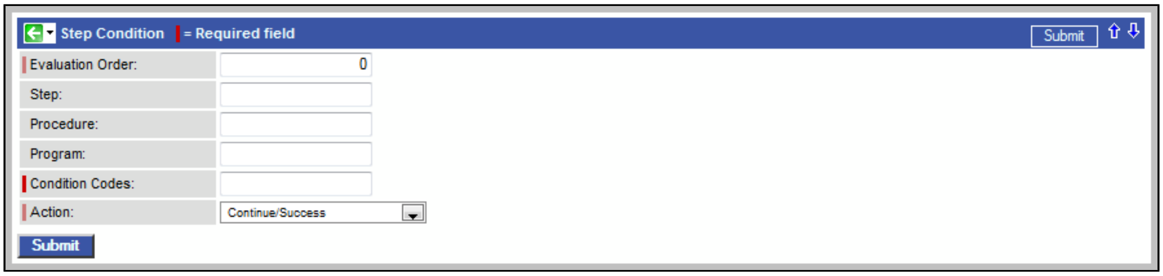
Task Name: Step Condition Task
Version: 3
Agent: ZOSAGENT
Agent Variable: □
Hold on Start: □
Credentials: ZOSUSER
Credentials Variable: □
New Jobname:
New Jobclass:
New Msgclass:
PROCLIB:
Schedule ID:
Task Description:
Member of Groups:
User Estimated Duration: 00 : 00 : 00 hh:mm:ss
JCL Location: TEST.OPSWISE(TSTJOB)
Exit Code Processing: Step Conditions
Add parameters by inputing the name and value and clicking "Add":
Name: Value: Add

4. Click the **Step Conditions** tab. The Step Conditions List screen displays.



	Evaluation Order	Step	Procedure	Program	Condition Codes	Action
□	0	STEP3			8	Continue/Success
□	0	STEP2			16	Continue/Success
□	10				1-4095	Continue/Failed

5. Click **New**. The Step Conditions screen displays.



Evaluation Order:	0
Step:	
Procedure:	
Program:	
Condition Codes:	
Action:	Continue/Success

6. Using the field descriptions provided below as a guide, complete the fields as needed.
7. Click the **Submit** button to save the record and return to the menu, or right-click on the title bar and select **Save** to save the record and remain on the current display.
8. If appropriate, repeat these steps for any additional step conditions you want to add.

Step Condition Field Descriptions

The table below describes the fields and buttons on the Step Conditions screen.

Field Name	Description
Evaluation Order	Order for the step condition evaluation. Step conditions with a lower order are evaluated before step conditions with a higher order.
Step	Step name for this step. If no step is specified, the step condition matches any step.
Procedure	Procedure step name. If no procedure is specified, the step condition matches any procedure.
Program	Program name. If no program is specified, the step condition matches any program.
Condition Codes	Condition codes in this format: • 1 • 6-4095 • Sxxx • Unnnn Condition codes can be entered as a comma-separated list (for example: 1,2,5-8).
Action	Step action and, where applicable, the impact the action will have on the task instance status (see Step Condition Logic).

Note
The Step, Procedure, and Program fields allow the use of wildcards (***** and **?**).

Step Condition Logic

When a task is run, and a step completes, Opswise first checks all step conditions in the specified order. If no match is found, Opswise then checks the task's Workflow, as applicable, for step conditions. If no match is found, Opswise then checks the Workflow's parent Workflow, as applicable, and so on to the highest Workflow level. The evaluation order of Workflow step conditions is independent of task-level step conditions.

Based on testing the step condition code against discreet values or a range of values, the Opswise z/OS agent will take one of the following actions for a matching step condition:

Continue/Success	Job/task will CONTINUE processing and the status will be marked as SUCCESS.
Continue/Failed	Job/task will CONTINUE processing and the status will be marked as FAILED.
Halt/Failed	Job/task will be HALTED at the current step and the status will be marked as FAILED.
Askoper	Opswise sends a WTOR message to the console operator requesting a reply on how job execution should proceed. System action is dependent upon the operator reply (see Example 4, below).

Example Steps and Condition Codes

This section provides a sample job and PROC, followed by example condition code checks for that job.

Example Job and PROC

Example Job

```
//JOBA JOB ...
//S1 EXEC ACCTBL10
```

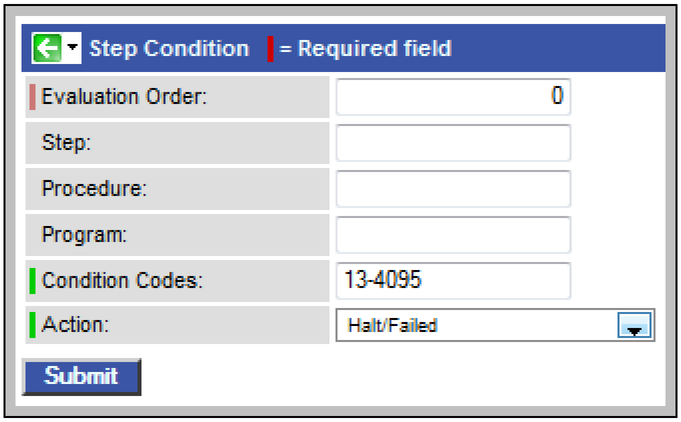
Example PROC (Cataloged Procedure)

```
//ACCTBL10 PROC
//STEP1 EXEC PGM=BALANCE
//STEP2 EXEC PGM=MERGE
//STEP3 EXEC PGM=IEBGENER
//STEP4 EXEC ACCTRP01

//ACCTRP01 PROC
//STEP1 EXEC PGM=REPORT
//STEP2 EXEC PGM=IEBGENER
```

Opswise User Interface Specifications and Actions

The following examples specify condition code checks for the example job above.

Example 1

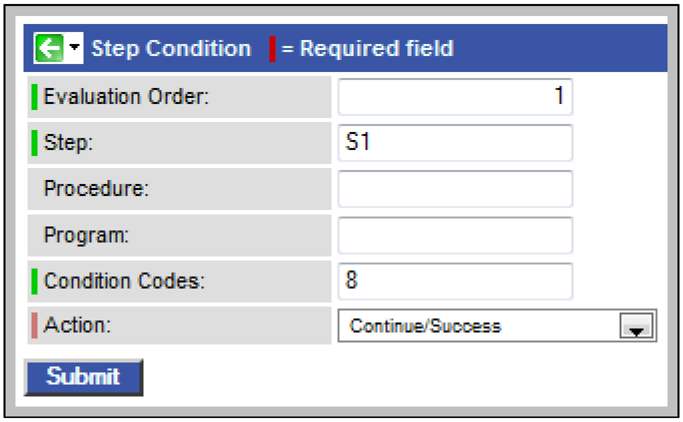


Step Condition = Required field

Evaluation Order:	0
Step:	
Procedure:	
Program:	
Condition Codes:	13-4095
Action:	Halt/Failed

In this example, if the condition code of any step of the job is greater than 12, the job halts and is marked FAILED.

Example 2

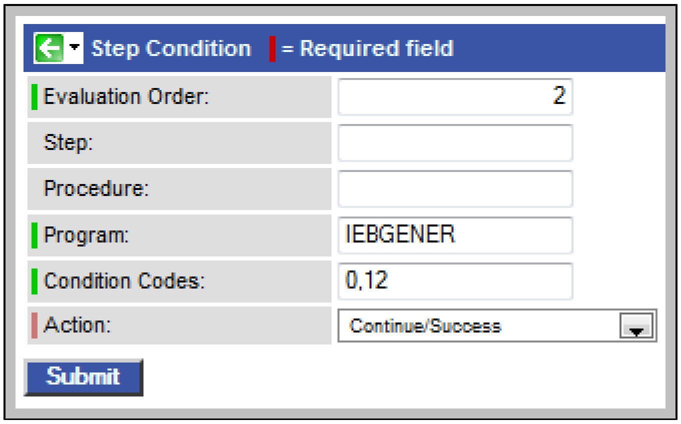


Step Condition = Required field

Evaluation Order:	1
Step:	S1
Procedure:	
Program:	
Condition Codes:	8
Action:	Continue/Success

In this example, if the condition code of step S1 of the job is equal to 8, the job continues and is marked SUCCESS.

Example 3

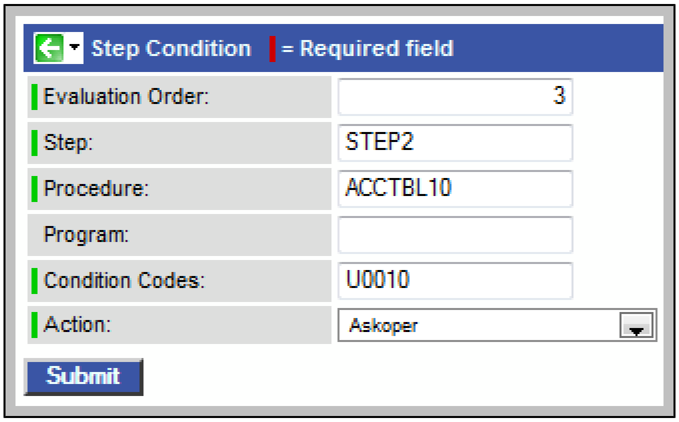


Step Condition = Required field

Evaluation Order:	2
Step:	
Procedure:	
Program:	IEBGENER
Condition Codes:	0,12
Action:	Continue/Success

In this example, if the condition code of either IEBGENER program is 0 or 12, the job continues and is marked SUCCESS.

Example 4



Evaluation Order:	3
Step:	STEP2
Procedure:	ACCTBL10
Program:	
Condition Codes:	U0010
Action:	Askoper

In this example, if the condition code from STEP2 in cataloged procedure ACCTBL10 is U0010, the operator is alerted via a WTOR that specifies the job, the step, the cataloged procedure, and the actual condition code.

Issued WTOR

```
UAG100nnnA Job: JOBA Step: STEP2 Proc: ACCTBL10 Code: U0010
has triggered condition code action.
 Reply (1) CONTINUE/SUCCESS, (2) CONTINUE/FAILED, (3) HALT/FAILED
```

In this WTOR:

- Job: Name of job requiring action.
- Step: Name of step within the job.
- Code: Condition code returned by the step.

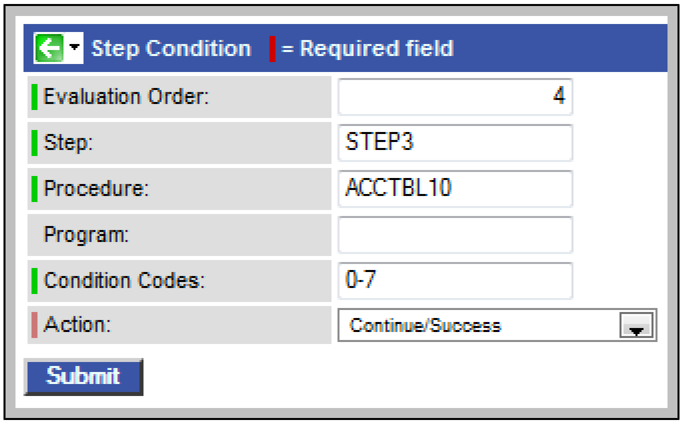
Operator Reply

The operator replies with one of the following:

- (1) CONTINUE/SUCCESS
- (2) CONTINUE/FAILED
- (3) HALT/FAILED

(See Step Condition Logic for an explanation of these replies.)

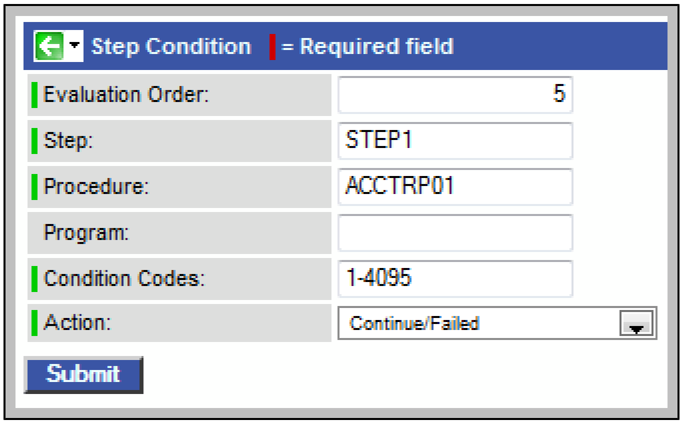
Example 5



Evaluation Order:	4
Step:	STEP3
Procedure:	ACCTBL10
Program:	
Condition Codes:	0-7
Action:	Continue/Success

In this example, if the condition code from STEP3 in cataloged procedure ACCTBL10 is within the range of 0-7, the job continues and is marked SUCCESS.

Example 6



Evaluation Order:	5
Step:	STEP1
Procedure:	ACCTRP01
Program:	
Condition Codes:	1-4095
Action:	Continue/Failed

In this example, if the condition code from STEP1 in cataloged procedure ACCTRP01 is greater than 0, the job continues and is marked FAILED.

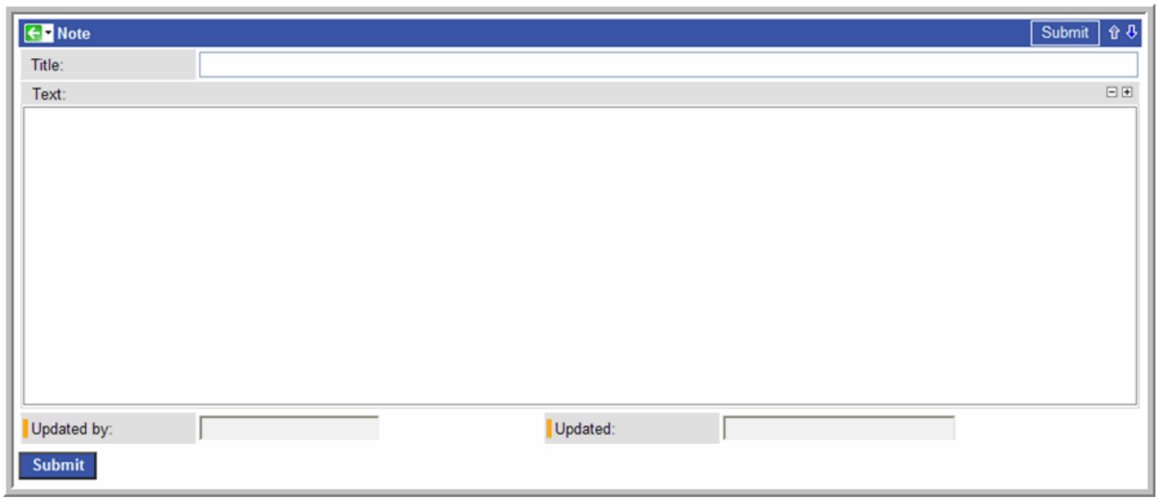
Creating Runbook Notes

Introduction

You can enter a note for any record in the Opswise system, including tasks, triggers, resources, calendars, and variables. The note may consist of information needed by operations personnel or other instructions or tips.

Adding a Note

1. Open the record to which you want to attach the note
2. Click on the **Notes** tab.
3. Click the **New** button. The Notes form displays, as shown below.



4. Type in the Title and Text.
5. Click the **Submit** button.

Note Field Descriptions

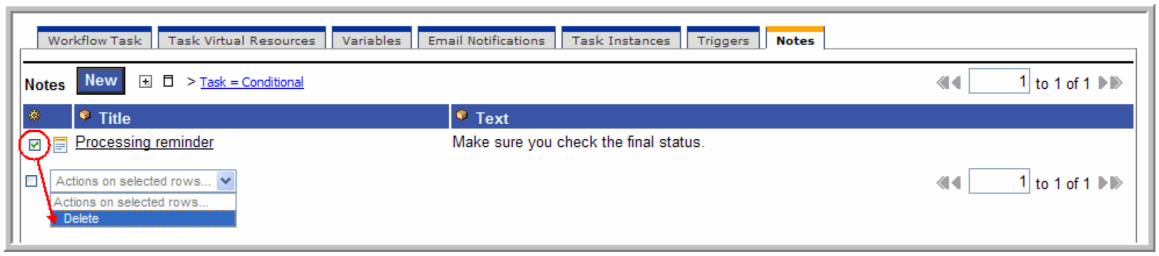
Field Name	Description
Title	Title of this note. Displays in the Title column on the Notes list.
Text	Text of the note.
Updated by	The user who last updated this record.
Updated	Date and time this record was last updated.

Submit button	Submits the new record to the database.
Update button	Saves updates to the record.
Delete button	Deletes the current record.

Deleting a Note

Display the note you want to delete and click the **Delete** button. Or, you can delete one or more notes as follows:

1. From the Notes list, click on the box associated with the note or notes you want to delete.



	Title	Text
☑	Processing reminder	Make sure you check the final status.

2. From the **Actions on selected rows** menu, select **Delete**.

Viewing a Note

1. From the Notes list, scroll to the note you want to read.
2. Click the underlined field displayed in the leftmost column. The system displays the contents of the note.

Creating Workflows

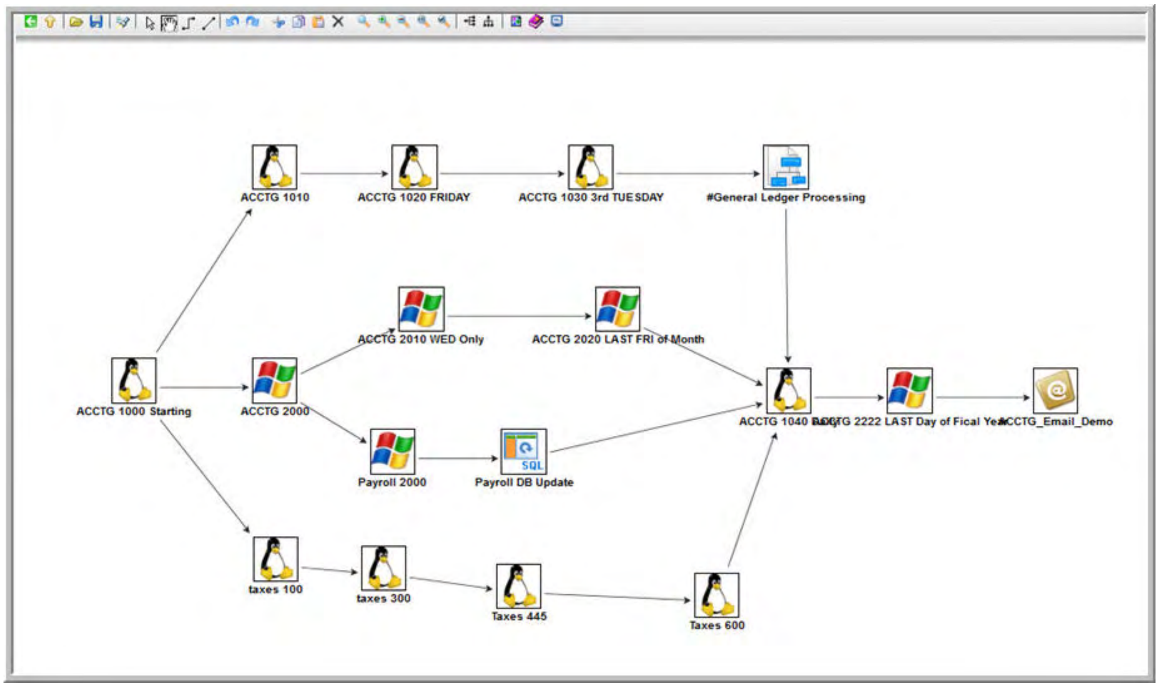
Overview

The Workflow definition tool is a graphical tool that allows you to select tasks, position them within a workflow and specify the dependency relationships between them.

The process of creating a workflow involves:

1. Creating a new workflow task.
2. Accessing the Workflow Editor and:
   - Opening the **Add Task** window.
   - Specifying parameters for fetching tasks.
   - Dragging tasks onto the canvas.
   - Adding connections and dependencies between the tasks.
   - Defining the layout of the workflow.
3. Saving the new workflow.

Workflows can be as simple or elaborate as necessary. Zooming and scrolling features are provided that allow you to work on small areas of a very large workflow, or to create simple ad-hoc workflows. Shown below is a sample workflow consisting of a variety of task types. There are no restrictions on the types of tasks that can be included as part of a single workflow and you can also put workflows within other workflows.



Use the icon reference for a quick guide to the tools available for building workflows; for more detailed instructions, see Defining a Workflow.

Rules for Creating Workflows

- You can include as many workflows within a workflow as needed.
- A single workflow can support multiple starting tasks. All starting tasks launch when the workflow launches.
- A workflow can include a task that is not connected to any other task. All disconnected tasks launch when the workflow launches.

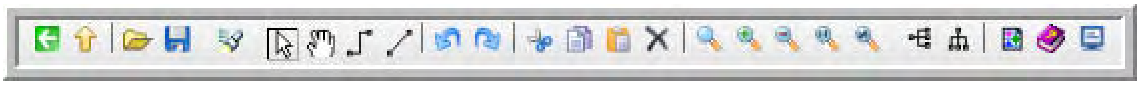
Workflow Modes

The actions you can perform within the Workflow Editor sometimes depend on what mode you are in. The mode is set by clicking on an icon in the toolbar. The Workflow Editor has three modes:

- **Connect** — Clicking on a **Connector** icon (⌐ or ⁄) puts the Workflow Editor into **Connect** mode. You must be in **Connect** mode to create connections. You can also move objects around in **Connect** mode.
- **Select** — Clicking on the **Select** icon ↖ puts the Workflow Editor into **Select** mode. You must be in **Select** or **Connect** mode to move objects around.
- **Pan** — Clicking on the **Pan** icon ✋ puts the Workflow Editor into **Pan** mode, which allows you to scroll around on the workflow. You can also use the **Outline** icon ▣ to scroll around.

The remaining operations, including **Save, Preview, Undo, Redo, Cut, Copy, Paste, Delete, Add Task, Zoom, Fit, Layout, Help**, and **Console** can be performed in any mode.

A shadowed outline around the icon indicates what mode the Workflow Editor is in. The following example shows that the Workflow Editor is in **Select** mode.



Icon Reference

The following table describes the icons used to define workflows.

Icon	Description
←	**Back** --- Goes back to the previous screen.
	Go to Parent --- If this workflow is embedded in another workflow, navigates to the parent. If this workflow has

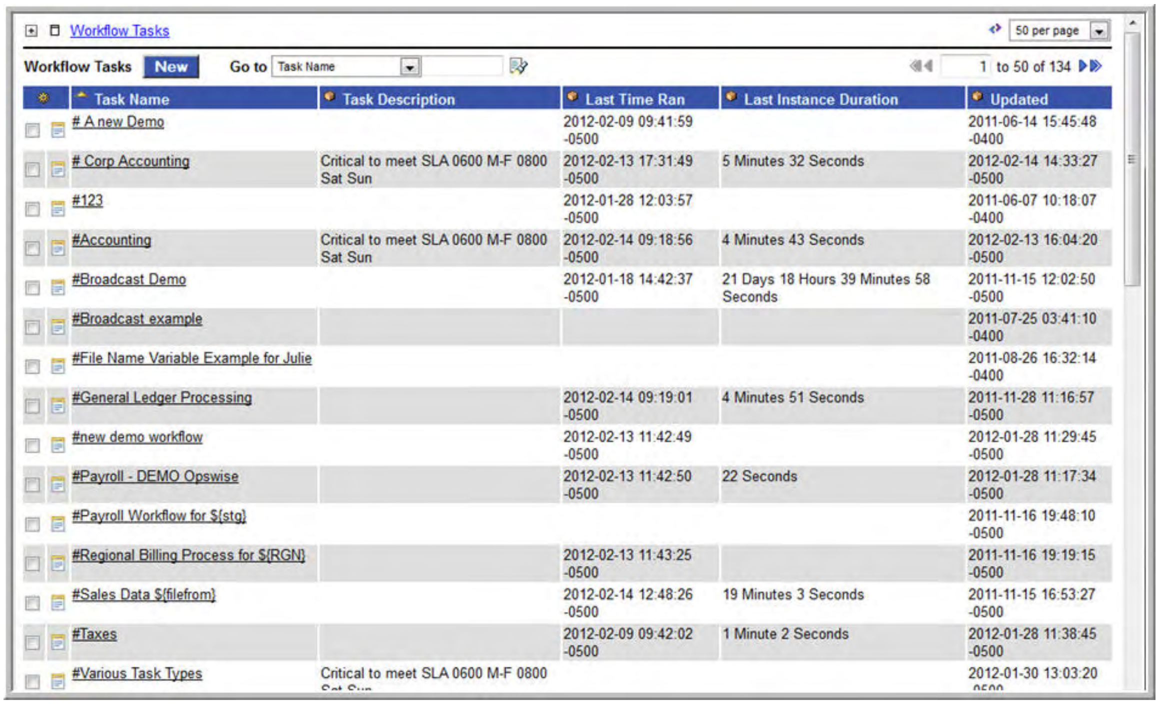
Unable to render embedded object: File (up.gifx.gif!	more than one parent, clicking this button displays a list of all parent workflows; in this case, to navigate to a parent workflow, double click the workflow's icon. To return to your original workflow, click its name. Note that the current workflow name displays in the lower right corner of the editor.
!Ops_open_folder.gif) not found.	**Open Workflow** --- Allows you to drag another workflow into the current workflow or temporarily switch to another workflow without closing the current workflow. Opens a window and displays a list of defined workflows. You can either double-click a workflow from the list to open it and edit it, or you can click and drag a workflow into the current workflow. If you open another workflow, use Open Workflow to return to the previous workflow. Hint: As a best practice, save your workflow before opening another one.
	Save --- Saves the current workflow.
	Add Task --- Displays the Task Find window, which allows you to search for and select tasks for the workflow. To search for and select tasks: 1. Click the **Add Task** icon. 2. Optionally, enter a task name, a partial string, optionally using or ? as wildcards , or select the type of task you are searching for. 3. Click **Search**. Opswise displays the tasks that meet your selection criteria. 4. To bring a task onto the workflow canvas, click and drag the task's icon out of the window and onto the canvas. 5. To close the Task Find window, click the **X** in the upper right corner.
	Select --- Enters **Select** mode, which allows you to click on tasks or links in order to move or delete them. Use this to perform any of the following: • Switch out of **Connect** mode • Select and open tasks • Select and delete tasks and links • Select and move tasks and links
	Pan --- Enters **Pan** mode, which allows you to scroll to different areas of the workflow.
!Ops_connect.gif!	**Connect (bent**) and **Connect (straight)** --- Enters **Connect** mode, which allows you to create links between tasks. To create a link: 1. Click either icon to enter **Connect** mode. 2. Click on the "upstream" task. If you are using Firefox, click the center of the task icon. In Internet Explorer, click the task name. 3. Drag the cursor to the "downstream" task. You will see a colored line as you drag. 4. When you reach the "downstream" task, release the cursor. The link appears as a straight or bent line, depending on which icon you selected.
!undo.gifx.gif!	**Undo** --- Click to undo the most recent change.
!redo.gifx.gif!	**Redo** --- Click to redo the most recent change that you undid by clicking **Undo**.
!cut.gifx.gif!	**Cut** --- Deletes the selected object or objects (tasks and links or both) and keeps a copy in memory. Use CTRL-Click to select and cut multiple objects. Hint: Do not use cut and paste to move workflow elements; use select and drag.
	Copy --- Copies the selected object or objects (tasks, links or both). Use CTRL-Click to select and copy multiple objects.
	Paste --- Pastes the copied or cut object or objects to the currently open workflow.
✕	**Delete** --- Permanently deletes the selected object or objects. Delete does not keep a copy of the deleted objects in memory.
	Fit --- Fits the workflow into the display. If necessary, this shrinks the icons and size of the workflow in order to make it fit. You can undo a Fit by selecting **Actual Size** .
!zoomin.gifx.gif!	**Zoom In** --- Zooms in (enlarges) the workflow. To return the workflow to its default size, select **Actual Size** Unable to render embedded object: File (Zoomactual.gifx.gif!.
!zoomout.gifx.gif) not found.	**Zoom Out** --- Zooms out (diminishes) the workflow. To return the workflow to its default size, select **Actual Size** .
!Zoomactual.gifx.gif!	**Actual Size** --- Returns the workflow to its default size after a **Fit** or **Zoom**.
	Zoom --- Opens a window that allows you to specify a zoom ratio. For example, to double the size of the workflow, enter 200 and click **OK**. To return the workflow to its default size, select **Actual Size** Unable to render embedded object: File (Zoomactual.gifx.gif!.
!Ops_tree_layout_horizontal.gif) not found.	**Horizontal Layout** --- Reformats the workflow into a horizontal layout.
	Vertical Layout --- Reformats the workflow into a vertical layout.
	Toggle Vertex Style --- For running workflows, switches the icon display between status-related icons and

	task-related icons. This icon only appears on the icon bar when you are monitoring a running workflow.
	Outline --- For large workflows, the outline provides a way of positioning a specific area of the workflow in the display, without using the **Pan** icon. 1. Click the **Outlin**e icon. The **Outline** window opens. 2. In the **Outline** window, move and/or resize the blue box to identify the area of the workflow you want to work on. The display repositions to show only the area within the blue box.
	Help --- Displays help documentation for workflows.
	Console --- While a workflow is running, you can click the **Console** icon to display processing messages. For more details, see Monitoring Workflows.

Defining a Workflow

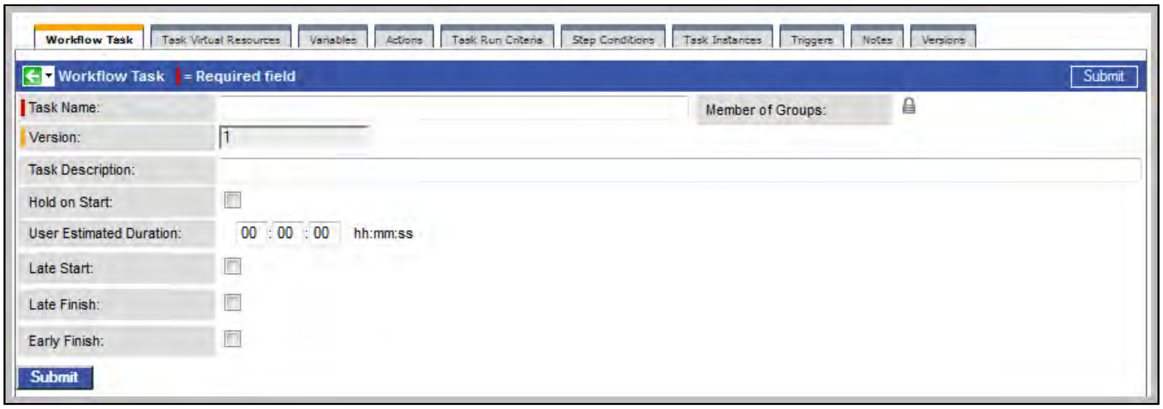
Creating a New Workflow

1. From the navigation pane, select **Workflow Tasks**. The Workflow Tasks List screen displays.



	Task Name	Task Description	Last Time Ran	Last Instance Duration	Updated
	# A new Demo		2012-02-09 09:41:59 -0500		2011-06-14 15:45:48 -0400
	# Corp Accounting	Critical to meet SLA 0600 M-F 0800 Sat Sun	2012-02-13 17:31:49 -0500	5 Minutes 32 Seconds	2012-02-14 14:33:27 -0500
	#123		2012-01-28 12:03:57 -0500		2011-06-07 10:18:07 -0400
	#Accounting	Critical to meet SLA 0600 M-F 0800 Sat Sun	2012-02-14 09:18:56 -0500	4 Minutes 43 Seconds	2012-02-13 16:04:20 -0500
	#Broadcast Demo		2012-01-18 14:42:37 -0500	21 Days 18 Hours 39 Minutes 58 Seconds	2011-11-15 12:02:50 -0500
	#Broadcast example				2011-07-25 03:41:10 -0400
	#File Name Variable Example for Julie				2011-08-26 16:32:14 -0400
	#General Ledger Processing		2012-02-14 09:19:01 -0500	4 Minutes 51 Seconds	2011-11-28 11:16:57 -0500
	#new demo workflow		2012-02-13 11:42:49 -0500		2012-01-28 11:29:45 -0500
	#Payroll - DEMO Opswise		2012-02-13 11:42:50 -0500	22 Seconds	2012-01-28 11:17:34 -0500
	#Payroll Workflow for ${stg}				2011-11-16 19:48:10 -0500
	#Regional Billing Process for ${RGN}		2012-02-13 11:43:25 -0500		2011-11-16 19:19:15 -0500
	#Sales Data ${filefrom}		2012-02-14 12:48:26 -0500	19 Minutes 3 Seconds	2011-11-15 16:53:27 -0500
	#Taxes		2012-02-09 09:42:02 -0500	1 Minute 2 Seconds	2012-01-28 11:38:45 -0500
	#Various Task Types	Critical to meet SLA 0600 M-F 0800			2012-01-30 13:03:20

2. Click **New**. The Workflow Task screen displays.



Task Name:	
Member of Groups:	
Version:	1
Task Description:	
Hold on Start:	□
User Estimated Duration:	00 : 00 : 00 hh:mm:ss
Late Start:	□
Late Finish:	□
Early Finish:	□

3. Using the field descriptions provided below as a guide, complete the fields as needed.

4. Right-click on the title bar and select **Save**. The workflow is added to the database and additional buttons appear at the bottom.
5. Click **Edit Workflow** to proceed into the Workflow Editor. Opswise displays a blank Workflow Editor screen.
6. Follow the instructions provided below for adding tasks, creating connections, specifying conditions on Connections, organizing, and saving your workflow.
   Or, see Icon Reference for a quick reference guide to each tool icon.

Field Descriptions on Workflow Task Screen

The table below describes the fields, buttons, and tabs on the task definition and task instance screens. Color coding is provided that differentiates the following three types of fields:

- Fields that display on the task definition and task instance screens are shown in black.
- Fields that display only on the task definition screen are shown in green.
- Fields that display only on the task instance screen are shown in maroon.

Field Name	Description
Task/Instance Name	Required. Name used within Opswise to identify this task. Up to 40 alphanumerics; variables supported. It is the user's responsibility to develop a workable naming scheme for tasks.
Invoked by	Task instance only; system-supplied. How the task instance was launched. One of the following: • Trigger: (Trigger Name) - The instance was launched by the named trigger. • Workflow: (Workflow Name) - The instance was launched by the named workflow. • Manually Launched - The instance was launched by a user. To determine the name of the user: 1. From the Activity or All Task Instances screen, click the task instance name to open the record. 2. The Execution User field identifies the user who launched the task instance.
Task	Task instance only; system-supplied. Hover over the paper icon to display more information about the task instance. Click the paper icon to display the task definition record.
Instance Reference Id	Task instance only; system-supplied. The system increments this number each time the task is run.
Member of Groups	User-defined. Allows you to select one or more Opswise groups that this record definition belongs to. Click on the lock icon to unlock the field and select groups.
Version	Task definition only; system-supplied. The version number of the current record, which is incremented by the system every time a user updates a record. Click on the Versions tab to view previous versions. For details, see Record Versioning.
Summary	User-supplied description of this record.
Status	Task instance only; system-supplied. See Task Statuses.
Status Description	Task instance only; system-supplied. Provides additional information, if any, about the status of the task.
Start Time	Task instance only; system-supplied. The date and time the task started.
Duration	Task instance only; system-supplied. The amount of time the task took to run.
End Time	Task instance only; system-supplied. The date and time the task instance completed.

User Estimated End Time	Task instance only; system-supplied. If the user entered information into the User Estimated Duration field in the task definition, the system uses this information to calculate an end time for the task instance, based on the date/time the task instance started.
Shortest Estimated End Time	Task instance only; system-supplied.
Average Estimated End Time	Task instance only; system-supplied.
Longest Estimated End Time	Task instance only; system-supplied.
Hold on Start	If enabled, when the task is launched it appears in the Activity display with a status of Held. The task runs when the user releases it.
Hold Reason	Information about why the task will be put on hold when it starts.
User Estimated Duration	Task definition only; optional. The estimated amount of time it should normally take to run this task. The system uses this information to calculate the User Estimated End Time on a task instance record.
Late Start	If enabled, and if the task instance starts after the time specified in the Late Start Time field (see below), the task instance is flagged as late. To determine whether a task instance started late, open the task instance and locate the Started Late field; the field is checked if the instance started late.
Late Start Type	Required if Late Start is enabled. Options are: • **Time** - Flag the task if it starts after the specified time. • **Duration** - Flag the task if it starts a certain amount of time after the programmed start time. The task must have a specific start time.
Late Start Time	Time after which the task start time is considered late. Use hh:mm, 24-hour time
Late Start Duration	Duration (amount of relative time) after which the task is considered to have started late. For a task within a workflow, the duration is the period between the time the workflow starts and the time the task itself starts. For example, a task might have a Late Start Duration of 60 minutes. If the workflow starts at 9:00 a.m. but the task itself does not start until 10:30, the task has started late. For a task that is not within a workflow, Late Start Duration has meaning only if the task has been held upon starting. For example, if a task has a Late Start Duration of 60 minutes and the Hold on Start field is enabled, if the task is not released from hold within the amount of time specified in the Late Start Duration field, the task has started late.
Started Late	Task instance only; system-supplied. This field is flagged if the task started later than the time specified in the Late Start fields.
Late Finish	If enabled, and if the task instance finishes after the time or period specified, the task instance is flagged as late. You can specify a time or duration to determine a late finish (see Late Finish Type below). To determine whether a task instance finished late, open the task instance and locate the Finished Late field; the field is checked if the instance finished after the specified time or lasted longer than expected. This field only appears on the task instance if the user specified a Late Finish in the task definition.

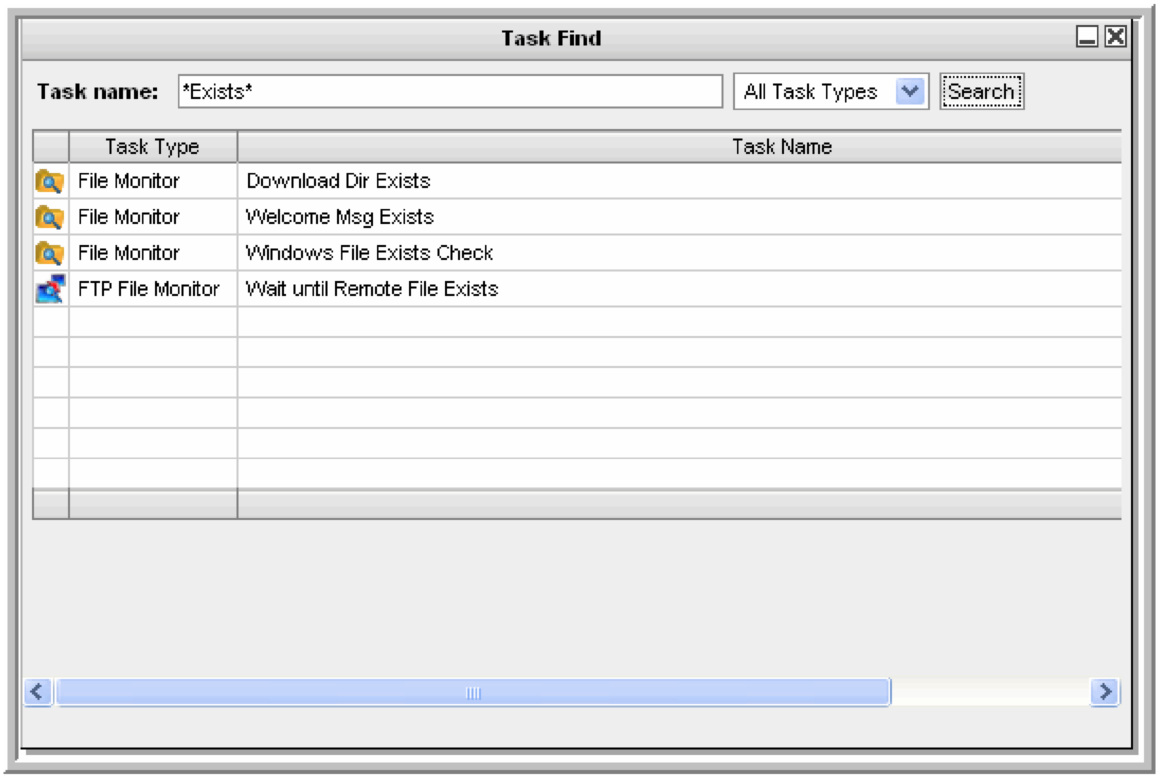
Late Finish Type	If Late Finish is enabled, use this field to specify whether you want to use time (see Late Finish Time below) or duration (see Late Finish Duration below) to determine whether the task instance finished late.
Late Finish Time	If Late Finish Type is Time, use this to specify the time after which the task finish time is considered late. Use hh:mm, 24-hour time.
Late Finish Duration	If Late Finish Type is Duration, use this to specify the longest amount of time this task instance should take to run. You can specify any combination of hours, minutes, and seconds.
Finished Late	Task instance only; system-supplied. This field is flagged if the task finished later than the time or duration specified in the Late Finish fields.
Early Finish	If enabled, and if the task instance finishes before the time or period specified, the task instance is flagged as early. You can specify a time or duration to determine an early finish (see Early Finish Type below). To determine whether a task instance finished early, open the task instance and locate the Finished Early field; the field is checked if the instance finished before the specified time or did not last as long as expected. This field only appears on the task instance if the user added Early Finish specifications to the task definition.
Early Finish Type	If Early Finish is enabled, use this field to specify whether you want to use time (see Early Finish Time below) or duration (see Early Finish Duration below) to determine whether the task instance finished early.
Early Finish Time	If Early Finish Type is Time, use this to specify the time before which the task finish time is considered early. That is, enter a time at which the task should still be running. Use hh:mm, 24-hour time.
Early Finish Duration	If Early Finish Type is Duration, use this to specify the shortest amount of time this task instance should take to run. You can specify any combination of hours, minutes, and seconds.
Finished Early	Task instance only; system-supplied. This field is flagged if the task finished earlier than the time specified in the Early Finish fields.
First Time Ran	Task definition only; system-supplied. Displays after the first time the task runs. The date and time this task first ran.
Lowest Instance Time	Task definition only; system-supplied. Displays after the first time the task runs. The shortest amount of time this task has taken to run.
Last Time Ran	Task definition only; system-supplied. Displays after the first time the task runs. The most recent date and time the task ran.
Average Instance Time	Task definition only; system-supplied. Displays after the first time the task runs. Shows the average amount of time this task takes to run.
Number of Instances	Task definition only; system-supplied. Displays after the first time the task runs. Shows the number of times this task has run.
Highest Instance Time	Task definition only; system-supplied. Displays after the first time the task runs. The longest amount of time this task has taken to run.

Last Instance Duration	Task definition only; system-supplied. Displays after the first time the task runs. The amount of time the task took to run the last time it ran.
Submit button	Submits the new record to the database.
Update button	Saves updates to the record.
Launch Task button	Manually launches the task.
Delete button	Deletes the current record.
View Parent button	Task instance only; Displays this task's parent task (workflow), if any.
Show Details button	Task instance only; displays detailed information about this task instance.
View Workflow button	Displays the graphical workflow.
Cancel button	Cancels a running task. See Cancelling a Task Run.
Release Recursive button	Task Instance only. Releases the entire held workflow and its tasks. See Releasing a Task From Hold.
Hold button	Task instance only; see Putting a Task on Hold.
Force Finish button	Task instance only; see Force Finishing a Task.
Output tab	Task instance only. Displays output generated from the process, if any, based on specifications provided by the user in the Automatic Output Retrieval fields in the task definition.
Task Virtual Resources tab	Lists Virtual Resources to which this task is assigned. See Creating Virtual Resources.
Variables tab	Displays all variables associated with this record.
Actions tab	Allows you to specify actions that the system will take automatically based on events that occur during the execution of this task. Events are task instance status, exit codes, late start, late finish, and early finish. Actions are: • **Abort Action** - Abort the task if certain events occur. For details, see Setting Up Abort Actions. • **Email Notification** - Send an email if certain events occur. For details, see Creating Email Notifications. • **SNMP Notification** - Send an email if certain events occur. For details, see Creating SNMP Notifications. • **Set Variable** - Used in workflows to set a variable based on the occurrence of certain events. For details, see Setting Variables in a Workflow.
Task Run Criteria tab	Workflows only. Allows you to specify skip and run criteria for specific tasks in the workflow.

Step Conditions tab	Displays a list of all step conditions defined for this task.
Task Instances tab	System-supplied. Displays a list of all task instances generated by this task definition.
Triggers tab	Displays a list of all triggers that have been defined to launch this task. Also allows you to add new triggers. If you add a new trigger from this location, the system automatically constructs a default trigger name as follows: <current task name>#TRIGGER#. You can change the default name if desired. For instructions on creating triggers, see Creating Triggers.
Notes tab	Displays all notes associated with this task. See Runbook Notes.
Versions tab	Stores copies of all previous versions of the current record (see Record Versioning).

Searching For and Adding Tasks

1. Click the **Add Task** icon .
   The **Task Find** window displays.
2. Several methods are available for finding tasks:
   - To find a specific task, type the name into the Task name field and click **Search**.
   - To bring up a list of tasks whose names match a string, type the string into the Task name field and use one or more wildcards . The wildcard can appear anywhere in the string and you can use more than one if necessary. For example, to find tasks whose name begins with "Fee", type Feein the Task name field and click **Search**.
   - To bring up a list of tasks of a specific type, such as Windows, select the task type from the drop down menu and click **Search**.
3. Opswise fetches the task or tasks that match your search criteria.



Task Find

Task name: *Exists*

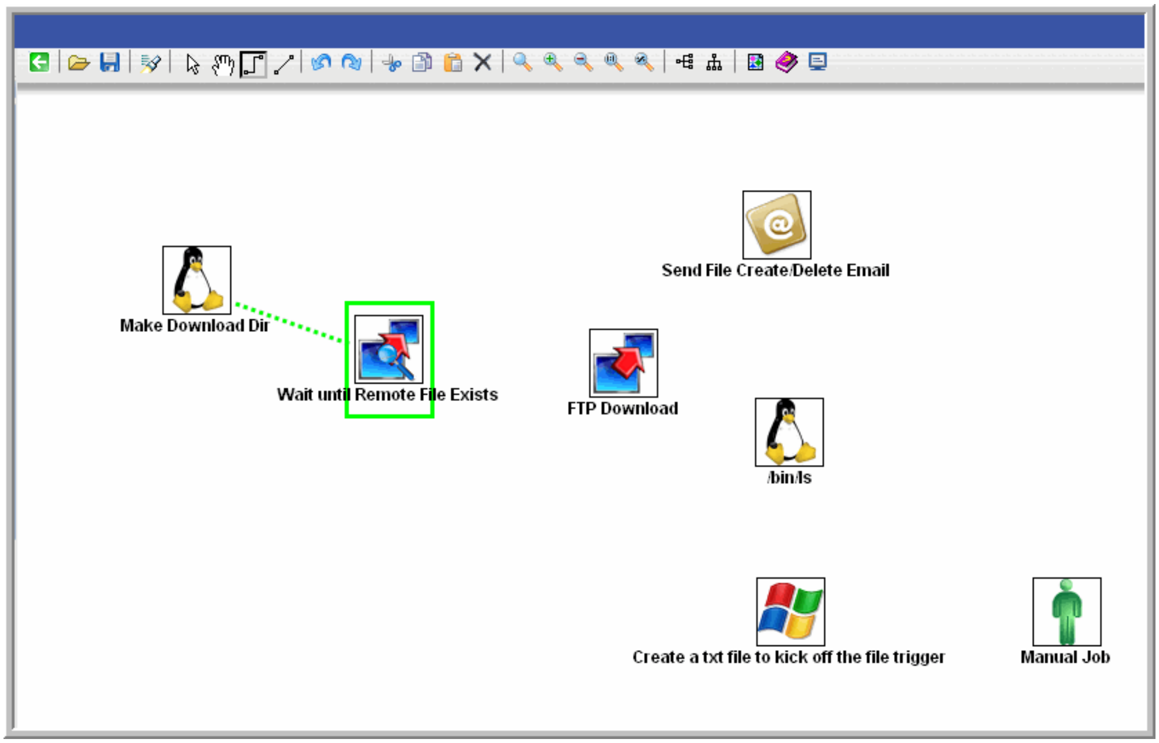
Task Type	Task Name
File Monitor	Download Dir Exists
File Monitor	Welcome Msg Exists
File Monitor	Windows File Exists Check
FTP File Monitor	Wait until Remote File Exists

4. To add a task to the workflow canvas, click on the icon to the left of the task and drag it onto the canvas.
5. Repeat these steps until you have added all the tasks you need.
6. To close the **Task Find** window, click the **X** in the upper right corner. Or, to keep the window open but minimized, click the minimize icon.

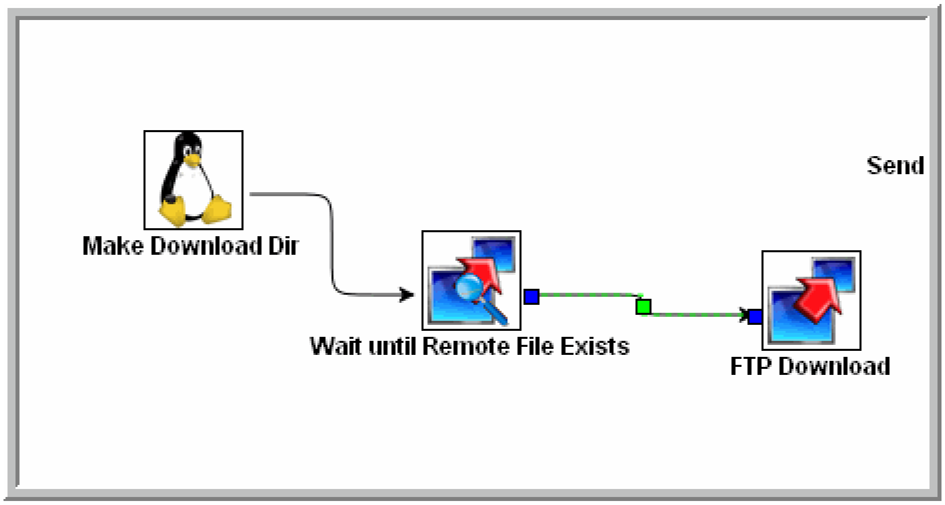
Specifying Connections

You must first drag tasks onto the canvas before specifying connections between them.

1. Click on one of the connectors, either , which is either straight or diagonal, or , which bends if necessary. This enters **Connection** mode.
2. Click on the upstream task, that is, the task whose status will trigger the next task. For Firefox, make sure you click the center of the task icon; for Internet Explorer, click on the task name, **hold down the cursor,** and drag the connector.
3. Drag the connector to the downstream task, that is, the task that will be triggered. As you drag, a dotted line appears to indicate the connection. Drag the connector all the way to the center of the downstream, highlighted task.


Send File Create/Delete Email
Make Download Dir
Wait until Remote File Exists
FTP Download
/bin/ls
Create a txt file to kick off the file trigger
Manual Job

4. When the cursor is at the center of the downstream task icon, let up on the mouse. The connector is in place and still highlighted. Optionally, you can reposition the connection by dragging it to a new location. Arrows on the connectors indicate the direction of the workflow.

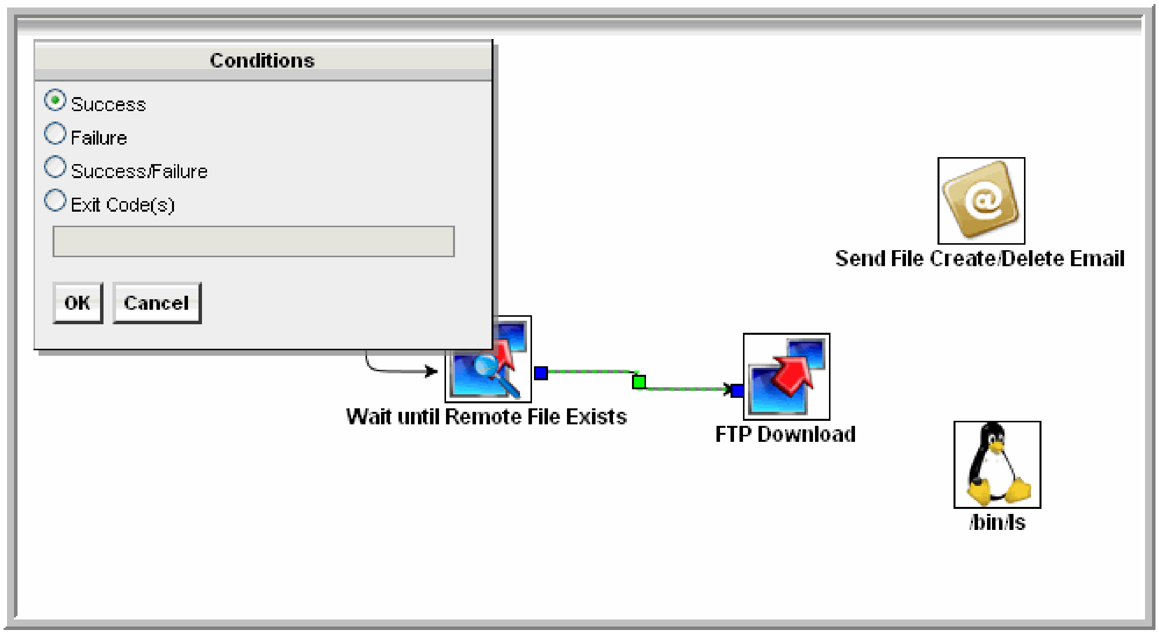


5. Repeat this process for other connectors you need to add.

Specifying Conditions on Connections

Once you have created connections between the tasks on a workflow, you can now specify what kind of dependency exists between the upstream and downstream task. The default condition is Success. That is, the downstream task runs if the upstream task runs to Success.

1. Right-click on the connection for which you want to specify a condition.
2. Select **Conditions**. The Conditions window displays.



3. Specify one of the following conditions:
   - **Success** -- Run the downstream task if the upstream task goes to Success.
   - **Failure** -- Run the downstream task if the upstream task goes to Failure.
   - **Success/Failure** -- Run the downstream task if the upstream task goes to Success or Failure.
   - **Exit Code(s)** -- Run the downstream task if the upstream task returns one of the specified exit code(s). Format: Numeric. Use commas to list a series of exitcodes; use hyphens to specify a range. Example: 1,5, 22-30.
     Success conditions are indicated by a solid line in the Workflow Editor; all other conditions display as a dotted line.
4. Click **OK**.

Creating Conditional Paths

Opswise allows you to specify separate processing paths for each condition. For example, you might specify a group of tasks that will run if the upstream task goes to Success and a second group of tasks that will run if the upstream task goes to Failure. When Opswise recognizes that conditional paths have been specified, the tasks in the path whose condition is met run, and the tasks in the path whose condition is not met go to a Skipped status.

Opswise identifies a conditional path when:

1. The upstream task goes to a finished status (Success or Failure).
2. As a result, at least one downstream dependency is satisfied and at least one downstream dependency is not satisfied.

For example, Task A is at the top of the workflow. Three conditional paths have been specified: one for failure and two for success. Task A executes and goes to Success status. Opswise identifies this as a conditional path and puts all the tasks in the failure path into Skipped status; the tasks in the two Success paths begin running normally. As another example, Task A is at the top of the workflow. Two conditional paths have been specified: one for exit codes 1 through 10, the second for exit codes 11 through 20. If Task A completes with exit code 5, the first path runs and the second path is skipped. If Task A completes with exit code 15, the first path is skipped and the second path runs. If Task A completes with exit code 25, neither condition is satisfied and both paths remain in Waiting status.

Using Multiple Connections

If a task has more than one upstream connection, the task remains in a Waiting status until all of the conditions of those multiple connections are evaluated. If all of the connections are with paths that have been skipped, the task goes to a Skipped status. If at least one of the connections is with a path that has executed and all connections have been evaluated, the task executes.

Moving Workflow Elements

Once you have positioned one or more tasks and connections on the workflow canvas, you can reposition the objects as needed.
To move a single task:

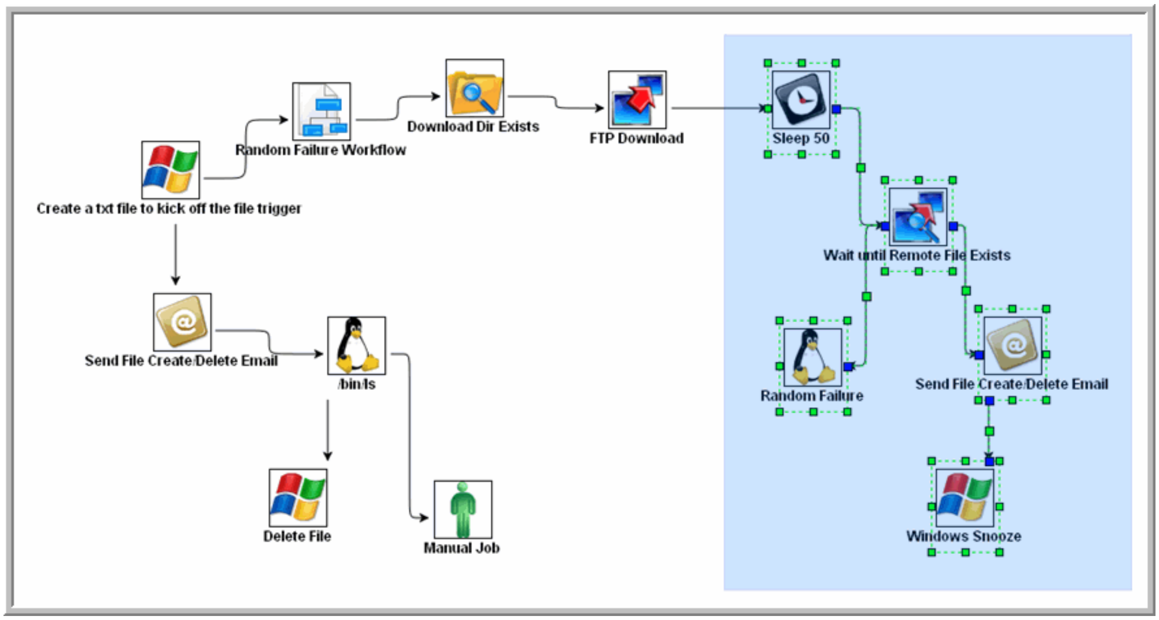
1. Click the **Select** icon to enter **Select** mode.
2. Click on a task and drag it to its new location. If the task has any connectors attached, they remain connected and lengthen or shorten as necessary.

Or,

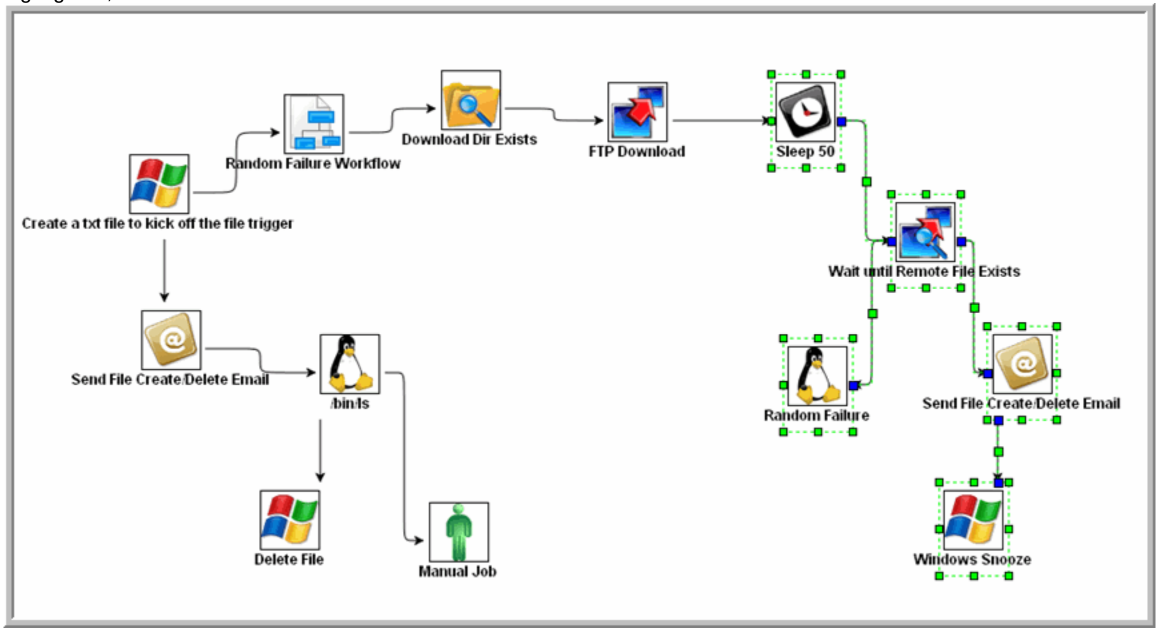
1. Click one of the **Connect** icons to enter **Connect** mode.
2. Click on the end of the connection, near where it connects to the task you want to move.
3. Drag the connection and task to their new location. The connector remains connected and shortens or lengthens as necessary.

To move a group of tasks:

1. Click the **Select** icon to enter **Select** mode.
2. Position the cursor near the group you want to select.
3. Click on the canvas and begin dragging. A blue shaded box appears. This identifies the selection area.
4. Drag the selection area over the tasks you want to select, as shown in the following example:


Random Failure Workflow
Download Dir Exists
FTP Download
Sleep 50
Create a txt file to kick off the file trigger
Wait until Remote File Exists
Send File Create/Delete Email
/bin/ls
Random Failure
Send File Create/Delete Email
Windows Snooze
Delete File
Manual Job

5. When you have selected all the tasks you want to move, release the mouse. The tasks and connectors included in the group are highlighted, as shown:


Random Failure Workflow
Download Dir Exists
FTP Download
Sleep 50
Create a txt file to kick off the file trigger
Wait until Remote File Exists
Send File Create/Delete Email
/bin/ls
Random Failure
Send File Create/Delete Email
Windows Snooze
Delete File
Manual Job

6. Click on one of the selected tasks and drag it to the new location. All the selected tasks are moved.
7. To deselect the group of tasks, click elsewhere on the canvas.

Or,

1. Click one of the **Connect** icons to enter **Connect** mode.
2. Position the cursor near the group you want to select.
3. Click on the canvas and begin dragging. A blue shaded box appears. This identifies the selection area.
4. Drag the selection area over the tasks and connectors you want to select. Make sure you select the entire connector(s), or you will not be able to move the objects.
5. When you have selected all the tasks and connectors you want to move, release the mouse. The tasks and connectors included in the group are highlighted.
6. Click on one of the selected connectors and drag it to the new location. All the selected tasks and connectors are moved.
7. To deselect the group of tasks and connectors, click elsewhere on the canvas.

Deleting Workflow Elements

1. Highlight the workflow element you want to delete. Or, select a group of elements you want to delete.
2. Click the **Delete** ✕ icon. The element an all its associated connectors are deleted.

Copying Workflow Elements

1. Highlight the workflow element you want to copy. Or, select a group of elements you want to copy.
2. Click the **Copy** icon. The element and all its associated connectors are copied.
3. To paste the copied elements onto the workflow, click **Paste** . The copied elements are pasted next to the originals.
4. To move the new elements to a new location, go into **Connect** mode and click and drag one of the connectors or go into **Select** mode and click and drag one of the icons. If you have copied multiple elements, do not click elsewhere in the display before dragging the group. Clicking elsewhere deselects the copied elements.

Undoing and Redoing Workflow Changes

To undo the most recent change, click **Undo** . To redo a change you just undid, click **Redo** .

Zooming In and Out

Several features are available for zooming in and out on large workflows:

- Click **Fit** to fit the entire workflow onto the display.
- Click **Zoom In** to increase the size of the workflow (to view details).
- Click **Zoom Out** to decrease the size of the workflow (to view the entire workflow)
- Click **Actual Size** to return the workflow to its actual (original) size.
- Click **Zoom** to enter a zoom percentage.

Panning Around in Large Workflows

For large workflows that cannot be displayed entirely on the screen, you can pan around from area to area. Two methods are provided: the **Pan** icon and the **Outline** icon.

1. Click the **Pan** icon. This enters **Pan** mode.
2. Click the display and drag the workflow so that it displays the area you want to work on.

OR

1. Click the **Outline** icon. The **Outline** window opens.
2. In the **Outline** window, move and/or resize the blue box to identify the area of the workflow you want to work on. The display repositions itself as indicated in the **Outline** window.

Automatically Formatting a Workflow

You can apply automatic formatting to your workflow. This process does not change any connections or content, but reorganizes the workflow into a more legible display.

- To create a horizontal layout, click .
- To create a vertical layout, click .

Displaying Workflow Documentation

Click the **Help** icon.

Displaying Processing Messages

While a workflow is running, you can click the **Console** icon to display processing messages. For more details, see Monitoring Workflows.

Saving the Workflow

Click the **Save** icon.

Switching Between Workflows

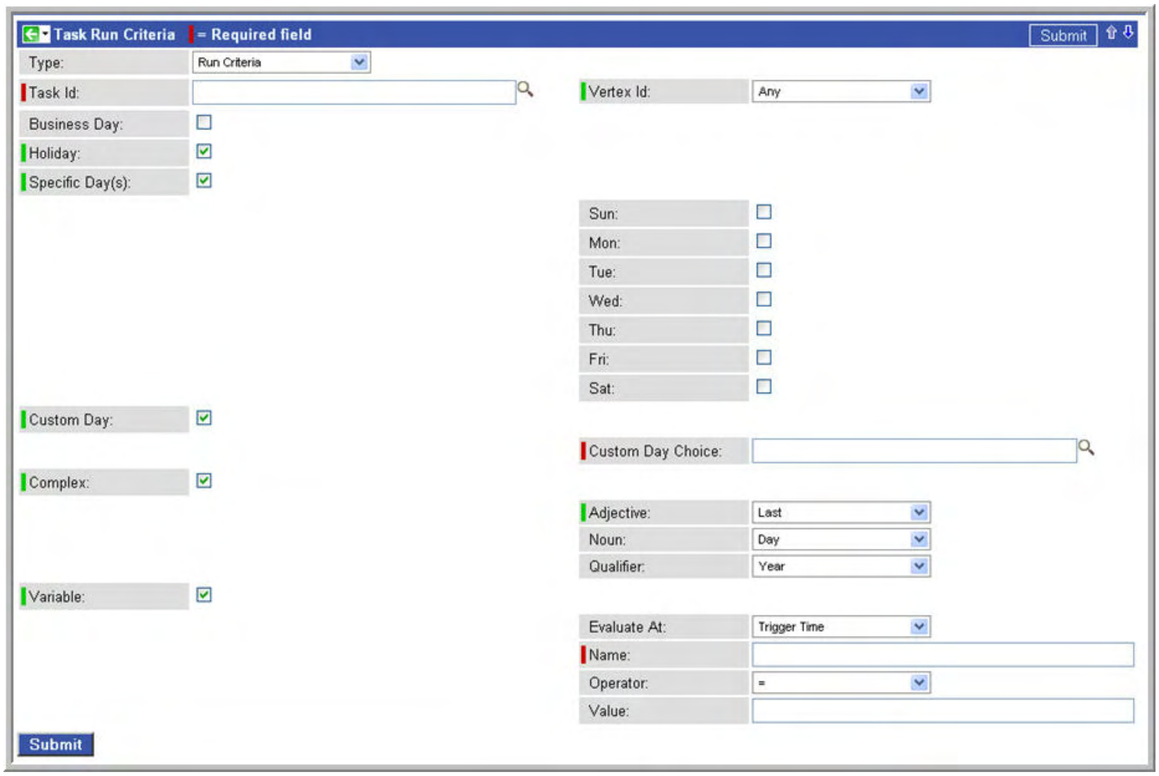
You can switch between workflows without using the navigation pane by clicking the **Open Workflow** icon and selecting the workflow you want to switch to. It is recommended, but not required, that you save your work before switching to another workflow. You can switch back to your original workflow by clicking **Open Workflow** again.

Adding Skip/Run Criteria for Specific Tasks

You can add special instructions that specify conditions under which a specific task (or sub-workflow) within the workflow should run or skipped. The system evaluates these instructions when determining whether to run each task within a workflow. For example, you might want to skip a specific task in a workflow on a certain day or date or you might want to run a specific task only if a certain variable is set to a specific value.

Creating New Run Criteria

1. Display the workflow for which you want to specify run criteria.
2. Click the **Task Run Criteria** tab. This list displays a list of specified task run criteria.
3. From the **Task Run Criteria** list, select **New**. Opswise displays the **Task Run Criteria** screen:



Type: Run Criteria

Task Id:

Vertex Id: Any

Business Day: ☐

Holiday: ☑

Specific Day(s): ☑

Sun: ☐

Mon: ☐

Tue: ☐

Wed: ☐

Thu: ☐

Fri: ☐

Sat: ☐

Custom Day: ☑

Custom Day Choice:

Complex: ☑

Adjective: Last

Noun: Day

Qualifier: Year

Variable: ☑

Evaluate At: Trigger Time

Name:

Operator: =

Value:

4. Using the field descriptions provided below as a guide, complete the fields as needed. For example, to tell the system not to run a specific task on business days or holidays, select **Skip Criteria** in the Type field, select the task and enable **Business Day** and **Holiday**.
5. Click the **Submit** button to save the record and return to the menu, or, right-click the title bar and select **Save** to save the record and remain on the current display.
6. If appropriate, repeat these steps for any additional run criteria you want to add.

Task Run Criteria Field Descriptions

The following table provides descriptions for each field on the Task Run Criteria screen.

Field Name	Description
Type	User-defined. Indicates whether this is providing instructions on when to run or not run a task. Options: • **Run Criteria** - Provides instructions on when to run the task. • **Skip Criteria** - Provides instructions on when to skip the task.
Task	User-defined. Select the task for which you are specifying run or skip criteria. Click on the magnifying glass to browse for and select a task from the task list.
Vertex ID	Each task within a workflow has a unique vertex ID, which distinguishes it from other tasks of the same name, if any.
Business Day	If enabled, the task runs or skips on all business days.
Calendar	User-defined. Defines Business Days or holidays, if selected. Type in a calendar name or click the magnifying glass to browse for an existing calendar or to add a new calendar. To display details about the calendar specified in this field, hover over the paper icon.
Holiday	If enabled, the task runs or skips on holidays.
Specific Day(s) – Sunday through Saturday	If enabled, the task runs or skips on the day(s) you select in the right-hand column.
Custom Day	If enabled, the task runs or skips on the day you select in the Custom Day Choice field.
Custom Day Choice	If Custom Day is enabled, select the custom day for which you are specifying run or skip criteria. Click on the magnifying glass to browse for and select a day from the custom day list.

Complex	If enabled, the task runs or skips on the day(s) indicated in the Adjective, Noun, and Qualifier fields.
Adjective	If Complex is enabled, you can use this field to specify which in a series of days you want to select. Used in conjunction with the Noun and the Qualifier fields. For example, to specify "the 1st business day of the month," select Adjective: 1st, Noun: Business Day, Qualifier: Month. Options: • Every • 1st • 2nd • 3rd • 4th • Last
Noun	If Complex is enabled, , you can use this field to specify the type of day you want to select. Used in conjunction with the Adjective and the Qualifier fields. For example, to specify "the 1st business day of the month," select Adjective: 1st, Noun: Business Day, Qualifier: Month. This drop-down menu is populated as follows: • Sunday through Saturday • Day = any day • Business Day = The business days specified in the calendar selected in the Calendar field. • The Custom Days specified in the calendar selected in the Calendar field.
Qualifier	If Complex is enabled, , you can use this field to specify the period for your selection formula. Used in conjunction with the Noun and the Adjective fields. For example, to specify "the 1st business day of the month," select Adjective: 1st, Noun: Business Day, Qualifier: Month. Options: • Month • Year • January through December
Variable	If enabled, instructs the system to run or not run the task, depending on the value of a specific variable. Used in conjunction with fields: Evaluate At, Name, Operator, and Value (see below).
Evaluate At	Allows you to specify when you want the system to evaluate the variable. Options: • Trigger Time – Evaluate the variable when the workflow is triggered. • Run Time – Evaluate the variable when the task is about to run.
Name	Specifies the name of the variable being evaluated. The variable does not need to exist when this record is created.
Operator	Allows you to select the operator to be used in the evaluation. Options: =, !=, >, >=, <, <=, regex (regular expression).
Value	Allows you to specify the value or regular expression the system should look for when evaluating the variable. Up to 40 alphanumerics.
Submit button	Submits the new record to the database.
Update button	Saves updates to the record.
Delete button	Deletes the current record.
Variables tab	Displays all variables associated with this record.

Modifying an Existing Workflow

1. Click **Workflow Tasks** from the Navigation Pane. The Workflow Tasks list displays.
2. Locate the workflow you want to modify and click on the Task Name (or other identifier in the leftmost column).
3. Click **Edit Workflow**. The Workflow Editor opens.
4. Make your changes and click the **Save** 💾 icon.

Deleting a Workflow

1. Click **Workflow Tasks** from the Navigation Pane. The Workflow Tasks list displays.
2. Locate the workflow or workflows you want to delete.
3. Click on the box associated with those records and select **Delete** from the **Actions on selected rows...** menu.
4. To delete all workflow records, click on the box next to the **Actions on selected rows..**. menu and select **Delete**.

Specifying When a Workflow Runs

As with other task types, you can run the workflow manually or specify triggers that run the workflow task automatically based on times or events. For workflows, you can also specify skip and run criteria using the Task Run Criteria tab.

Monitoring Workflow Execution

You can monitor all system activity from the Activity display .

Using Variables and Functions

Opswise supports the following types of variables, all of which can be used in free text fields within tasks:

- User-Defined Variables – These variables are created by the user.
- Built-In Variables – These variables are maintained by the system and allow you to access information about task instances and other related data, such as task name, task status, and trigger name.
- Functions – These calculate some value, such as current date and time, or perform some function, such as ReplaceAll.

Opswise also supports several features that allow you to set variables under special circumstances:

- You can manually launch tasks and temporarily set user-defined variables. Click here for instructions.
- Opswise provides two methods that allow you to manually launch all the tasks associated with a trigger while supplying variable values used by the task(s). Click here for instructions.
- Setting Variables in a Workflow
- Using the ops-variable-set command to set variables. See Command Line Reference for instructions.

User-Defined Variables

Introduction

You can define Opswise variables in four locations:

1. Define **Trigger variables** by clicking the **Variables** tab in a trigger. Trigger variables are stored in the table **ops_local_variable**.
2. Define **Task variables** by clicking the **Variables** tab in a task. Task variables are stored in the table **ops_local_variable**.
3. Define **Workflow variables** by clicking the **Variables** tab in a workflow. Workflow variables are stored in the table **ops_local_variable**.
4. Define **Global variables** by selecting **Automation Center > Variables** from the Navigation Pane. Global variables are stored in the table **ops_variable**.

Variable Naming Conventions

- Variable names must begin with a letter.
- Allowable characters are alphanumerics (upper or lower case), and underscore (_).
- White spaces are not permitted
- Variable names are not case-sensitive.

Warning
Do not define Opswise variables with the prefix "ops_". That prefix is reserved for built-in variables.

Resolving User-Defined Variables

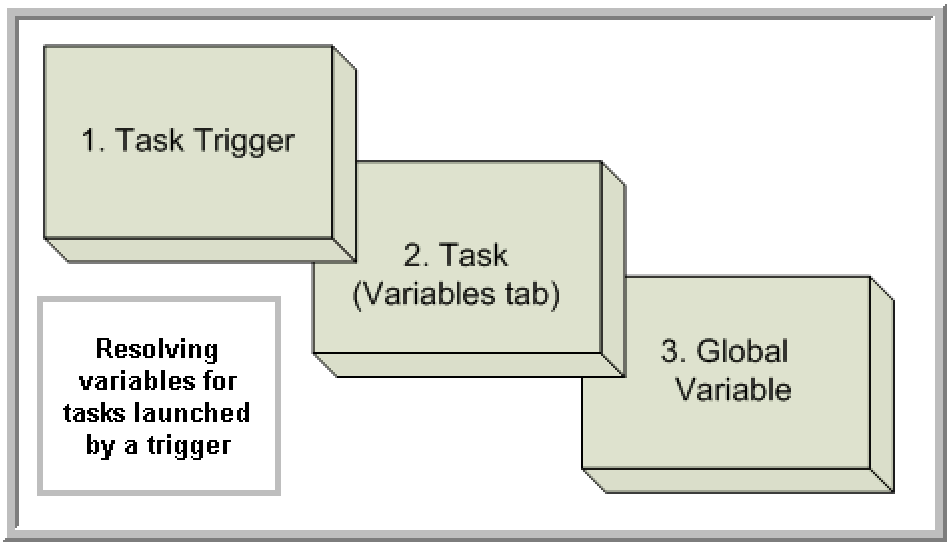
When Opswise creates a task instance from a task definition, it also resolves all variables specified in its free text fields. Because you can define variables at four different levels (trigger, task, workflow, and global), Opswise follows a prescribed formula to determine which variable takes precedence if duplicate variables have been defined. The general order of precedence, each of which may or may not exist in any given situation, is as follows:

1. Task trigger (highest precedence)
2. Task
3. Workflow trigger
4. Workflow
5. Global (lowest precedence)

The following scenarios provide more detailed information about how Opswise variables are resolved.

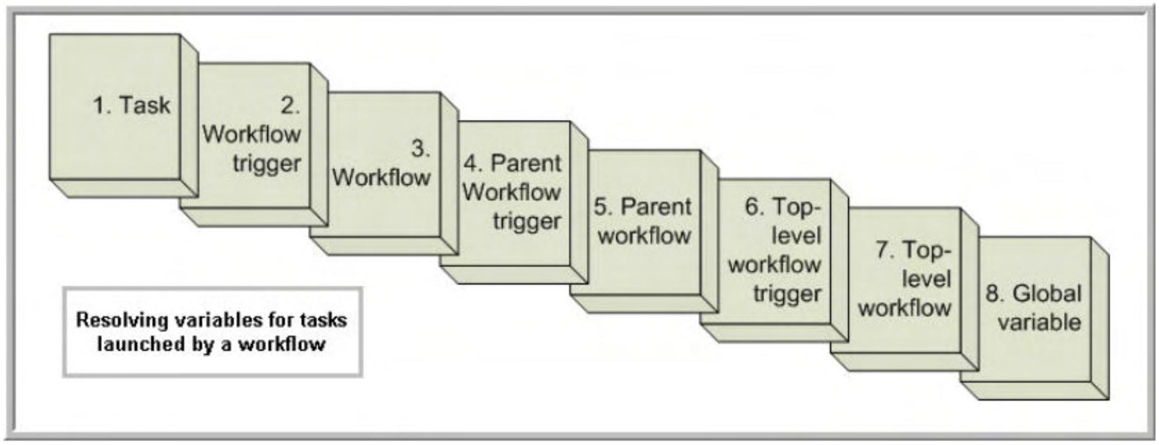
For Tasks Launched by a Trigger

1. If the trigger defines the variable in the variables tab, that value is used to resolve the variable.
2. If the trigger does not define the variable, the value from the variable tab in the task definition is used.
3. If neither the trigger nor the task define the variable, the variable definition in the global variables table is used.
4. If the global variables table does not define the variable, the variable remains unresolved.



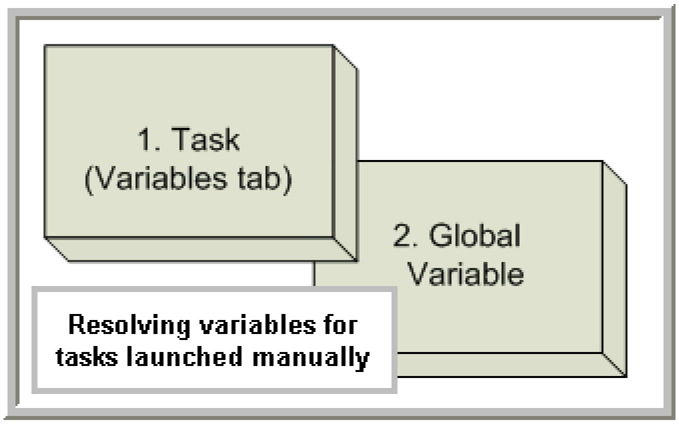
For Tasks Launched by a Workflow

1. If the task defines the variable in the variables tab, that value is used to resolve the variable.
2. If the task does not define the variable, and the workflow was launched by a trigger, the value defined in the trigger is used.
3. If the workflow's trigger does not define the variable or the workflow was not launched by a trigger, the value defined in the workflow is used.
4. If the workflow does not define the variable, and there is a parent workflow, the value defined in the parent workflow's trigger is used.
5. If the parent workflow's trigger does not define the variable or if there is no trigger, the value defined in the parent workflow is used.
6. If the parent workflow does not define the variable, the system checks up a level for the trigger on the next parent workflow.
7. If that trigger does not define the variable, it checks for variables associated with the workflow. (This continues until the top level workflow is reached.)
8. If the top-level workflow does not define the variable, the variable definition in the global variables table is used.
9. If the global variables table does not define the variable, the variable remains unresolved.


1. Task
2. Workflow trigger
3. Workflow
4. Parent Workflow trigger
5. Parent workflow
6. Top-level workflow trigger
7. Top-level workflow
8. Global variable
Resolving variables for tasks launched by a workflow

For Tasks Launched Manually

1. If the task defines the variable in the variables tab, that value is used to resolve the variable.
2. If the task does not define the variable, the variable definition in the global variables table is used.
3. If the global variables table does not define the variable, the variable remains unresolved.



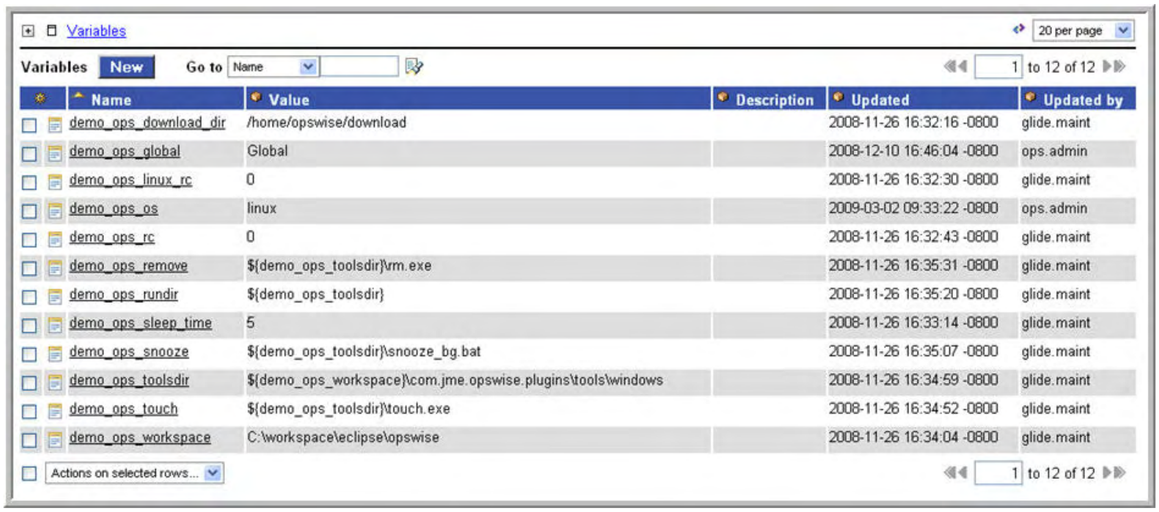
Format for Using Variables

When you enter a variable into a text field, precede the variable with the dollar sign ($) and enclose the variable in curly braces ({ }). You can enter a series of variables or nested variables. Examples are:

```
${variable_name}
${v1}${v2}
${${inner_variable}}
```

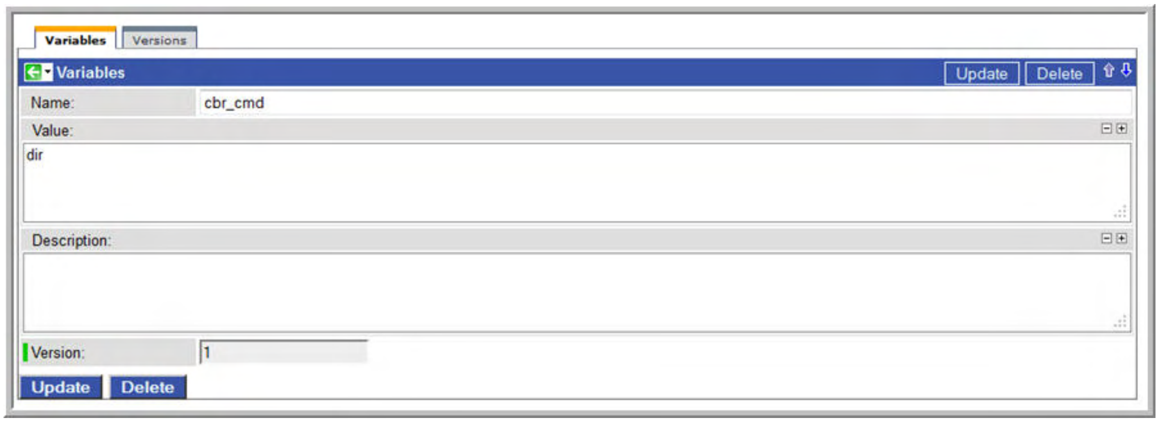
Defining a New Variable

1. For global variables: Select **Automation Center > Variables** from the Navigation Pane; for trigger, task, or workflow variables: Click the **Variables** tab from the record. The Variables List screen displays.



	Name	Value	Description	Updated	Updated by
□	demo_ops_download_dir	/home/opswise/download		2008-11-26 16:32:16 -0800	glide.maint
□	demo_ops_global	Global		2008-12-10 16:46:04 -0800	ops.admin
□	demo_ops_linux_rc	0		2008-11-26 16:32:30 -0800	glide.maint
□	demo_ops_os	linux		2009-03-02 09:33:22 -0800	ops.admin
□	demo_ops_rc	0		2008-11-26 16:32:43 -0800	glide.maint
□	demo_ops_remove	${demo_ops_toolsdir}\rm.exe		2008-11-26 16:35:31 -0800	glide.maint
□	demo_ops_rundir	${demo_ops_toolsdir}		2008-11-26 16:35:20 -0800	glide.maint
□	demo_ops_sleep_time	5		2008-11-26 16:33:14 -0800	glide.maint
□	demo_ops_snooze	${demo_ops_toolsdir}\snooze_bg.bat		2008-11-26 16:35:07 -0800	glide.maint
□	demo_ops_toolsdir	${demo_ops_workspace}\com.jme.opswise.plugins\tools\windows		2008-11-26 16:34:59 -0800	glide.maint
□	demo_ops_touch	${demo_ops_toolsdir}\touch.exe		2008-11-26 16:34:52 -0800	glide.maint
□	demo_ops_workspace	C:\workspace\eclipse\opswise		2008-11-26 16:34:04 -0800	glide.maint

2. Click **New**. The Variable Definition screen displays:



Name: cbr_cmd

Value: dir

Version: 1

3. Using the field descriptions provided below as a guide, complete the fields as needed.
4. Click the **Submit** button to save the record and return to the menu, or right-click and select **Save** to save the record and remain on the current display.
5. If appropriate, repeat these steps for any additional variables you want to add.

Variable Field Descriptions

The table below describes the fields and buttons on the Variables screen.

Field Name	Description
Name	Required. Name of variable. Up to 40 alphanumerics. The name must begin with an alphabetic character and can consist of: alphas (a-z, A-Z), numerics 0-9, _ (underscore). White spaces are not permitted; names are not case-sensitive. **Important Note** – Do not define Opswise variables with the prefix "ops_". That prefix is reserved for built-in variables.
Value	Optional. The value of the variable.
Description	Optional. Description of this variable.
Version	Task definition only; system-supplied. The version number of the current record, which is incremented by the system every time a user updates a record. Click on the Versions tab to view previous versions. For details, see Record Versioning.
Submit button	Submits the new record to the database.
Update button	Saves updates to the record.
Delete button	Deletes the current record.
Versions tab	Stores copies of all previous versions of the current record (see Record Versioning).

Built-In Variables

Introduction

Built-in variables are maintained by Opswise and provide information about task instances, agents, connectors, and core processors (cluster nodes). They can be used in free text fields on trigger variable values, tasks, task actions, agent notifications, connector notifications, and cluster node notifications.

To propagate the value of a built-in variable in a task instance to a parent workflow variable scope, you can use the Set Variable action. This allows the values of built-in variables to be shared among task instances within the same workflow.

Supported built-in variables and their descriptions are provided below. All built-in variables are prefixed with "ops_".

Task and Task Instance Variables

The following built-in variables are associated with task instances:

Variable	Description
`${ops_cmd}`	For tasks that launch a command on a Windows, Linux/Unix, or z/OS machine, resolves to the task command.
`${ops_cmd_parms}`	For tasks that launch a command on a Windows, Linux/Unix, or z/OS machine, resolves to the task command parameters.
`${ops_duration}`	Resolves to the task running time.
`${ops_end_time}`	Resolves to the task ending time.
`${ops_execution_user}`	Resolves to the ID of the user who launched the task or to the ID of the user who enabled the trigger that launched the task.
`${ops_exit_code}`	Resolves to the task instance exit code, if any.

`${ops_retry_count}`	Resolves to the current retry count.
`${ops_retry_interval}`	Resolves to the retry interval (seconds).
`${ops_retry_maximum}`	Resolves to the maximum retry count.
`${ops_start_time}`	Resolves to the task starting time.
`${ops_status}`	Resolves to the current task instance status.
`${ops_task_id}`	Resolves to the sys_id of the task instance.
`${ops_task_name}`	Resolves to the task name.
`${ops_task_type}`	Resolves to the task type.
`${ops_workflow_id}`	Resolves to the sys_id of the parent workflow task instance.
`${ops_workflow_name}`	Resolves to the name of the parent workflow.

Trigger Variable

When a task is launched by a trigger of any type, the value of the following built-in variable, if it is specified in the task, is passed into the task instance.

Variable	Description
`${ops_trigger_name}`	Resolves to the name of the trigger that launched the task instance.

File Monitor Variables

When one or more tasks are launched by a File Monitor trigger after the conditions in its associated File Monitor task are met, the built-in variables described below are passed into the tasks being launched by the trigger.

For example, the File Monitor trigger may specify the launch of a Windows task each time the associated File Monitor task detects the creation of a specific file. The Windows task might use one of these built-in variables as a command argument. Or, if the File Monitor task is not associated with a trigger but is running within a workflow, on completion you can propagate one or more of these built-in variable values to the parent workflow level using the Set Variable action. This allows you to pass information from the File Monitor task to a successor task within the same workflow hierarchy.

Variable	Description
`${ops_trigger_file_name}`	Resolves to the name of the file that fired the trigger.
`${ops_trigger_file_name_nopath}`	Resolves to the file name without any path information.
`${ops_trigger_file_name_simple}`	Resolves to the base file name.
`${ops_trigger_file_name_extension}`	Resolves to the file extension of a file.
`${ops_trigger_file_path}`	Resolves to the directory where the new file was created, but not the file itself. If the existence or non-existence of the final directory separator is a requirement, we recommend the use of ${ops_trigger_file_fullpath} and ${ops_trigger_file_fullpath_no_separator}, respectively.
`${ops_trigger_file_fullpath}`	Resolves to the directory where the new file was created, but not the file itself; includes the final directory separator.
`${ops_trigger_file_fullpath_no_separator}`	Resolves to the directory where the new file was created, but not the file itself; does not include the final directory separator.
`${ops_trigger_file_separator}`	Resolves to the separator appropriate to the platform where the agent is running. For Windows, resolves to a backslash; for Linux/Unix, resolves to forward slash. This variable may be useful if you want to piece together a pathname using a combination of text and variables. For example: `${ops_trigger_file_fullpath}sub_folder_name ${ops_trigger_file_separator}filename.txt`
`${ops_trigger_file_size}`	Resolves to the file size of the file that fired the trigger.
`${ops_trigger_file_date}`	Resolves to the file date of the file that fired the trigger.
`${ops_trigger_file_scan}`	Resolves to the file scan results.

`${ops_trigger_file_owner}`	Resolves to the file owner of the file that fired the trigger.
`${ops_trigger_file_group}`	Resolves to the file group of the file that fired the trigger.

FTP File Monitor Variables

The built-in variables described below are available for FTP File Monitor task instances and provide information about the file or file(s) that matched the monitor's criteria. You can use these in an FTP File Monitor action or in a successor task instance by propagating one or more of these built-in variable values to a parent workflow using the Set Variable action.

Variable	Description
`${ops_trigger_file_name}`	Resolves to the remote file name.
`${ops_trigger_file_name_nopath}`	Resolves to the remote file name without any path information.
`${ops_trigger_file_name_simple}`	Resolves to the base file name.
`${ops_trigger_file_name_extension}`	Resolves to the file extension of the file.
`${ops_trigger_file_path}`	Resolves to the directory where the remote file is located, but not the file itself. ${ops_trigger_file_path} is an alias of ${ops_trigger_file_fullpath_no_separator}
`${ops_trigger_file_fullpath}`	Resolves to the directory where the remote file is located, but not the file itself; includes the final directory separator.
`${ops_trigger_file_fullpath_no_separator}`	Resolves to the directory where the remote file is located, but not the file itself; does not include the final directory separator.

`${ops_trigger_files}`	Resolves to a comma-separated list of files that matched the wildcard, if one was specified in the Remote Filename field in the FTP File Monitor task. For example:
	`ops_trigger_files=COMPANY-2011-11-22.xls,`
	`COMPANY-2011-11-23.xls,COMPANY-2011-11-24.xls`

`${ops_trigger_wildcard}`	Resolves to the contents of the Remote Filename field from the task definition. For example:
	`ops_trigger_wildcard=/home/prod/stonebranch/`
	`COMPANY*.xls`

`${ops_trigger_wildcard_path}`	Resolves to the path only, with the final slash but without the file name, from the Remote Filename field in the task definition. For example: `ops_trigger_wildcard_path=/home/prod/ stonebranch/`
`${ops_trigger_wildcard_path_no_separator}`	Resolves to the path only, without the final slash and without the file name, from the Remote Filename field in the task definition. For example: `ops_trigger_wildcard_path_no_separator= /home/prod/stonebranch`

Task Monitor Variables

When the conditions of a Task Monitor task are met and its associated Task Monitor trigger launches one or more tasks, the built-in variables described below are passed into the task instances being launched by the trigger. For example, the Task Monitor trigger may specify an Email task that will launch each time the conditions in the associated Task Monitor task are met. You might want to specify one or more of these variables in the body of the email. Or, if the Task Monitor task is not associated with a trigger but is running within a workflow, on completion you can propagate one or more of these built-in variable values to the parent workflow level by using the Set Variable action. This allows you to pass information from the Task Monitor task to a successor task within the same workflow hierarchy.

Variable	Description
`${ops_trigger_task_name}`	Resolves to the task instance name that fired the trigger.
`${ops_trigger_task_id}`	Resolves to the task instance sys_id that fired the trigger.
`${ops_trigger_task_status}`	Resolves to the task instance status that fired the trigger.
`${ops_trigger_task_type}`	Resolves to the task instance type that fired the trigger.

Script Variables

For Windows, Linux/Unix, and SAP tasks where a Script or SAP Definition from the Script Library is specified, the following built-in variables are passed into the task instance. You can use these in a Windows, Linux/Unix, or SAP task action or propagate one or more of these built-in variable values to the parent workflow using the Set Variable action.

Variable	Description
`${ops_script_id}`	Resolves to the Opswise system ID of the script.
`${ops_script_name}`	Resolves to the Opswise name of the script.

SAP Task Variables

For an SAP task instance, where applicable, the following built-in variables resolve to the SAP jobname and SAP jobid of the job running in the SAP system. If you need to use the SAP jobname and/or the SAP jobid from one SAP task instance in a successor SAP task instance, you can use the Set Variable action to propagate these built-in variable values to the parent workflow.

Variable	Description
`${ops_sap_jobid}`	Resolves to the SAP job ID.
`${ops_sap_jobname}`	Resolves to the SAP job name.

Application Monitor Variables

When a task is launched by an Application Monitor trigger, the built-in variables described below are passed into the task being launched by the trigger:

Variable	Description
`${ops_trigger_appl_name}`	Resolves to the name of the Application being monitored by the trigger.
`${ops_trigger_appl_status}`	Resolves to the status of the Application being monitored by the trigger.
`${ops_trigger_appl_type}`	Resolves to the type of Application being monitored by the trigger, as defined by the Application Type field.
`${ops_trigger_appl_id}`	Resolves to the sys_id of the application.

SQL and Stored Procedure Variables

The following built-in variables are used in SQL tasks and Stored Procedure Task tasks to collect SQLException data, if any:

Field Name	Description
`${ops_sql_error_msg}`	Resolves to any error message generated by the database.
`${ops_sql_state}`	Resolves to a return code that indicates the outcome of the most recently executed SQL statement.

Agent Variables

All agent variables listed below can be used to pass information into an agent notification. Several of the agent variables can also be used to pass agent information into an agent-based task (Windows, Linux/Unix, z/OS, and SAP). Those supported for use in tasks are noted below.

Field Name	Description
`${ops_agent_hostname}`	Resolves to the agent hostname. You can also use this variable in task notifications; see Creating Email Notifications and Creating SNMP Notifications.
`${ops_agent_id}`	Resolves to the agent queue name. You can also use this variable in task notifications; see Creating Email Notifications and Creating SNMP Notifications.
`${ops_agent_ipaddr}`	Resolves to the agent IP address. Or, you can use ${ops_agent_ip}.
`${ops_agent_ip}`	Resolves to the agent IP address. You can also use this variable in task notifications; see Creating Email Notifications and Creating SNMP Notifications.
`${ops_agent_name}`	Resolves to the agent name. You can also use this variable in task notifications; see Creating Email Notifications and Creating SNMP Notifications.
`${ops_agent_mode}`	Resolves to the agent operational mode (Active, Offline).

Core Processor Variables

The following core processor (also referred to as cluster node) variables allow you to pass information into a cluster node (server) notification:

Field Name	Description
`${ops_cluster_start_time}`	Resolves to the date and time the cluster node (server) was started. Example: `ops_cluster_start_time = 2011-09-26 17:35:01 -0400`
`${ops_cluster_id}`	Resolves to the cluster node's internally-generated build ID. Example: `ops_cluster_id = MACHINEC19A:8080-opswise`
`${ops_cluster_uptime}`	Resolves to the numbers of days, hours, and minutes that this cluster node has been running since it was last started. Example: `ops_cluster_uptime = 7 Seconds`
`${ops_cluster_name}`	Resolves to the URL, or Node ID, of this cluster node. Example: `ops_cluster_name = MACHINEC19A:8080-opswise`

`${ops_cluster_hostname}`	Resolves to the hostname of this cluster node. Example: `ops_cluster_hostname = MACHINEC19A`
`${ops_cluster_ipaddr}`	Resolves to the IP address of this cluster node. Example: `ops_cluster_ipaddr = 10.N.N.NN`
`${ops_cluster_mode}`	Resolves to the current mode of this cluster node. One of the following: Offline, Active, Passive/Available, Passive/Unavailable. For more information, see Viewing Node Status. Example: `ops_cluster_mode = Active`

Connector Variables

The connector variables below allow you to pass information into a connector notification. A connector is an Opswise hub or transporter.

Field Name	Description
`${ops_connector_hostname}`	Resolves to the connector hostname.
`${ops_connector_id}`	Resolves to the connector queue name.
`${ops_connector_ipaddr}`	Resolves to the connector IP address.
`${ops_connector_name}`	Resolves to the connector name.
`${ops_connector_mode}`	Resolves to the connector operational mode (Active, Offline).

Launching With Variables

See the Launch With Variables feature on the Manually Running and Controlling Tasks page.

Trigger With Variables

See Triggering with Variables.

Setting Variables in a Workflow

Introduction

Within a workflow, you can specify instructions that set a variable to a specific value for use within the workflow. The value you set using this method exists only in memory for the period this workflow is running or until another Set Variable instruction sets it to another value. You can create the Set Variable instruction at the workflow level or task level; that is, you can attach the Set Variable action to the workflow or a task running within the workflow. If you set it at the workflow level, you have the option of setting it based on information in the workflow record, on any of the tasks (children) in the workflow, or both. If you set it at the task level, you have the option of setting it based on information in the specific task, in the parent workflow, or both.

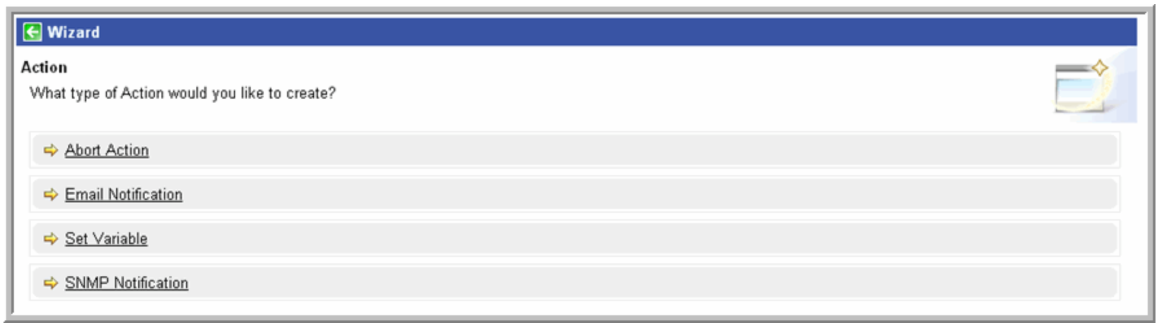
The variable need not exist in the database; you can create a new variable using the Set Variable instruction.

When creating a Set Variable instruction, you can trigger the Set Variable action based on one or more of the following:

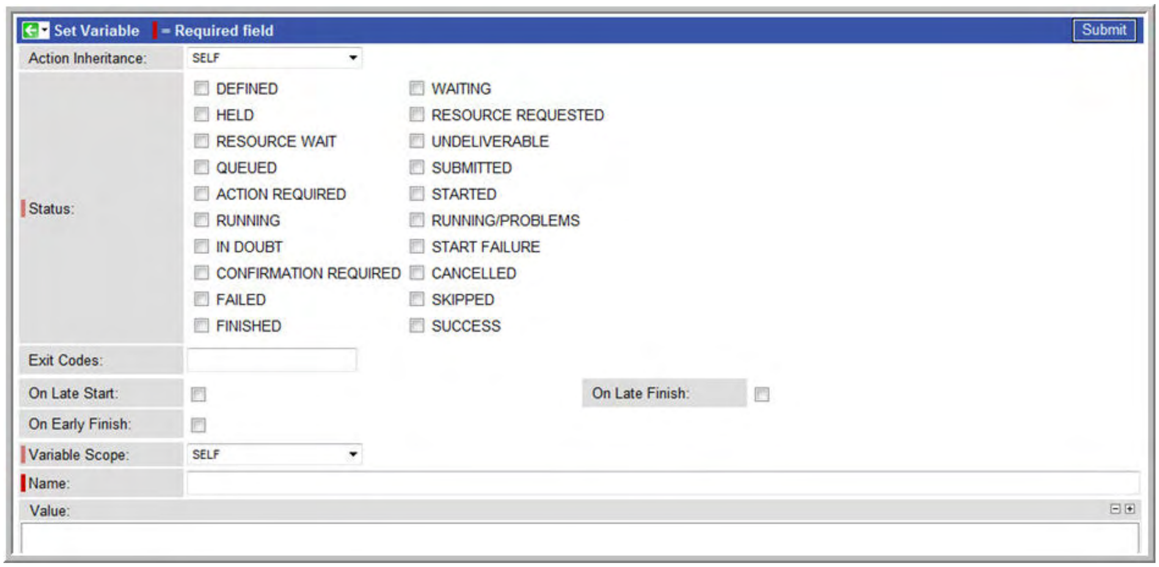
- Status
- Exit codes
- Late start
- Late or early finish

Creating a Set Variable Instruction

1. Display the workflow for which you want to create Set Variable instructions.
2. Click on the **Actions** tab. This tab displays a list of all Actions for this task, including Set Variables, Email and SNMP Notifications, and Abort Actions.
3. Click **New**. The Actions wizard displays.



4. Click **Set Variable**. The Set Variables screen displays.



Action Inheritance: SELF

Status:

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

Exit Codes:

On Late Start: □ On Late Finish: □

On Early Finish: □

Variable Scope: SELF

Name:

Value:

5. Using the field descriptions provided below as a guide, complete the fields as needed.
6. Click the **Submit** button to save the record and return to the Actions list, or, right-click and select **Save** to save the record and remain on the current display.
7. If appropriate, repeat these steps for any additional Set Variable instructions you want to add.

Set Variables Field Descriptions

The table below describes the fields and buttons on the Set Variables screen.

Field Name	Description
Action Inheritance	Workflow tasks only. Specifies what records these instructions apply to. Options: • SELF - These instructions apply only to the workflow and are not inherited by its children tasks. • SELF/CHILDREN - These instructions apply to the workflow and its contained tasks (children). • CHILDREN - These instructions apply only to the tasks within the workflow (children).

<table>
<tr><td>Status</td><td>The status(es) that will trigger the action. To trigger a Set Variable action, you can specify status only, or status and exit code. You can specify as many statuses as needed. Options:
<table>
<tr><th>Status</th><th>Description</th></tr>
<tr><td><b>Defined</b></td><td>All task types. The new task instance has been created (the task has been launched). Not yet implemented.</td></tr>
<tr><td><b>Waiting</b></td><td>All task types. The task has been loaded by a workflow and is waiting to run.</td></tr>
<tr><td><b>Held</b></td><td>All task types. The task has been put on hold by a user.</td></tr>
<tr><td><b>Resource Requested</b></td><td>All tasks with a virtual resource defined go immediately to a status of Resource Requested. If the resource is available, the task then moves to the next appropriate processing status.</td></tr>
<tr><td><b>Resource Wait</b></td><td>All tasks with a virtual resource defined go immediately to a status of Resource Requested. If the resource is not available, the task goes to a status of Resource Wait. When the resource becomes available, the task moves to the next appropriate processing status</td></tr>
<tr><td><b>Undeliverable</b></td><td>Agent-based tasks. The agent is unavailable.</td></tr>
<tr><td><b>Queued</b></td><td>Agent-based tasks only. The task has been queued on a resource.</td></tr>
<tr><td><b>Submitted</b></td><td>z/OS only. The task has been submitted to the z/OS Job Entry subsystem and scheduled by the z/OS Job Scheduler.</td></tr>
<tr><td><b>Action Required</b></td><td>Manual tasks only. When a manual task launches, it goes into Action Required status, meaning a user must perform some manual activity. For details, see Manual task.</td></tr>
<tr><td><b>Started</b></td><td>Agent-based and Manual tasks only. The task has started. For agent-based tasks, this means the agent has received the task.</td></tr>
<tr><td><b>Running</b></td><td>All task types. The task is running. For agent-based tasks, the agent has started running the program.</td></tr>
<tr><td><b>Running Problems</b></td><td>Workflows only. One or more tasks within the workflow has one of the following statuses:<ul><li>Held</li><li>Undeliverable</li><li>Running Problems (for sub-workflows)</li><li>Cancel Pending</li><li>In Doubt</li><li>Start Failure</li><li>Cancelled</li></ul></td></tr>
<tr><td><b>In Doubt</b></td><td>Agent-based tasks only. The agent is "in doubt" about the current status of the task instance. This may occur if an agent or agent connection goes down. In this case, the agent restarts and reviews its data about tasks in progress. If the agent finds a task still running, it resumes normal monitoring. If the agent cannot find the task, this usually indicates that the task completed, but the agent considers the task status to be "in doubt."</td></tr>
<tr><td><b>Start Failure</b></td><td>All task types. The task was unable to start.</td></tr>
<tr><td><b>Confirmation Required</b></td><td>z/OS only. If you make JCL changes and restart a z/OS task, Opswise will put the task into Confirmation Required status and prompt you for a confirmation. For detailed processing steps, see Rerunning a z/OS Task .</td></tr>
<tr><td><b>Cancelled</b></td><td>All task types. The task was cancelled by a user.</td></tr>
<tr><td><b>Failed</b></td><td>All task types. The task ran to a failure status.</td></tr>
<tr><td><b>Skipped</b></td><td>All task types. The task was skipped by a user.</td></tr>
<tr><td><b>Finished</b></td><td>All task types. The task was forced by the user to finish. The user may do this in cases where the task had "Cancelled" or "Failed" status, and the user needed to release other task instances depending on the successful completion of this task instance in a workflow. For more information, see Force Finishing a Task.</td></tr>
<tr><td><b>Success</b></td><td>All task types. The task has completed successfully.</td></tr>
</table>
</td></tr>
<tr><td>Exit Codes</td><td>Specifies one or more exit codes that will trigger the event. If you specify an exit code, you must also specify at least one status. Use commas to separate multiple exit codes; use a hyphen to specify a range. Example: 1, 5, 22-30.</td></tr>
<tr><td>On Late Start</td><td>Generate the action or notification if the task started late, based on the Late Start Time specified in the task.</td></tr>
</table>

On Late Finish	Generate the action or notification if the task finishes late, based on the Late Finish time specified in the task.
On Early Finish	Generate the action or notification if the task finishes early, based on the Early Finish Time specified in the task.
Variable Scope	Applies to variables associated with a task in a workflow. Options: • SELF - The variable is only set within the scope of the task that executes the Set Variable action. • PARENT - The variable is set within the scope of the (immediate) parent workflow. After it is set, any task within the parent workflow can access that variable. • TOP_LEVEL_PARENT - The variable is set within the scope of the top level parent. Example: Workflow A contains workflow B and workflow B contains workflow C. If a Set Variable action is executed by a task within workflow C with Variable Scope set to TOP_LEVEL_PARENT, then the variable will be set in workflow A's scope. This means that after it is set, tasks in workflow A, workflow B and workflow C can access that variable.
Name	Required. Name of variable. Up to 40 alphanumerics. The name must begin with an alphabetic character and can consist of: alphas (a-z, A-Z), numerics 0-9, _ (underscore). White spaces are not permitted; names are not case-sensitive. **Important Note** – Do not define Opswise variables with the prefix "ops_". That prefix is reserved for built-in variables.
Value	Optional. The value of the variable.
Description	Optional. Description of this variable.
Submit button	Submits the new record to the database.
Update button	Saves updates to the record.
Delete button	Deletes the current record.

Listing and Setting Variables From the Command Line

See the ops-variable-list and ops-variable-set CLI commands on the Command Line Interface Reference page.

Functions

Introduction

Opswise supports a number of functions that can be specified in free text fields. They are resolved when a task instance runs or when a Set Variable action containing a function is executed.

Functions are entered using the following formats:

```
${_function}
${_function(arg1,...,argN)}
```

Formatting Rules

- Functions must be written either in all lower case or exactly as shown in the table below.
- Any parameter can be quoted. Strings must be quoted with single or double quotes.
- All functions allow nesting to one level. That is, a function can be an argument to another function.
  A nested function has the format:

  ```
  ${__function(arg1,...,argN)}
  ```

- For nested functions, make sure you use a double underscore preceding the function name.
  For example:

  ```
  ${_substring("${ops_trigger_file_name_simple}","${__indexOf
  ("${ops_trigger_file_name_simple}","-")}")}
  ```

Table of Functions

The following table lists and describes each supported function.

Format	Description
`${_currentTimeMillis}`	Resolves to the current time in milliseconds.

`${_date([format, day_offset, hour_offset, minute_offset])}`	Resolves to the current date and time. All parameters are optional.
	Parameters:
	• format - Date format. The default format is yyyy-MM-dd HH:mm:ss Z. For details on the format parameter, go to: http://docs.oracle.com/javase/6/docs/api/java/text/SimpleDateFormat.html
	• day_offset - Day offset.
	• hour_offset - Hour offset.
	• minute_offset - Minute offset.
	Examples
	`${_date} --> 2008-07-14 12:43:06 -0400`
	`${_date()} --> 2008-07-14 12:43:06 -0400`
	`${_date("yyyy-MM-dd",5)} --> 2008-07-19`
	`${_date("yyyy-MM-dd HH:mm:ss",-2,-1)}`
	`  --> 2008-07-12 11:43:06`
	`${_date("",0,0,10)} --> 2008-07-14 12:53:06 -0400`

`${_dateadv([format, year_offset, month_offset, day_offset, hour_offset, minute_offset])}`	Resolves to the current date and time. All parameters are optional.
	Parameters:
	• format - Date format. The default format is yyyy-MM-dd HH:mm:ss Z. For details on the format parameter, go to: http://docs.oracle.com/javase/6/docs/api/java/text/SimpleDateFormat.html
	• year_offset - Year offset.
	• month_offset - Month offset.
	• day_offset - Day offset.
	• hour_offset - Hour offset.
	• minute_offset - Minute offset.
	Examples:
	`${_dateadv} --> 2008-07-29 09:31:42 -0700`
	`${_dateadv("yyyy-MMM",-1)} --> 2007-Jul`
	`${_dateadv("yyyy-MMM",0,-1)}`
	`  --> 2008-Jun`

`${_guid}`	Resolves to a 32 byte GUID (Globally Unique ID).
`${_hostname}`	Resolves to the hostname of the machine running the Core, if available.
`${_indexOf(value, str)}`	Returns the index within the string value of the first occurrence of the specified substring, str. Parameters: • value - Any string. • str - Substring to search for. If the str argument occurs as a substring within the value, then the index of the first character of the first such substring is returned; if it does not occur as a substring, \-1 is returned. Indexing functions use zero-based numbering; that is, the initial element is assigned the index 0.

`${_indexOfWithOffset(value, str, offset)}`	Returns the index within this string of the first occurrence of the specified substring plus the specified offset. The integer returned is the smallest value. Parameters: - value - Any string. - str - Substring to search for. If the str argument occurs as a substring within the value, then the index of the first character of the first such substring is returned; if it does not occur as a substring, -1 is returned. - offset - Number (positive or negative) to offset the found index. ⚠ Indexing functions use zero-based numbering; that is, the initial element is assigned the index 0.
`${_ipaddress}`	Resolves to the IP address of the machine running the Core.
`${_lastIndexOf(value, str)}`	Returns the index within the string value of the rightmost occurrence of the specified substring, str. Parameters: - value - Any string. - str - Substring to search for. If the str argument occurs one or more times as a substring within the value, then the index of the first character of the last such substring is returned. If it does not occur as a substring, -1 is returned. ⚠ Indexing functions use zero-based numbering; that is, the initial element is assigned the index 0.
`${_lastIndexOfWithOffset (value, str, offset)}`	Returns the index within this string of the rightmost occurrence of the specified substring, plus the specified offset. The returned index is the largest value. Parameters: - value - Any string. - str - Substring to search for. If the str argument occurs as a substring within the value, then the index of the first character of the first such substring is returned; if it does not occur as a substring, -1 is returned. - offset - Number (positive or negative) to offset the found index. ⚠ Indexing functions use zero-based numbering; that is, the initial element is assigned the index 0.
`${_random([max, min])}`	Generates a random number between max (inclusive) and min (inclusive). Parameters: - max - Upper bound (inclusive) on the random number (default 9). - min - Lower bound (inclusive) on the random number (default 0).

`${_replaceAll(value, regex, replacement)}`	Replaces each substring of value that matches the specified regular expression, regex, with the specified replacement. Parameters: • value - Input string. • regex - Regular expression. • replacement - Replacement string.
`${_resolve(variable_name, default_value)}`	Resolves the variable specified by name variable name and substitutes the default value if the variable cannot be resolved. Parameters: • variable_name - Variable name. • default_value - Default value to use if the variable cannot be resolved.
`${_resultsAll([separator])}`	Returns all SQL results from the current SQL or Stored Procedure task. Columns are separated by the specified separator and rows are separated by a new line. Parameters: • separator - Column separator (default = comma).
`${_resultsAllFromTask(name [, separator])}`	Returns all SQL results from a sibling SQL or Stored Procedure task, within the same workflow. Columns are separated by the specified separator and rows are separated by a new line. Parameters: • name - Name of the sibling task that the results should come from. The task must be within the same workflow. • separator - Column separator (default = comma).
`${_resultsColumn(name, colname[, rownum, default_value])}`	Returns the string value of a row/column from a previously executed SQL task within the same workflow, or from the current SQL task. Parameters: • name - Name of a sibling SQL task within the same workflow from which you want the function to fetch results. If you want to execute the function against the current taskUse an empty string for the name parameter. • colname - Name of column to retrieve. • rownum - Numeric row number in result set to retrieve (default 1). • default_value - Default value to return if result not found.
`${_resultsColumnByNo (name, colnum[, rownum, default_value])}`	Returns the string value of a row/column from a previously executed SQL task within the same workflow, or from the current SQL task. Parameters: • name - Name of a sibling SQL task within the same workflow from which you want the function to fetch results. If you want to execute the function against the current taskUse an empty string for the name parameter. • colnum - Number of column to retrieve. First column in result is 1, 2nd is 2, and so on. • rownum - Numeric row number in result set to retrieve (default 1). • default_value - Default value to return if result not found.

`${_resultsColumnsCSV (name[, rownum])}`	Returns the string values of columns in a specific row in CSV (comma-separated values) format, from a previously executed SQL task within the same workflow, or from the current SQL task. Parameters: • name - Name of a sibling SQL task within the same workflow from which you want the function to fetch results. If you want to execute the function against the current task, use an empty string for the name parameter. • rownum - Numeric row number in result set to retrieve (default 1).

`${_scope}`	This displays all the defined and built-in variables associated with the task instance. Example:
	`${_scope} --> {ops_workflow_id=, ops_task_type=Unix,`
	`ops_status=DEFINED, ops_retry_interval=60,`
	`ops_exit_code=0, ops_retry_maximum=0, ops_cmd_parms=,`
	`ops_cmd=ls -la; exit ${_random('9')};, ops_retry_count=0,`
	`ops_agent_id=67e4994143d2617201cdf4ba9df9ab0a,`
	`ops_task_id=84880af243d26172019aa1d25988a8f9,`
	`ops_task_name=Opswise - Linux Ls}`

`${_siblingid(sibling_name)}`	Resolves to the sys_id of the first task instance found within the same workflow specified by the sibling name.
	Parameters:
	• sibling_name - Sibling name.
	Example:
	`${_siblingid("Sleep 60")}`
	`  --> 5dbaaab943d26172015e10ab3e894e10`

`${_SQLWarnings([separator])}`	Returns all SQL warnings from the current SQL or Stored Procedure task. Columns are separated by the specified separator and rows are separated by a new line. Parameters: • separator - Column separator (default = comma).
`${_SQLWarningsFromTask(name [, separator])}`	Returns all SQL warnings from a sibling SQL or Stored Procedure task, within the same workflow. Columns are separated by the specified separator and rows are separated by a new line. Parameters: • name – Name of the sibling task that the warnings should come from. The task must be within the same workflow. • separator - Column separator (default = comma).

`${_substring(value, begin_index[, end_index])}`	Returns a new string that is a substring of value. The substring begins at the specified begin_index and extends to the character at index end_index -1.
	Parameters:
	• value - String to make a substring from.
	• begin_index - Beginning index, inclusive.
	• end_index - Ending index, exclusive.
	Examples:
	`${_substring("hamburger", 4, 8)}`
	`  resolves to "urge".`
	`${_substring("smiles", 1, 5)}`
	`  resolves to "mile".`
	⚠ Indexing functions use zero-based numbering; that is, the initial element is assigned the index 0.

`${_toLowerCase(value)}`	Converts all of the characters in the value to lower case using the rules of the default locale. Parameters: • value - String to convert to lower case.
`${_toUpperCase(value)}`	Converts all of the characters in the value to upper case using the rules of the default locale. Parameters: • value - String to convert to upper case.
`${_trim(value)}`	Returns a copy of value, with leading and trailing whitespace omitted. Parameters: • value - String to trim.
`${_varLookup(sibling_name, variable_name[,def])}`	Locates the specified variable in the specified sibling task instance within the same workflow and resolves to the variable value. Parameters: • sibling_name - Name of the sibling task instance from which the function is collecting the variable value. • variable_name - Name of the variable being collected by the function. • def - Default value to return if the variable is not defined in the sibling task instance. (Optional.)

Creating Triggers

Triggers

A trigger specifies times and/or events that trigger one or more tasks.

When a trigger is satisfied, Opswise launches the tasks specified in the trigger. Each trigger can have an unlimited number of tasks associated with it. All the specified tasks are run each time the trigger is satisfied. If you want to specify dependencies such as "run TaskB only if TaskA fails," create a Workflow, which is a series of inter-connected tasks. For more information, see Workflows.

A built-in variable is available for returning the trigger name; see Trigger Variable. Additional built-in variables are supported for specific trigger types; see the Built-In Variables page or documentation for each trigger type for details.

You can use triggers to launch tasks based on the following:

- Specific date and time, or a series of dates and times (Time, Cron, and Temporary triggers)
- Creation, deletion, change, existence or non-existence of a file on a particular machine (File Monitor trigger)
- Status of specific tasks (Task Monitor trigger)
- Status of one or more application resources (Application Monitor trigger)

A Manual trigger is supported that allows you to manually launch a task while providing temporary values for variables used in the task.

Accessing a Triggers List

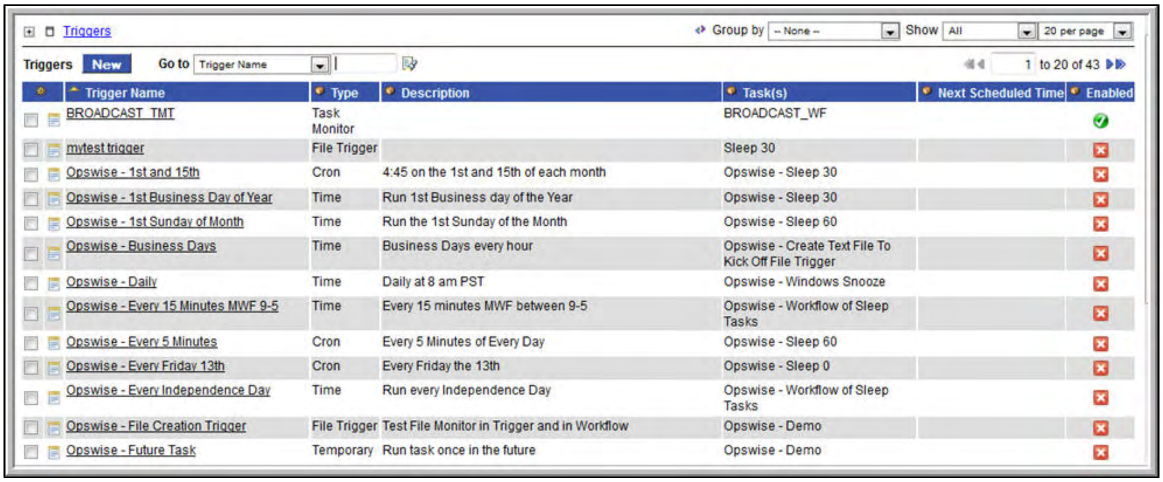
Opswise provides two lists of triggers:

- All triggers
- Active triggers

Active Triggers are triggers that have their Enabled flag set.

To access either list, select **Automation Center > Triggers > All Triggers** or **Automation Center > Triggers > Active Triggers**, as appropriate, from the navigation pane.

The following illustrates an All Triggers screen:



Trigger Name	Type	Description	Task(s)	Next Scheduled Time	Enabled
BROADCAST_TMT	Task Monitor		BROADCAST_WF		✓
mytest trigger	File Trigger		Sleep 30		☒
Opswise - 1st and 15th	Cron	4:45 on the 1st and 15th of each month	Opswise - Sleep 30		☒
Opswise - 1st Business Day of Year	Time	Run 1st Business day of the Year	Opswise - Sleep 30		☒
Opswise - 1st Sunday of Month	Time	Run the 1st Sunday of the Month	Opswise - Sleep 60		☒
Opswise - Business Days	Time	Business Days every hour	Opswise - Create Text File To Kick Off File Trigger		☒
Opswise - Daily	Time	Daily at 8 am PST	Opswise - Windows Snooze		☒
Opswise - Every 15 Minutes MWF 9-5	Time	Every 15 minutes MWF between 9-5	Opswise - Workflow of Sleep Tasks		☒
Opswise - Every 5 Minutes	Cron	Every 5 Minutes of Every Day	Opswise - Sleep 60		☒
Opswise - Every Friday 13th	Cron	Every Friday the 13th	Opswise - Sleep 0		☒
Opswise - Every Independence Day	Time	Run every Independence Day	Opswise - Workflow of Sleep Tasks		☒
Opswise - File Creation Trigger	File Trigger	Test File Monitor in Trigger and in Workflow	Opswise - Demo		☒
Opswise - Future Task	Temporary	Run task once in the future	Opswise - Demo		☒

The following table provides column descriptions for the default column display.

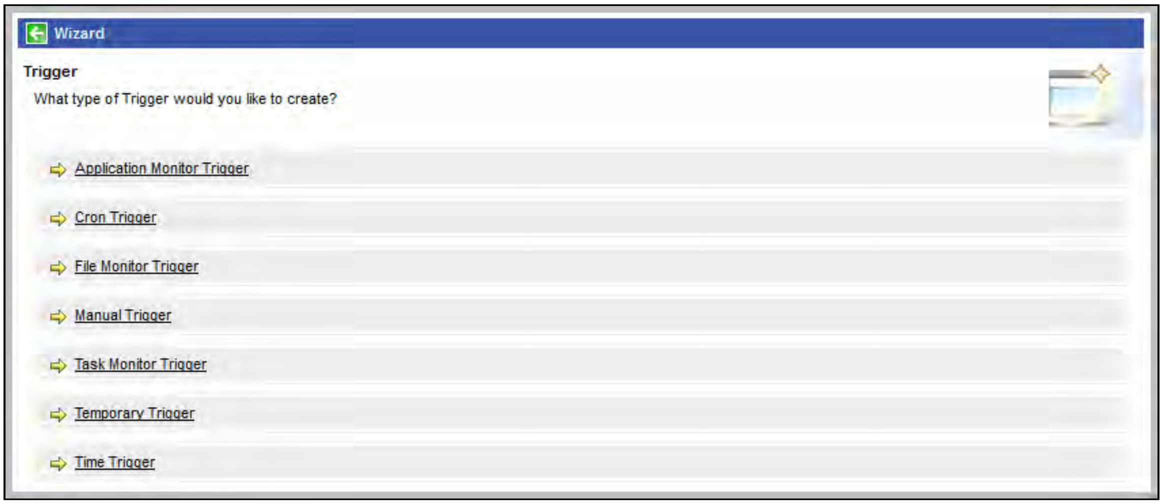
For information about customizing this list, including filtering, sorting, searching, and other list features, see Using Lists.

Field Name	Description
Trigger Name	User-defined. Name used within Opswise to identify this trigger. It can contain a maximum of 40 alphanumeric characters. It is the user's responsibility to develop a workable naming scheme for triggers.
Type	User-defined. The type of trigger. Options: • CRON • Time • Temporary • Manual • File • Task Monitor • Application Monitor
Description	User-defined. Copied from the Description field in the trigger.
Task(s)	User-defined. The name of the task(s) being triggered when this trigger is satisfied. When selecting tasks from the definition screen, click on the lock icon to unlock the field and select tasks.
Next Scheduled Time	Supplied by Opswise. For time-based triggers, the next date and time this trigger will be satisfied. See Displaying Trigger Forecast Information.
Enabled	User-defined. Whether or not the Enabled field is checked. The user enables and disables the trigger by clicking the **Enable/Disable Trigger** buttons. Only enabled triggers are processed by Opswise.

Creating a Trigger

You can create a trigger either of two ways.

1. From the Navigation Pane:
   a. Select a trigger type from the navigation pane. The Trigger List screen for that trigger type displays.
   b. Click the **New** button. The Trigger Definition screen for that trigger type displays.
2. From the Trigger List screen:
   a. Click the New button on the Trigger List screen. The Trigger Wizard screen displays.



   b. Click the trigger type for trigger you want to create. The Trigger List screen for that trigger type displays.

For detailed information on creating triggers for specific trigger types, click that trigger type in the navigation panel on the left-hand side of this page.

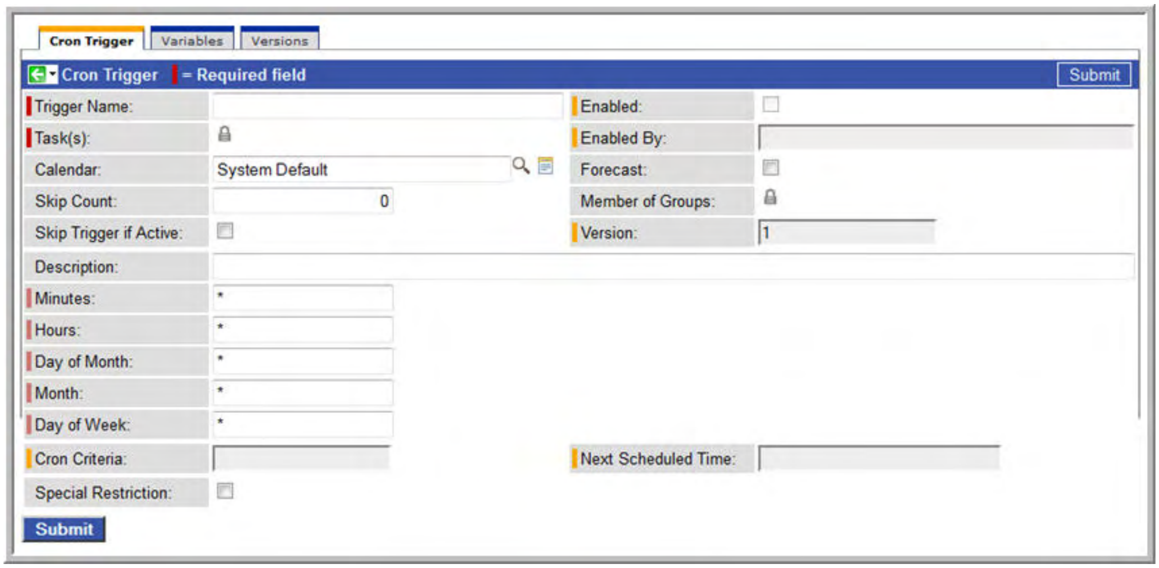
Cron Trigger

Overview

The Opswise Cron trigger uses standard Cron syntax. Once the Cron trigger is entered into the system, Opswise interprets it and processes it as it would any other trigger. The trigger is satisfied when the current date and time match all the values specified in the Minutes, Hours, Day of Month, Month, and Day of Week fields. Click here for details about how the system handles Daylight Savings Time.

Creating a New Cron Trigger

1. From the navigation pane, select **Automation Center > Triggers > Cron Triggers**. The Cron Triggers List displays.
2. Click **New**. The Cron Trigger Definition screen displays.



Trigger Name:		Enabled:	□
Task(s):		Enabled By:	
Calendar:	System Default	Forecast:	□
Skip Count:	0	Member of Groups:	
Skip Trigger if Active:	□	Version:	1
Description:			
Minutes:	*		
Hours:	*		
Day of Month:	*		
Month:	*		
Day of Week:	*		
Cron Criteria:		Next Scheduled Time:	
Special Restriction:	□		

3. Using the field descriptions provided below as a guide, complete the fields as needed.
4. Click the **Submit** button to save the record and return to the menu, or right-click the title bar and select **Save** to save the record and remain on the current display.
5. If appropriate, repeat these steps for any additional triggers you want to add.
6. Enable the trigger(s).

CRON Trigger Field Descriptions

Field Name	Description
Trigger Name	User-defined. Name used within Opswise to identify this trigger. It can contain a maximum of 40 alphanumeric characters. It is the user's responsibility to develop a workable naming scheme for triggers.
Enabled	User-defined. Whether or not the Enabled field is checked. The user enables and disables the trigger by clicking the **Enable/Disable Trigger** buttons. Only enabled triggers are processed by Opswise.

Task(s)	User-defined. The name of the task(s) being triggered when this trigger is satisfied. When selecting tasks from the definition screen, click on the lock icon to unlock the field and select tasks.
Enabled By	System-supplied. Displays the ID of the user who most recently enabled this trigger.
Forecast	Enabled or disabled by user. If enabled, the system calculates the date and time when this trigger will be satisfied for the next number days specified in the Forecast Period In Days property. The system writes the forecasting entries to the **Triggers > Forecasts** display. For details, see Displaying Trigger Forecast Information.
Member of Groups	User-defined. Allows you to select one or more Opswise groups that this record definition belongs to. Click on the lock icon to unlock the field and select groups.
Calendar	If the user selects **Special Restriction**, the calendar defines what the Holidays or Non Business days are. Type in a calendar name or click the magnifying glass to browse for an existing calendar or to add a new calendar. To display details about the calendar specified in this field, hover over the paper icon.
Skip Count	User-defined. Allows you to specify that Opswise skip the next *N* times this task is triggered.
Skip Trigger if Active	User-defined. Allows you to specify that Opswise skip the next run of the specified tasks if the previous run has not gone to a complete status (is still active).
Version	System-supplied. The version number of the current record, which is incremented by the system every time a user updates a record. Click on the Versions tab to view previous versions. For details, see Record Versioning.
Minutes	Time in minutes, using standard Cron syntax.
Hours	Time in hours, using standard Cron syntax. Click here for details about how the system handles Daylight Savings Time.
Day of Month	Day of the month, using standard Cron syntax.
Month	Month, using standard Cron syntax.
Day of Week	Day of the week, using standard Cron syntax.
Description	User-defined. Copied from the Description field in the trigger.
Cron Criteria	System-supplied. Provides a summary of the Cron specifications. Displays in the Cron Criteria field on the Triggers list.
Next Scheduled Time	Supplied by Opswise. For time-based triggers, the next date and time this trigger will be satisfied. See Displaying Trigger Forecast Information.

Special Restriction	Enable this field to specify additional parameters that tell the system how to handle exceptions such as when the trigger is satisfied on a holiday or non-business day. You can specify Simple or Complex Restrictions or both (see field descriptions below for details). For example, you may specify a Simple Restriction that disables the trigger if it is satisfied on a holiday. Or, you may chose Complex Restriction and specify more detailed parameters such do not trigger on the last business day of the month.
Simple Restriction	If enabled, allows you to specify an Action such as Do Not Trigger (see below) on a non-business day or holiday (see Situation field below). For example, do not trigger on a non business day.
Situation	If Simple Restriction is enabled, allows you to select the situation that causes the system to initiate the action specified in the Action field (see Action field below). Options: • On Non Business Day • On Holiday
Action	If Special Restriction is enabled, allows you to select an action to take on a non business day or holiday (see Situation field above). Options: • Do Not Trigger • Next Day (run on the next day) • Next Business Day (run on the next business day, as defined in the calendar) • Previous Day (run on the previous day) • Previous Business Day (run on the previous business day, as defined in the calendar)
Complex Restriction	If enabled, allows you to specify a set of parameters that determine one or more situations when this trigger should not be satisfied. Used in conjunction with the following fields: Restriction Mode, Restriction Adjective, Restriction Noun, Restriction Qualifier (see details below). For example, you may specify that you do not want to satisfy this trigger on the last business day of the year or the first day of each month.
Restriction Mode	If both Simple and Complex Restriction are enabled, specifies whether you want to use both restriction types (AND) or one or the other (OR). Options: • And • Or
Restriction Adjective	If Complex Restriction is enabled, the type of selection. Options: 1st, 2nd, 3rd, 4th, Last. Example: The **last** business day of the month.
Restriction Noun	If Complex Restriction is enabled, the day you want to select. Options: Sunday through Saturday, Day, or Business Day. Example: The last **business day** of the month.
Restriction Qualifier	If Complex Restriction is enabled, the period you are selecting from. Options: January through December, Month or Year. Example: The last business day of the **month**.
Submit button	Submits the new record to the database.
Update button	Saves updates to the record.
List Qualifying Times button	Opens a new browser tab and displays a list of the next 20 dates and times this trigger will be satisfied.

Enable Trigger button	Activates this trigger and writes your UserID to the Enabled By field.
Disable Trigger button	Deactivates this trigger.
Trigger Now button	Immediately triggers all the tasks specified in this trigger.
Delete button	Deletes the current record.
Variables tab	Displays all variables associated with this record.
Versions tab	Stores copies of all previous versions of the current record (see Record Versioning).

Generating a List of Qualifying Times

Opswise allows you to generate a list of future dates and times that this trigger will trigger the specified task. Click the **List Qualifying Times** button. Opswise opens a new browser tab and displays a list of the next 20 dates and times.

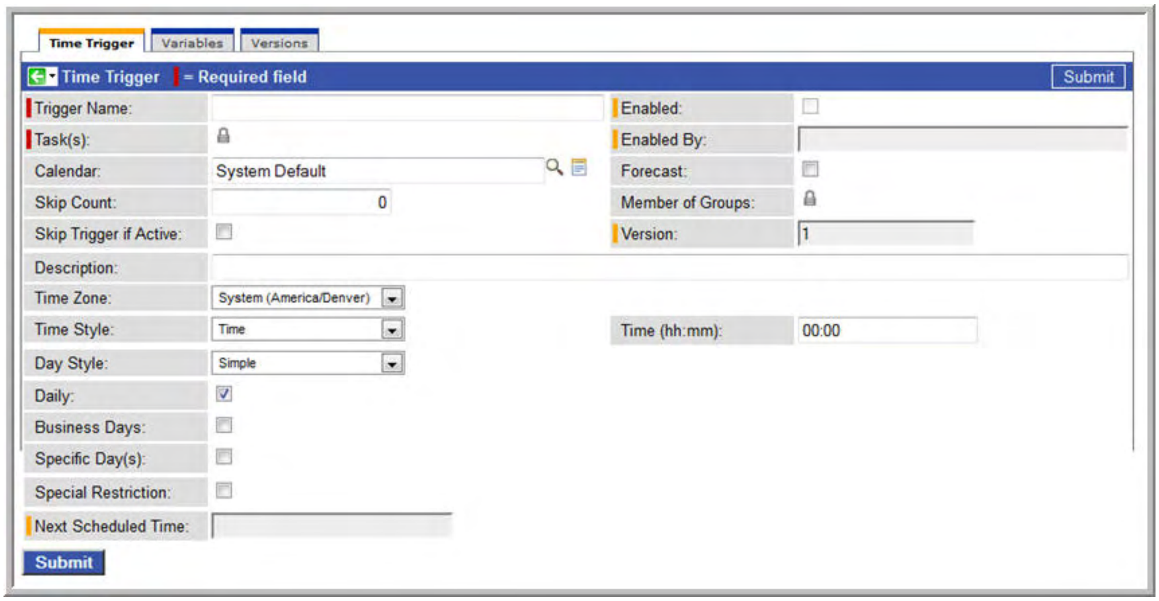
Time Trigger

Overview

The Time trigger allows you to specify dates and times at which a task will be triggered. You can define specific dates and times, such as "March 15 at 12:00 a.m.," or a series, such as "every hour of every Friday," or a mixture, such as "9 a.m. every Monday." You can specify simple date and time selection parameters, such as "every weekday at 12:00," or create more complex formulas such as "every 3 hours on the last business day of the year." Click here for details about how the system handles Daylight Savings Time.

Creating a New Time Trigger

1. From the navigation pane, select **Automation Center > Triggers > Time Triggers**. The Time Triggers List screen displays.
2. Click **New**. The Time Trigger Definition screen displays.



Trigger Name:
Task(s):
Calendar: System Default
Skip Count: 0
Skip Trigger if Active: □
Description:
Time Zone: System (America/Denver)
Time Style: Time
Day Style: Simple
Daily: ☑
Business Days: □
Specific Day(s): □
Special Restriction: □
Next Scheduled Time:
Enabled: □
Enabled By:
Forecast: □
Member of Groups:
Version: 1
Time (hh:mm): 00:00

3. Using the field descriptions provided below as a guide, complete the fields as needed.
4. Click the **Submit** button to save the record and return to the menu, or, right-click the title bar and select **Save** to save the record and remain on the current display.
5. If appropriate, repeat these steps for any additional triggers you want to add.
6. Enable the trigger(s).

Time Trigger Field Descriptions

Field Name	Description
Trigger Name	User-defined. Name used within Opswise to identify this trigger. It can contain a maximum of 40 alphanumeric characters. It is the user's responsibility to develop a workable naming scheme for triggers.
Enabled	User-defined. Whether or not the Enabled field is checked. The user enables and disables the trigger by clicking the **Enable/Disable Trigger** buttons. Only enabled triggers are processed by Opswise.

Task(s)	User-defined. The name of the task(s) being triggered when this trigger is satisfied. When selecting tasks from the definition screen, click on the lock icon to unlock the field and select tasks.
Enabled By	System-supplied. Displays the ID of the user who most recently enabled this trigger.
Calendar	If the user selects **Special Restriction**, the calendar defines what the Holidays or Non Business days are. Type in a calendar name or click the magnifying glass to browse for an existing calendar or to add a new calendar. To display details about the calendar specified in this field, hover over the paper icon.
Forecast	Enabled or disabled by user. If enabled, the system calculates the date and time when this trigger will be satisfied for the next number days specified in the Forecast Period In Days property. The system writes the forecasting entries to the **Triggers > Forecasts** display. For details, see Displaying Trigger Forecast Information.
Skip Count	User-defined. Allows you to specify that Opswise skip the next *N* times this task is triggered.
Member of Groups	User-defined. Allows you to select one or more Opswise groups that this record definition belongs to. Click on the lock icon to unlock the field and select groups.
Skip Trigger if Active	User-defined. Allows you to specify that Opswise skip the next run of the specified tasks if the previous run has not gone to a complete status (is still active).
Version	System-supplied. The version number of the current record, which is incremented by the system every time a user updates a record. Click on the Versions tab to view previous versions. For details, see Record Versioning.
Description	User-defined. Copied from the Description field in the trigger.
Time Zone	User-defined. Allows you to specify the timezone that will be applied to the time(s) specified in the trigger. For example, if you specify 23:00 and a time zone of Canada/Central, the task will run at 11:00 p.m. Central Canada time.
Time Style	Specifies whether this trigger is a specific time or a series of times. Options: • Time - Triggers the task at a specific time. Required field: Time. • Time Interval - Triggers the task at specific intervals of times. Required fields: Time Interval, Time Interval Units. Optional fields: Starting Time, Restrict Times. Click here for details about how the system handles Daylight Savings Time.
Time	Required if Time Style = Time. Specifies the time of the trigger in hours, minutes, and seconds, using 24-hour time. For example, 01:45:00 means trigger the task at 1:45 a.m.; 13:45:00 means trigger the task at 1:45 p.m. Click here for details about how the system handles Daylight Savings Time.
Time Interval	Required if Time Style = Time Interval. Specify a number indicating the number of Time Interval Units (see below). For example, for a Time Interval of every three hours, specify 3 in the Time Interval Field and select Hours in the Time Interval Units field.

Time Interval Units	Required if Time Style = Time Interval. Select a type of time interval. Used in conjunction with the Time Interval field (see above). For example, for a Time Interval of every three hours, specify 3 in the Time Interval Field and select Hours in the Time Interval Units field. Options are: • Minutes • Hours
Starting Time	Optional. If Time Style = Time Interval, you can specify a Starting Time at which time the trigger activates. Enable the Starting Time field and specify the Starting Time in the Starting At field (see below).
Starting At	For a trigger that is satisfied at intervals (for example, every 3 hours), and if Starting Time is enabled, use Starting At to specify the time at which the trigger activates. In this case, the trigger will first be be satisfied at the Starting At time, and again satisfied thereafter at the specified interval. Use 24-hour time.
Restrict Times	If Time Style = Time Interval, you can specify a period during which the trigger is active. Enable the Restrict Times field and specify the start and end times in the Enabled Start and Enabled End fields (see below).
Enabled Start	If Restrict Times is enabled, specify the start time of the period during which the trigger should be active. Use 24-hour time.
Enabled End	If Restrict Times is enabled, specify the end time of the period during which the trigger should be active. Use 24-hour time.
Day Style	Allows you to indicate whether this trigger will use the typically used calendar days (Simple) or whether you are specifying a formula for the selection of days (Complex). Options: • **Simple** - The trigger is active every day, on business days, or on one or more specific days, depending on what you select in the Daily, Business Days, and Custom Days fields (see below). • **Complex** - The trigger is active on one or more days selected by a formula specified using the Date Adjective, Date Noun, and Date Qualifier fields (see below).
Daily	If Date Style = Simple, you can select this field to specify that the trigger is active every day of the week.
Business Days	If Date Style = Simple, you can select this field to specify that the trigger is active on the business days specified in the calendar selected in the trigger's Calendar field.
Custom Days	If Date Style = Simple, you can select this field to specify that the trigger is active on specific days. Select one or more days by selecting the Sun, Mon, Tue, Wed, Thu, Fri, and Sat fields.
Date Adjective	If Day Style is Complex, you can use this field to specify which in a series of days you want to select. Used in conjunction with the Date Noun and the Date Qualifier fields. For example, to specify "the 1st business day of the month," select Date Adjective: 1st, Date Noun: Business Day, Date Qualifier: Month. Options: • Every • 1st • 2nd • 3rd • 4th • Last

Date Noun	If Day Style is Complex, you can use this field to specify the type of day you want to select. Used in conjunction with the Date Adjective and the Date Qualifier fields. For example, to specify "the 1st business day of the month," select Date Adjective: 1st, Date Noun: Business Day, Date Qualifier: Month. This drop-down menu is populated as follows: • Sunday through Saturday • Day = any day • Business Day = The business days specified in the calendar selected in the trigger's Calendar field. • The Custom Days specified in the calendar selected in the trigger's Calendar field.
Date Qualifier	If Day Style is Complex, you can use this field to specify the period for your selection formula. Used in conjunction with the Date Noun and the Date Adjective fields. For example, to specify "the 1st business day of the month," select Date Adjective: 1st, Date Noun: Business Day, Date Qualifier: Month. Options: • Month • Year • January through December
Special Restriction	Enable this field to specify additional parameters that tell the system how to handle exceptions such as when the trigger is satisfied on a holiday or non-business day. You can specify Simple or Complex Restrictions or both (see field descriptions below for details). For example, you may specify a Simple Restriction that disables the trigger if it is satisfied on a holiday. Or, you may chose Complex Restriction and specify more detailed parameters such do not trigger on the last business day of the month.
Simple Restriction	If enabled, allows you to specify an Action such as Do Not Trigger (see below) on a non-business day or holiday (see Situation field below). For example, do not trigger on a non business day.
Situation	If Simple Restriction is enabled, allows you to select the situation that causes the system to initiate the action specified in the Action field (see Action field below). Options: • On Non Business Day • On Holiday
Action	If Special Restriction is enabled, allows you to select an action to take on a non business day or holiday (see Situation field above). Options: • Do Not Trigger • Next Day (run on the next day) • Next Business Day (run on the next business day, as defined in the calendar) • Previous Day (run on the previous day) • Previous Business Day (run on the previous business day, as defined in the calendar)
Complex Restriction	If enabled, allows you to specify a set of parameters that determine one or more situations when this trigger should not be satisfied. Used in conjunction with the following fields: Restriction Mode, Restriction Adjective, Restriction Noun, Restriction Qualifier (see details below). For example, you may specify that you do not want to satisfy this trigger on the last business day of the year or the first day of each month.
Restriction Mode	If both Simple and Complex Restriction are enabled, specifies whether you want to use both restriction types (AND) or one or the other (OR). Options: • And • Or
Restriction Adjective	If Complex Restriction is enabled, the type of selection. Options: 1st, 2nd, 3rd, 4th, Last. Example: The **last** business day of the month.

Restriction Noun	If Complex Restriction is enabled, the day you want to select. Options: Sunday through Saturday, Day, or Business Day. Example: The last **business day** of the month.
Restriction Qualifier	If Complex Restriction is enabled, the period you are selecting from. Options: January through December, Month or Year. Example: The last business day of the **month**.
Next Scheduled Time	Supplied by Opswise. For time-based triggers, the next date and time this trigger will be satisfied. See Displaying Trigger Forecast Information.
Submit button	Submits the new record to the database.
Update button	Saves updates to the record.
List Qualifying Times button	Opens a new browser tab and displays a list of the next 20 dates and times this trigger will be satisfied.
Enable Trigger button	Activates this trigger and writes your UserID to the Enabled By field.
Disable Trigger button	Deactivates this trigger.
Trigger Now button	Immediately triggers all the tasks specified in this trigger.
Delete button	Deletes the current record.
Variables tab	Displays all variables associated with this record.
Versions tab	Stores copies of all previous versions of the current record (see Record Versioning).

Generating a List of Qualifying Times

Opswise allows you to generate a list of future dates and times that this trigger will trigger the specified task. Click the **List Qualifying Times** button. Opswise opens a new browser tab and displays a list of the next 20 dates and times.

Manual Trigger

Overview

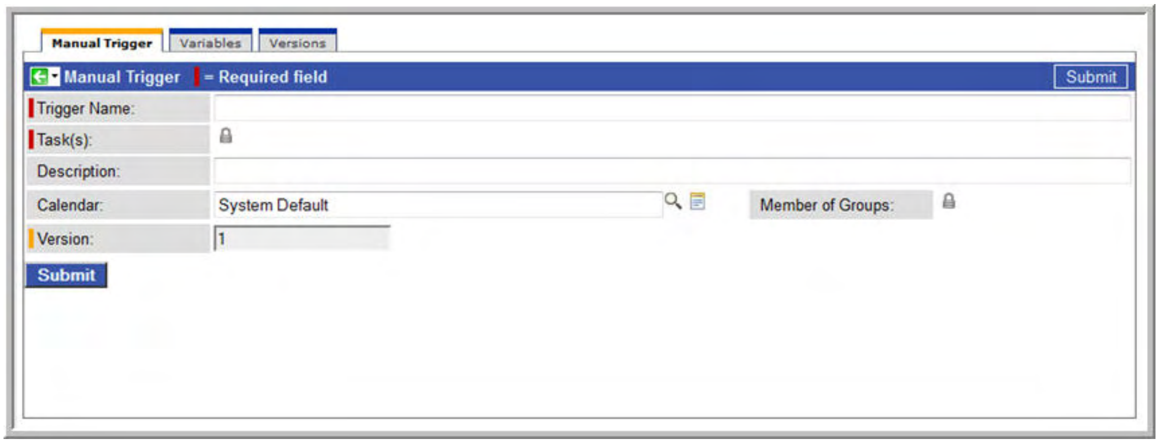
The Manual trigger allows you to launch a task(s) immediately, while setting or overriding the value of one or more user-defined variables specified in the task. You will use this trigger if you want to manually launch a task but cannot use the Launch Task or Trigger Now buttons because you need to set or override one or more variables. For example, you might choose to use a Manual trigger to launch a "generic" workflow that you run occasionally with a slight variation in specific details. In this case, you will launch the workflow and pass in the appropriate details using variables.

You can use the Manual trigger to set values to pre-existing variables or create new variables. The variable values you enter here override all others. However, the change in value only persists while this launched task instance(s) is running. Future executions of the task(s), unless they are launched by a Manual trigger, will use the standard methods for resolving user-defined variables.

The audit message created when you use Manual trigger is the same as Trigger Now.

Creating a New Manual Trigger and Triggering One or More Tasks

1. From the navigation pane, select **Automation Center . triggers > Manual Triggers**. The Manual Triggers List screen displays.
2. Click **New**. The Manual Trigger Definition screen displays.



Calendar: System Default

Version: 1

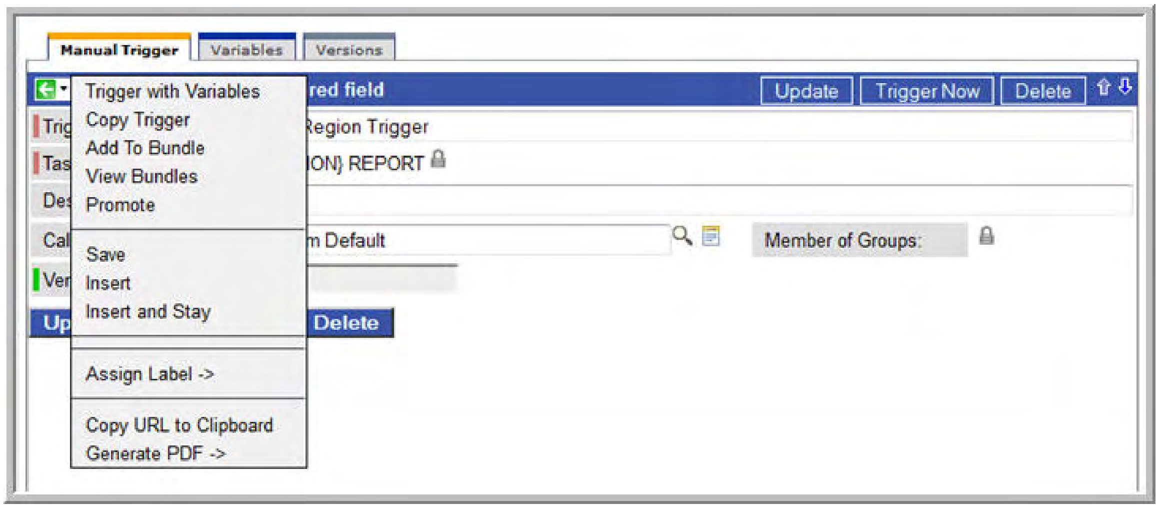
3. Using the field descriptions provided below as a guide, complete the fields as needed.
4. Right-click the title bar and select **Save** to save the record and remain on the current display.

Entering Variables

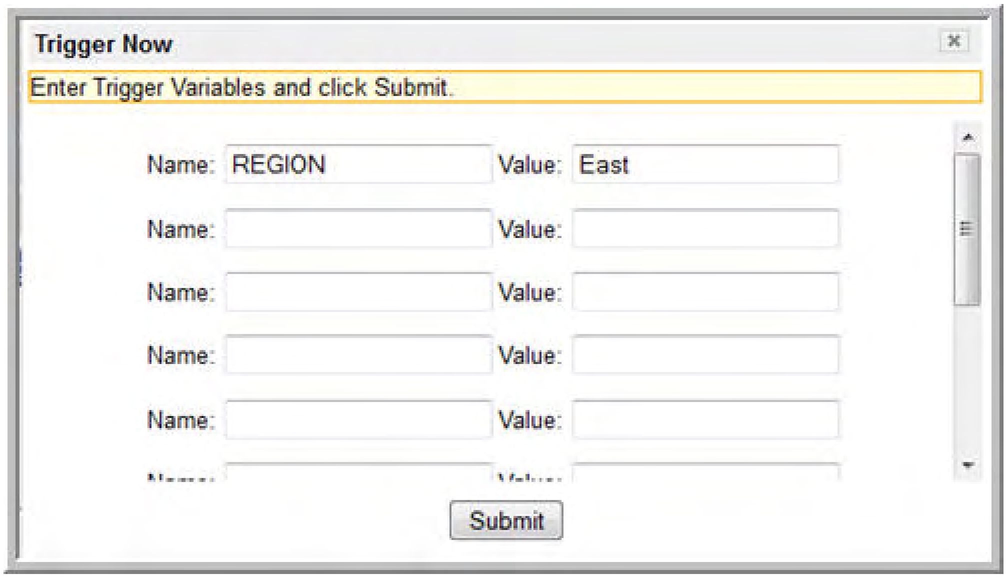
Two methods are available for entering variables: using the **Trigger With Variables** menu option and using the Variables tab and **Trigger Now** button. If you want to preserve information about the variables you are setting or overriding (the name and value), or if you want to specify default values, use the Variables tab. If you set up a Manual trigger with default values in the Variables tab, any values you set using the Trigger With Variables popup window override the values in the Variables tab. Each method is described below.

Using the Trigger with Variables Menu Option

1. Click the down-arrow to access the menu (see image below). A menu displays showing the Trigger With Variables option.



2. Select **Trigger With Variables**. A pop-up window displays that allows you to set or override the values of up to ten variables that will be used in the execution of the task(s) named in the Manual trigger. These can be existing or new variables. Any existing variables are automatically populated in the window.


Trigger Now

Enter Trigger Variables and click Submit.

Name: REGION Value: East

3. When you are finished entering the variables, click <b>Submit</b> in the pop-up window to launch the tasks named in the trigger. When you click <b>Submit</b>, the variable information in this window is passed into the task instance(s) where referenced and the contents of the window are deleted.

Using the Variables Tab

1. Click the Variables tab.
2. Use the **New** button to add the variables you want to set.
3. When you are finished, return to the main Trigger page and click **Trigger Now** to launch the tasks named in the trigger.

Manual Trigger Field Descriptions

Field Name	Description
Trigger Name	User-defined. Name used within Opswise to identify this trigger. It can contain a maximum of 40 alphanumeric characters. It is the user's responsibility to develop a workable naming scheme for triggers.

Task(s)	User-defined. The name of the task(s) being triggered when this trigger is satisfied. When selecting tasks from the definition screen, click on the lock icon to unlock the field and select tasks.
Description	User-defined. Copied from the Description field in the trigger.
Calendar	If the user selects **Special Restriction**, the calendar defines what the Holidays or Non Business days are. Type in a calendar name or click the magnifying glass to browse for an existing calendar or to add a new calendar. To display details about the calendar specified in this field, hover over the paper icon.
Member of Groups	User-defined. Allows you to select one or more Opswise groups that this record definition belongs to. Click on the lock icon to unlock the field and select groups.
Version	System-supplied. The version number of the current record, which is incremented by the system every time a user updates a record. Click on the Versions tab to view previous versions. For details, see Record Versioning.
Submit button	Submits the new record to the database.
Update button	Saves updates to the record.
Trigger Now button	Immediately triggers all the tasks specified in this trigger.
Delete button	Deletes the current record.
Variables tab	Displays all variables associated with this record.
Versions tab	Stores copies of all previous versions of the current record (see Record Versioning).

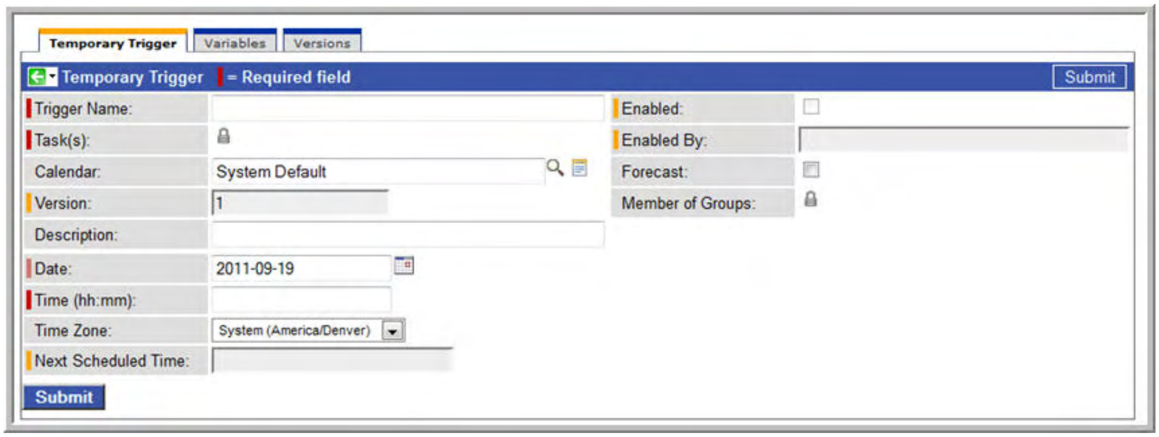
Temporary Trigger

Overview

The Temporary trigger allows you to set up a one-time trigger for a task, based on a single date and time. You will use this trigger if you want to set up a task to run once at some time in the future.

Creating a New Temporary Trigger

1. From the navigation pane, select **Automation Center > Triggers > Temporary Triggers**. The Temporary Triggers List screen displays.
2. Click **New**. The Temporary Trigger Definition screen displays.



Trigger Name:		Enabled:	□
Task(s):		Enabled By:	
Calendar:	System Default	Forecast:	□
Version:	1	Member of Groups:	
Description:			
Date:	2011-09-19		
Time (hh:mm):			
Time Zone:	System (America/Denver)		
Next Scheduled Time:			

3. Using the field descriptions provided below as a guide, complete the fields as needed.
4. Click the **Submit** button to save the record and return to the menu, or, right-click the title bar and select **Save** to save the record and remain on the current display.
5. If appropriate, repeat these steps for any additional triggers you want to add.
6. Enable the trigger(s).

Temporary Trigger Field Descriptions

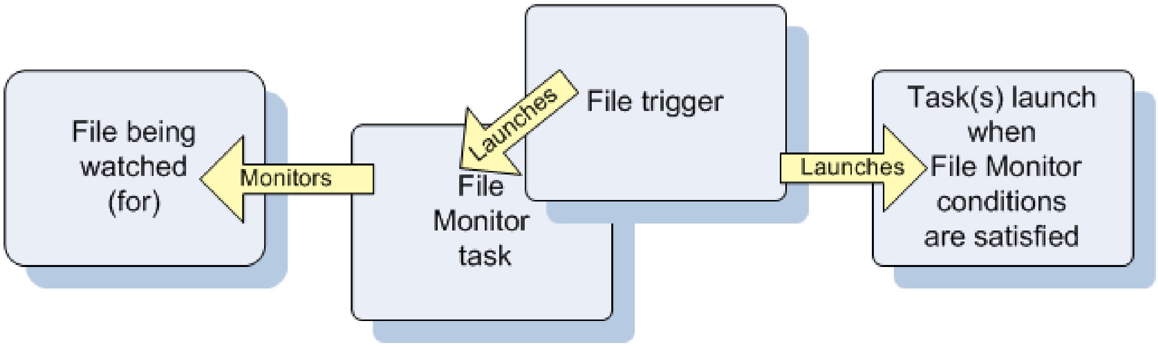
Field Name	Description
Trigger Name	User-defined. Name used within Opswise to identify this trigger. It can contain a maximum of 40 alphanumeric characters. It is the user's responsibility to develop a workable naming scheme for triggers.
Enabled	User-defined. Whether or not the Enabled field is checked. The user enables and disables the trigger by clicking the **Enable/Disable Trigger** buttons. Only enabled triggers are processed by Opswise.
Task(s)	User-defined. The name of the task(s) being triggered when this trigger is satisfied. When selecting tasks from the definition screen, click on the lock icon to unlock the field and select tasks.
Enabled By	System-supplied. Displays the ID of the user who most recently enabled this trigger.

Calendar	If the user selects **Special Restriction**, the calendar defines what the Holidays or Non Business days are. Type in a calendar name or click the magnifying glass to browse for an existing calendar or to add a new calendar. To display details about the calendar specified in this field, hover over the paper icon.
Forecast	Enabled or disabled by user. If enabled, the system calculates the date and time when this trigger will be satisfied for the next number days specified in the Forecast Period In Days property. The system writes the forecasting entries to the **Triggers > Forecasts** display. For details, see Displaying Trigger Forecast Information.
Version	System-supplied. The version number of the current record, which is incremented by the system every time a user updates a record. Click on the Versions tab to view previous versions. For details, see Record Versioning.
Member of Groups	User-defined. Allows you to select one or more Opswise groups that this record definition belongs to. Click on the lock icon to unlock the field and select groups.
Description	User-defined. Copied from the Description field in the trigger.
Date	The date you want the trigger to be satisfied.
Time	Required if Time Style = Time. Specifies the time of the trigger in hours, minutes, and seconds, using 24-hour time. For example, 01:45:00 means trigger the task at 1:45 a.m.; 13:45:00 means trigger the task at 1:45 p.m. Click here for details about how the system handles Daylight Savings Time.
Next Scheduled Time	Supplied by Opswise. For time-based triggers, the next date and time this trigger will be satisfied. See Displaying Trigger Forecast Information.
Submit button	Submits the new record to the database.
Update button	Saves updates to the record.
Enable Trigger button	Activates this trigger and writes your UserID to the Enabled By field.
Disable Trigger button	Deactivates this trigger.
Trigger Now button	Immediately triggers all the tasks specified in this trigger.
Delete button	Deletes the current record.
Variables tab	Displays all variables associated with this record.
Versions tab	Stores copies of all previous versions of the current record (see Record Versioning).

File Monitor Trigger

Overview

The File (Monitor) trigger allows you to trigger one or more tasks based on the creation, deletion, change, existence or non-existence of a file on a particular machine. The trigger works in conjunction with the File Monitor task, as illustrated in the image below. For a detailed description, see Launching a File Monitor Task Using a File (Monitor) Trigger.



Built-In Variables

The built-in variables outlined below can be used to pass data where appropriate:

- Task and Task Instance Variables
- File Monitor Variables.

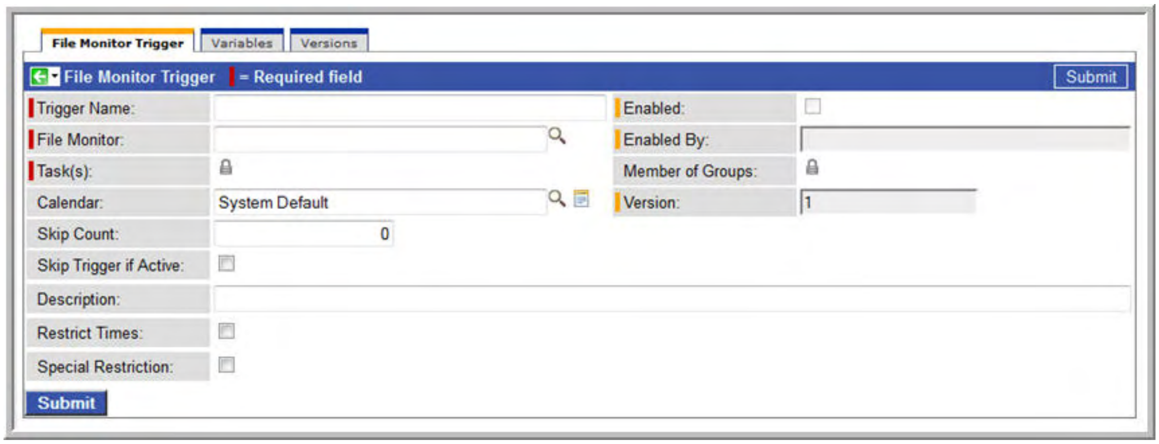
Prerequisites

Before you can use a File Monitor Trigger, you need the following:

- A Windows, Linux/Unix, or z/OS agent, which will execute the File Monitor task.
- A File Monitor task, which watches for the creation, deletion, change, existence, or non-existence of a file.

Creating a New File Monitor Trigger

1. From the navigation pane, select **Automation Center > Triggers > File Triggers**. The File Triggers List screen displays.
2. Click **New**. The File Monitor Trigger Definition screen displays.



Trigger Name:		Enabled:	□
File Monitor:		Enabled By:	
Task(s):		Member of Groups:	
Calendar:	System Default	Version:	1
Skip Count:	0		
Skip Trigger if Active:	□		
Description:			
Restrict Times:	□		
Special Restriction:	□		

3. Using the field descriptions provided below as a guide, complete the fields as needed.
4. Click the **Submit** button to save the record and return to the menu, or, right-click the title bar and select **Save** to save the record and remain on the current display.
5. If appropriate, repeat these steps for any additional triggers you want to add.
6. Enable the trigger(s).

File Monitor Trigger Field Descriptions

Field Name	Description
Trigger Name	User-defined. Name used within Opswise to identify this trigger. It can contain a maximum of 40 alphanumeric characters. It is the user's responsibility to develop a workable naming scheme for triggers.
Enabled	User-defined. Whether or not the Enabled field is checked. The user enables and disables the trigger by clicking the **Enable/Disable Trigger** buttons. Only enabled triggers are processed by Opswise.
File Monitor	User-defined. The File Monitor task being executed. Type in a task name or click the magnifying glass to browse for an existing task or to add a new task. To display details about the task specified in this field, hover over the paper icon. For details on file monitor types, see Monitor Type.
Enabled By	System-supplied. Displays the ID of the user who most recently enabled this trigger.
Task(s)	User-defined. The name of the task(s) being triggered when this trigger is satisfied. When selecting tasks from the definition screen, click on the lock icon to unlock the field and select tasks.
Member of Groups	User-defined. Allows you to select one or more Opswise groups that this record definition belongs to. Click on the lock icon to unlock the field and select groups.
Calendar	If the user selects **Special Restriction**, the calendar defines what the Holidays or Non Business days are. Type in a calendar name or click the magnifying glass to browse for an existing calendar or to add a new calendar. To display details about the calendar specified in this field, hover over the paper icon.
Version	System-supplied. The version number of the current record, which is incremented by the system every time a user updates a record. Click on the Versions tab to view previous versions. For details, see Record Versioning.
Running Monitor	System-supplied. Lists File Monitor tasks currently running that were launched by this trigger.

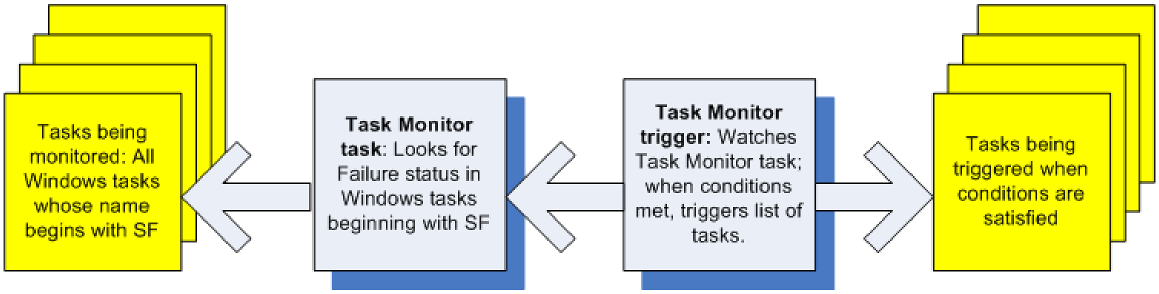
Skip Count	User-defined. Allows you to specify that Opswise skip the next *N* times this task is triggered.
Skip Trigger if Active	User-defined. Allows you to specify that Opswise skip the next run of the specified tasks if the previous run has not gone to a complete status (is still active).
Description	User-defined. Copied from the Description field in the trigger.
Restrict Times	If Time Style = Time Interval, you can specify a period during which the trigger is active. Enable the Restrict Times field and specify the start and end times in the Enabled Start and Enabled End fields (see below).
Time Zone	User-defined. Allows you to specify the timezone that will be applied to the times specified in the Restrict Times parameters. For example, if you specify an Enabled Start of 23:00, an Enabled End of 24:00, and a time zone of Canada/Central, the trigger is enabled at 11:00 p.m. and disabled at 12:00 a.m., Central Canada time.
Enabled Start	If Restrict Times is enabled, specify the start time of the period during which the trigger should be active. Use 24-hour time.
Enabled End	If Restrict Times is enabled, specify the end time of the period during which the trigger should be active. Use 24-hour time.
Special Restriction	Enable this field to specify additional parameters that tell the system how to handle exceptions such as when the trigger is satisfied on a holiday or non-business day. You can specify Simple or Complex Restrictions or both (see field descriptions below for details). For example, you may specify a Simple Restriction that disables the trigger if it is satisfied on a holiday. Or, you may chose Complex Restriction and specify more detailed parameters such do not trigger on the last business day of the month.
Simple Restriction	If enabled, allows you to specify an Action such as Do Not Trigger (see below) on a non-business day or holiday (see Situation field below). For example, do not trigger on a non business day.
Situation	If Simple Restriction is enabled, allows you to select the situation that causes the system to initiate the action specified in the Action field (see Action field below). Options: • On Non Business Day • On Holiday
Action	If Special Restriction is enabled, allows you to select an action to take on a non business day or holiday (see Situation field above). Options: • Do Not Trigger • Next Day (run on the next day) • Next Business Day (run on the next business day, as defined in the calendar) • Previous Day (run on the previous day) • Previous Business Day (run on the previous business day, as defined in the calendar)
Complex Restriction	If enabled, allows you to specify a set of parameters that determine one or more situations when this trigger should not be satisfied. Used in conjunction with the following fields: Restriction Mode, Restriction Adjective, Restriction Noun, Restriction Qualifier (see details below). For example, you may specify that you do not want to satisfy this trigger on the last business day of the year or the first day of each month.

Restriction Mode	If both Simple and Complex Restriction are enabled, specifies whether you want to use both restriction types (AND) or one or the other (OR). Options: - And - Or
Restriction Adjective	If Complex Restriction is enabled, the type of selection. Options: 1st, 2nd, 3rd, 4th, Last. Example: The **last** business day of the month.
Restriction Noun	If Complex Restriction is enabled, the day you want to select. Options: Sunday through Saturday, Day, or Business Day. Example: The last **business day** of the month.
Restriction Qualifier	If Complex Restriction is enabled, the period you are selecting from. Options: January through December, Month or Year. Example: The last business day of the **month**.
Submit button	Submits the new record to the database.
Update button	Saves updates to the record.
Enable Trigger button	Activates this trigger and writes your UserID to the Enabled By field.
Disable Trigger button	Deactivates this trigger.
Trigger Now button	Immediately triggers all the tasks specified in this trigger.
Delete button	Deletes the current record.
Variables tab	Displays all variables associated with this record.
Versions tab	Stores copies of all previous versions of the current record (see Record Versioning).

Task Monitor Trigger

Overview

The Task Monitor Trigger allows you to trigger one or more tasks based on the conditions specified in an associated Task Monitor task, as illustrated in the example below. For details, see Launching a Task Monitor Task Using a Task Monitor Trigger.


Tasks being monitored: All Windows tasks whose name begins with SF
Task Monitor task: Looks for Failure status in Windows tasks beginning with SF
Task Monitor trigger: Watches Task Monitor task; when conditions met, triggers list of tasks.
Tasks being triggered when conditions are satisfied

Built-In Variables

The built-in variables outlined below can be used to pass data where appropriate:

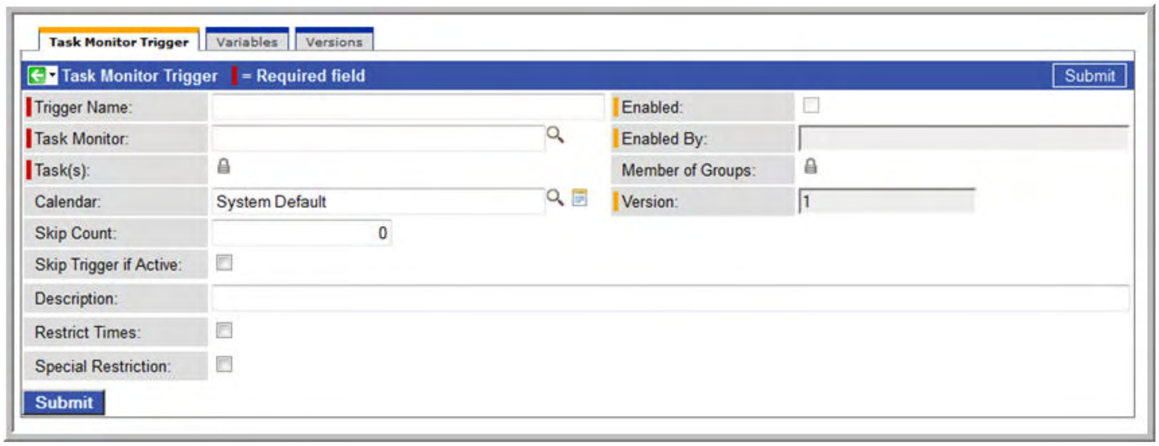
- Task and Task Instance Variables
- Task Monitor Variables.

Prerequisites

Before you can use a Task Monitor Trigger, you need a Task Monitor task, which defines the statuses being monitored for and the tasks being monitored.

Creating a New Task Monitor Trigger

1. From the navigation pane, select **Automation Center > Triggers > Task Monitor Triggers**. The Task Monitor Triggers List screen displays.
2. Click **New**. The Task Monitor Trigger Definition screen displays.



Trigger Name:		Enabled:	□
Task Monitor:		Enabled By:	
Task(s):		Member of Groups:	
Calendar:	System Default	Version:	1
Skip Count:	0		
Skip Trigger if Active:	□		
Description:			
Restrict Times:	□		
Special Restriction:	□		

3. Using the field descriptions provided below as a guide, complete the fields as needed.
4. Click the **Submit** button to save the record and return to the menu, or, right-click the title bar and select **Save** to save the record and remain on the current display.
5. If appropriate, repeat these steps for any additional triggers you want to add.
6. Enable the trigger(s). When you enable the trigger, its associated Task Monitor task launches.

Task Monitor Trigger Field Descriptions

Field Name	Description
Trigger Name	User-defined. Name used within Opswise to identify this trigger. It can contain a maximum of 40 alphanumeric characters. It is the user's responsibility to develop a workable naming scheme for triggers.
Enabled	User-defined. Whether or not the Enabled field is checked. The user enables and disables the trigger by clicking the **Enable/Disable Trigger** buttons. Only enabled triggers are processed by Opswise.
Task Monitor	User-defined. The Task Monitor task being executed. Type in a task name or click the magnifying glass to browse for an existing task or to add a new task. To display details about the task specified in this field, hover over the paper icon.
Enabled By	System-supplied. Displays the ID of the user who most recently enabled this trigger.
Task(s)	User-defined. The name of the task(s) being triggered when this trigger is satisfied. When selecting tasks from the definition screen, click on the lock icon to unlock the field and select tasks.
Member of Groups	User-defined. Allows you to select one or more Opswise groups that this record definition belongs to. Click on the lock icon to unlock the field and select groups.
Calendar	If the user selects **Special Restriction**, the calendar defines what the Holidays or Non Business days are. Type in a calendar name or click the magnifying glass to browse for an existing calendar or to add a new calendar. To display details about the calendar specified in this field, hover over the paper icon.
Version	System-supplied. The version number of the current record, which is incremented by the system every time a user updates a record. Click on the Versions tab to view previous versions. For details, see Record Versioning.
Skip Count	User-defined. Allows you to specify that Opswise skip the next *N* times this task is triggered.
Skip Trigger if Active	User-defined. Allows you to specify that Opswise skip the next run of the specified tasks if the previous run has not gone to a complete status (is still active).
Description	User-defined. Copied from the Description field in the trigger.
Restrict Times	Allows you to specify a period during which the trigger is active. Enable the Restrict Times field and specify the start and end times in the Enabled Start and Enabled End fields (see below).
Enabled Start	If Restrict Times is enabled, specify the start time of the period during which the trigger should be active. Use 24-hour time.

Enabled End	If Restrict Times is enabled, specify the end time of the period during which the trigger should be active. Use 24-hour time.
Timezone	Specifies the timezone for the times specified in the Restrict Time parameters.
Special Restriction	Enable this field to specify additional parameters that tell the system how to handle exceptions such as when the trigger is satisfied on a holiday or non-business day. You can specify Simple or Complex Restrictions or both (see field descriptions below for details). For example, you may specify a Simple Restriction that disables the trigger if it is satisfied on a holiday. Or, you may chose Complex Restriction and specify more detailed parameters such do not trigger on the last business day of the month.
Simple Restriction	If enabled, allows you to specify an Action such as Do Not Trigger (see below) on a non-business day or holiday (see Situation field below). For example, do not trigger on a non business day.
Situation	If Simple Restriction is enabled, allows you to select the situation that causes the system to initiate the action specified in the Action field (see Action field below). Options: • On Non Business Day • On Holiday
Action	If Special Restriction is enabled, allows you to select an action to take on a non business day or holiday (see Situation field above). Options: • Do Not Trigger • Next Day (run on the next day) • Next Business Day (run on the next business day, as defined in the calendar) • Previous Day (run on the previous day) • Previous Business Day (run on the previous business day, as defined in the calendar)
Complex Restriction	If enabled, allows you to specify a set of parameters that determine one or more situations when this trigger should not be satisfied. Used in conjunction with the following fields: Restriction Mode, Restriction Adjective, Restriction Noun, Restriction Qualifier (see details below). For example, you may specify that you do not want to satisfy this trigger on the last business day of the year or the first day of each month.
Restriction Mode	If both Simple and Complex Restriction are enabled, specifies whether you want to use both restriction types (AND) or one or the other (OR). Options: • And • Or
Restriction Adjective	If Complex Restriction is enabled, the type of selection. Options: 1st, 2nd, 3rd, 4th, Last. Example: The **last** business day of the month.
Restriction Noun	If Complex Restriction is enabled, the day you want to select. Options: Sunday through Saturday, Day, or Business Day. Example: The last **business day** of the month.
Restriction Qualifier	If Complex Restriction is enabled, the period you are selecting from. Options: January through December, Month or Year. Example: The last business day of the **month**.
Submit button	Submits the new record to the database.
Update button	Saves updates to the record.

Enable Trigger button	Activates this trigger and writes your UserID to the Enabled By field.
Disable Trigger button	Deactivates this trigger.
Trigger Now button	Immediately triggers all the tasks specified in this trigger.
Delete button	Deletes the current record.
Variables tab	Displays all variables associated with this record.
Versions tab	Stores copies of all previous versions of the current record (see Record Versioning).

Application Monitor Trigger

Overview

The Application Monitor Trigger allows you to trigger one or more tasks based on the status of:

- A specific application resource
- One or more application resources, based on selection criteria you supply.

You can launch any number of tasks when the conditions in the trigger are satisfied.

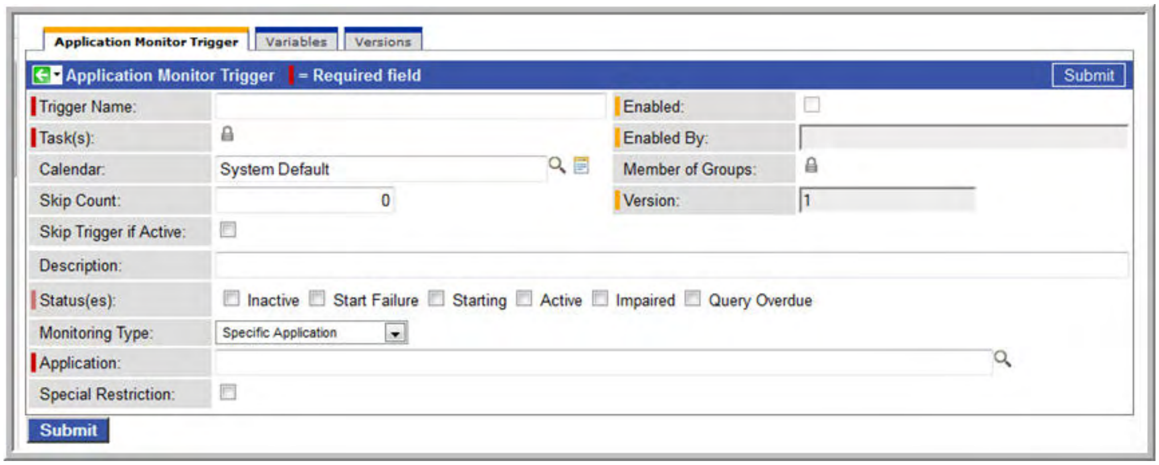
For example, if you specify Application Monitor Condition = ALL, and select all Application types, the trigger monitors all Application resource records you have defined. Any time any one of them goes to any of the Statuses you specified in the Status(es) fields, the trigger launches the task(s) specified in the Task(s) field. For example, you might use this trigger to send an email notification to technical support if any of the monitored applications goes into a failure status.

Built-In Variables

Several built-in variables are provided to pass information about the Application being monitored into the task or tasks being launched by the trigger. You can pass the information into the launched tasks by including the variables in a text field in the task definition. See Application Monitor Variables for details.

Creating a New Application Monitor Trigger

1. From the navigation pane, select **Automation Center > Triggers > Application Monitor Triggers**. The Application Monitor Triggers List screen displays.
2. Click **New**. The Application Monitor Trigger Definition screen displays.



Trigger Name:		Enabled:	□
Task(s):		Enabled By:	
Calendar:	System Default	Member of Groups:	
Skip Count:	0	Version:	1
Skip Trigger if Active:	□		
Description:			
Status(es):	□ Inactive □ Start Failure □ Starting □ Active □ Impaired □ Query Overdue		
Monitoring Type:	Specific Application		
Application:			
Special Restriction:	□		

3. Using the field descriptions provided below as a guide, complete the fields as needed.
4. Click the **Submit** button to save the record and return to the menu, or, right-click the title bar and select **Save** to save the record and remain on the current display.
5. If appropriate, repeat these steps for any additional triggers you want to add.
6. Enable the trigger(s). Once the trigger is enabled, it monitors the Application resource(s) specified in the Monitoring Type fields.

Application Monitor Trigger Field Descriptions

Field Name	Description
Trigger Name	User-defined. Name used within Opswise to identify this trigger. It can contain a maximum of 40 alphanumeric characters. It is the user's responsibility to develop a workable naming scheme for triggers.
Enabled	User-defined. Whether or not the Enabled field is checked. The user enables and disables the trigger by clicking the **Enable/Disable Trigger** buttons. Only enabled triggers are processed by Opswise.
Task(s)	User-defined. The name of the task(s) being triggered when this trigger is satisfied. When selecting tasks from the definition screen, click on the lock icon to unlock the field and select tasks.
Enabled By	System-supplied. Displays the ID of the user who most recently enabled this trigger.
Member of Groups	User-defined. Allows you to select one or more Opswise groups that this record definition belongs to. Click on the lock icon to unlock the field and select groups.
Calendar	If the user selects **Special Restriction**, the calendar defines what the Holidays or Non Business days are. Type in a calendar name or click the magnifying glass to browse for an existing calendar or to add a new calendar. To display details about the calendar specified in this field, hover over the paper icon.
Skip Count	User-defined. Allows you to specify that Opswise skip the next *N* times this task is triggered.
Version	System-supplied. The version number of the current record, which is incremented by the system every time a user updates a record. Click on the Versions tab to view previous versions. For details, see Record Versioning.
Skip Trigger if Active	User-defined. Allows you to specify that Opswise skip the next run of the specified tasks if the previous run has not gone to a complete status (is still active).
Description	User-defined. Copied from the Description field in the trigger.
Status(es)	System-supplied. The application status being monitored for. One or more of the following: • **Inactive** - The initial state of the Application. The Application is stopped and unmonitored. • **Start Failure** - The agent experienced a failure while attempting to execute the Start command. • **Starting** - The Start command was executed and Opswise is waiting for Query command response. • **Active** - The Query command response is reporting that the Application is Active. • **Impaired** - The Query command response is reporting that the Application is experiencing a problem and is possibly down. • **Query Overdue** - The agent is late sending Opswise an updated Query command response.
Monitoring Type	Indicates whether you are monitoring one specific Application resource or want to provide selection parameters to monitor multiple Application resources. See Applications for information about setting up Application resources. Options: • **Specific Application** - Use the Application field below to browse for and select the Application resource you want to monitor. • **General Applications** - Use the Application Monitor Condition and Application Type(s) fields to provide parameters for selecting which Application resources you want to monitor.
Application	If Monitoring Type is "Specific Application": name of a specific application resource to be monitored.

Application Monitor Condition	If Monitoring Type is "General Application(s)," this allows you to specify selection parameters, as follows: • **ALL** - Monitor all Application resources. • **Starts With** - Monitor all Application resources whose name starts with the string you provide in the Condition Value field. • **Contains** - Monitor all Application resources whose name contains the string you provide in the Condition Value field. • **Ends With** - Monitor all Application resources whose name ends with the string you provide in the Condition Value field.
Condition Value	If Application Monitor Condition = Starts With, Contains, or Ends With, use this field to specify the search string.
Application Type(s)	All Application resource definitions include a field called Application Type. You can use this information to select Application resources for monitoring based on what type they are. For example, you may want to monitor all Windows services. The options in the Application resource definitions match the options on this screen: • Windows Service • Linux/Unix Daemon • z/OS Started Task
Special Restriction	Enable this field to specify additional parameters that tell the system how to handle exceptions such as when the trigger is satisfied on a holiday or non-business day. You can specify Simple or Complex Restrictions or both (see field descriptions below for details). For example, you may specify a Simple Restriction that disables the trigger if it is satisfied on a holiday. Or, you may chose Complex Restriction and specify more detailed parameters such do not trigger on the last business day of the month.
Simple Restriction	If enabled, allows you to specify an Action such as Do Not Trigger (see below) on a non-business day or holiday (see Situation field below). For example, do not trigger on a non business day.
Situation	If Simple Restriction is enabled, allows you to select the situation that causes the system to initiate the action specified in the Action field (see Action field below). Options: • On Non Business Day • On Holiday
Action	If Special Restriction is enabled, allows you to select an action to take on a non business day or holiday (see Situation field above). Options: • Do Not Trigger • Next Day (run on the next day) • Next Business Day (run on the next business day, as defined in the calendar) • Previous Day (run on the previous day) • Previous Business Day (run on the previous business day, as defined in the calendar)
Complex Restriction	If enabled, allows you to specify a set of parameters that determine one or more situations when this trigger should not be satisfied. Used in conjunction with the following fields: Restriction Mode, Restriction Adjective, Restriction Noun, Restriction Qualifier (see details below). For example, you may specify that you do not want to satisfy this trigger on the last business day of the year or the first day of each month.
Restriction Mode	If both Simple and Complex Restriction are enabled, specifies whether you want to use both restriction types (AND) or one or the other (OR). Options: • And • Or
Restriction Adjective	If Complex Restriction is enabled, the type of selection. Options: 1st, 2nd, 3rd, 4th, Last. Example: The **last** business day of the month.

Restriction Noun	If Complex Restriction is enabled, the day you want to select. Options: Sunday through Saturday, Day, or Business Day. Example: The last **business day** of the month.
Restriction Qualifier	If Complex Restriction is enabled, the period you are selecting from. Options: January through December, Month or Year. Example: The last business day of the **month**.
Submit button	Submits the new record to the database.
Update button	Saves updates to the record.
Enable Trigger button	Activates this trigger and writes your UserID to the Enabled By field.
Disable Trigger button	Deactivates this trigger.
Trigger Now button	Immediately triggers all the tasks specified in this trigger.
Delete button	Deletes the current record.
Variables tab	Displays all variables associated with this record.
Versions Tab	Stores copies of all previous versions of the current record (see Record Versioning).

Enabling and Disabling Triggers

Introduction

Opswise only processes triggers that are flagged as Enabled. (Enable triggers are Active triggers.) For tracking and compliance purposes, you must manually enable and disable triggers using the **Enable Trigger** and **Disable Trigger** buttons or menu options, or the command to enable and disable triggers. This process saves an audit record detailing the event. The trigger record also displays the ID of the user who enabled it. When you define and submit a new trigger, you must then enable it in order for the system to begin processing it.

⚠ This does not apply to Manual triggers.

Enabling/Disabling One or More Triggers

1. Display the Triggers list or Active Triggers list.
2. For each trigger you want to enable or disable, click the box in the leftmost column.
3. From the **Actions on selected rows...** menu at the bottom of the list, select **Enable Triggers** or **Disable Triggers** as appropriate. The Enabled flag on the trigger is modified as appropriate.

Enabling/Disabling a Single Trigger

1. From the triggers list, right-click the trigger you want to enable or disable. A drop-down menu displays.
2. Select **Enable Trigger** or **Disable Trigger** as appropriate. The Enabled flag on the trigger is modified.

Or,

1. Display the trigger you want to enable or disable.
2. Click the **Enable Trigger** or **Disable Trigger** button as appropriate. The Enabled flag on the trigger is modified.

Enabling/Disabling One or More Triggers from the Command Line

See the Command Line Interface Reference for instructions.

Copying Triggers

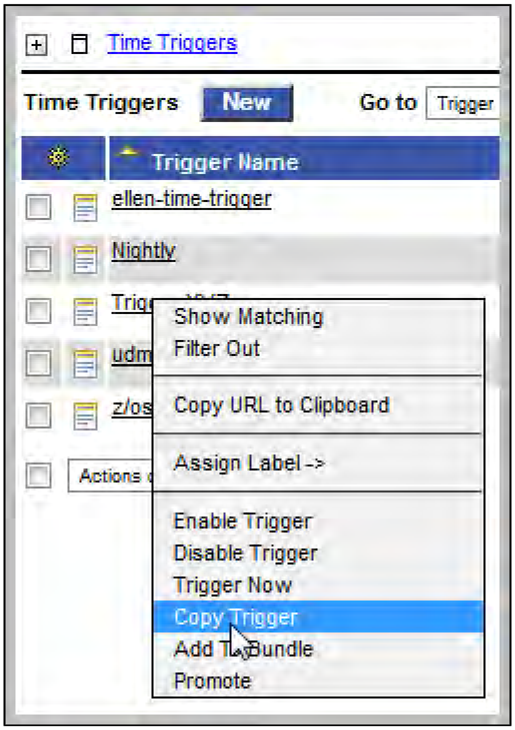
Overview

You can make copies of all Opswise records, including triggers, using the standard methods for Saving, Updating, Deleting, and Copying Records . However, the standard method does not make copies of other records that may be attached to the trigger, such as Notifications, Actions, Variables, and so on.

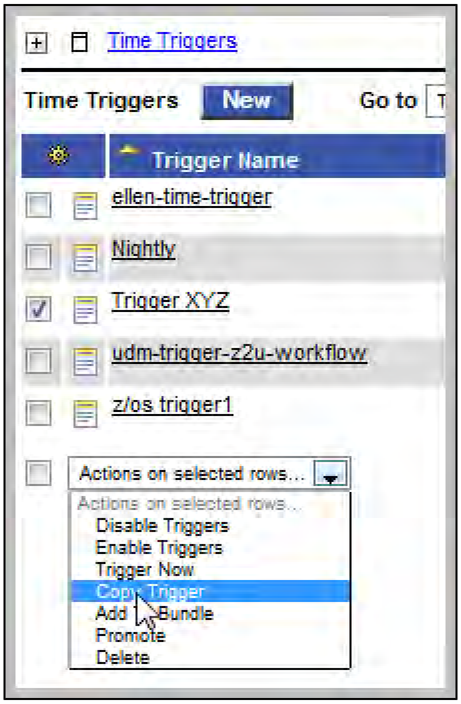
The Copy Trigger option allows you to make a complete copy of a trigger, including all of its associated records, such as variables and notes. It does not copy referenced records, such as virtual resources, but retains the relationship to these records for the copied trigger.

Copying One or More Triggers on the Trigger List

1. From the navigation pane, select a trigger type from **Automation Center > Triggers**. The Triggers List for the selected trigger type dis0plays.
2. Locate the trigger(s) you want to copy (see Searching for Records).
3. Cop y the trigger(s) using either of two methods:
   a. To copy a single trigger, hover over the trigger name and right-click. From the pop-up menu, select **Copy Trigger**.



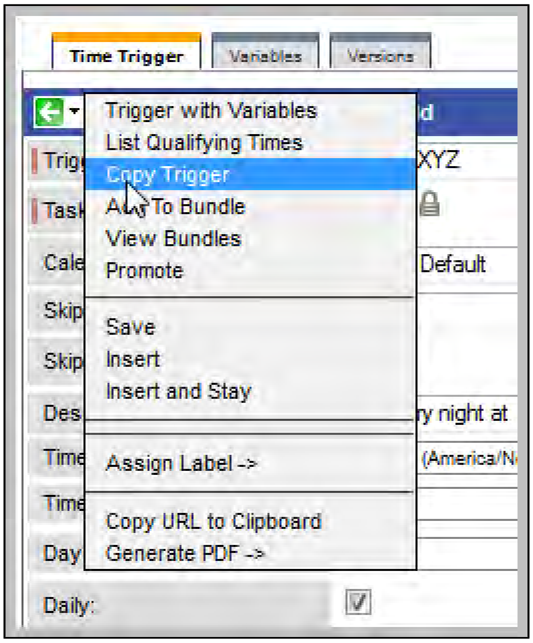
   b. To copy one or more triggers, click the box to the left of each trigger name. From the **Action on selected rows...** drop-down list at the bottom of the page, select **Copy Trigger**.



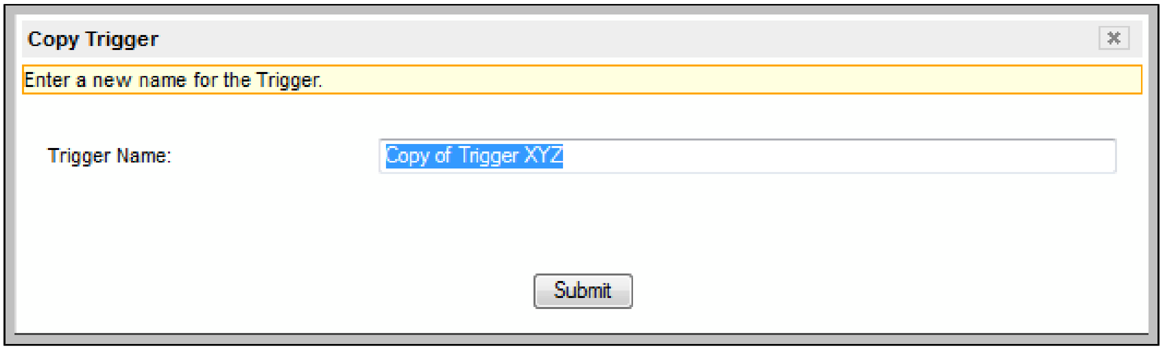
4. The system copies the trigger(s), automatically creating the new name by prepending the original name with "Copy of" (for example, "Copy of Trigger XYZ"), and adds it to the list. If the new name already exists, the system appends a counter to the name, such as "Copy of Trigger XYZ 1", "Copy of Trigger XYZ 2", and so on, until it finds a name that is available.
5. To modify the name or any other information in the trigger, open the new trigger, make your changes, and click **Update**.

Copying a Trigger on the Trigger Definition Form

1. Open the trigger you want to copy.
2. Hover over the down arrow above **Trigger Name:**. A pop-up menu displays:



3. Click **Copy Trigger**. Another window appears, prompting for a name for the new trigger. The default is the original trigger name, prepended with "Copy of," as shown in the following example:



Copy Trigger

Enter a new name for the Trigger.

Trigger Name: Copy of Trigger XYZ

4. Enter a new name for the trigger and click **Submit**. The system copies the trigger and all its attachments and saves it under the new name. (If the new name already exists, the copy will fail.)

Triggering with Variables

Overview

Opswise provides two methods for manually launching all of the tasks associated with a trigger while supplying values for variables used by the task(s):

- Use the **Trigger with Variables pop-up** method if you do not want the values that you enter for variables to persist. The values will apply only for the time the task(s) is running.
- Use the **Variables tab** method if you want to preserve the information (name and value) about the variables you are setting.

Both methods are available for all trigger types. You can use either method to manually launch task(s) when you cannot use the **Launch Task** button (on the task screen) because you want to override one or more variables.

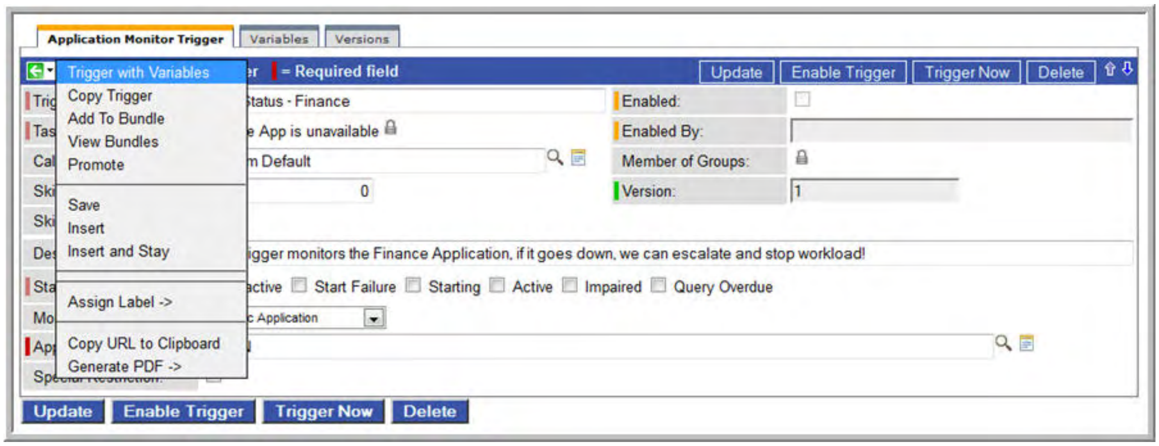
The values that you enter when using either method override the all other values, set elsewhere, for those variables.

Variables set with the **Trigger with Variables pop-up** method override any specified with the **Variables tab** method, but only for that run of the task(s).

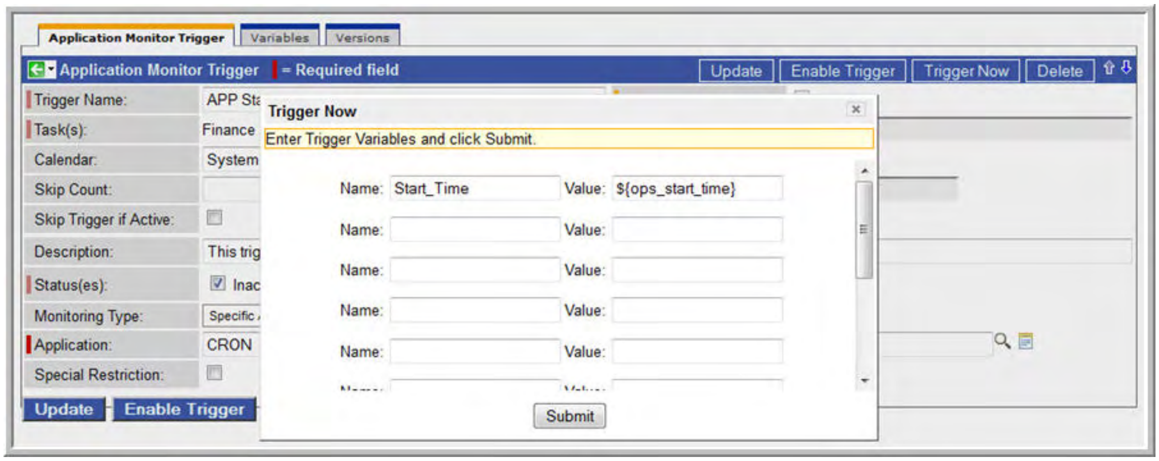
The audit message created when you use either method is the same.

Using the Trigger with Variables Method

1. From the navigation pane, select **Automation Center > Triggers** and a trigger type.
2. Select the trigger whose tasks you want to launch.
3. Hover your cursor over the the down-arrow or right-click on the title bar. The pop-up Trigger menu displays.



4. Click **Trigger with Variables**. The Trigger Variables pop-up displays.



Trigger Name: APP St
Task(s): Finance
Calendar: System
Skip Count:
Skip Trigger if Active: ☐
Description: This trig
Status(es): ☑ Inac
Monitoring Type: Specific
Application: CRON
Special Restriction: ☐

Trigger Now

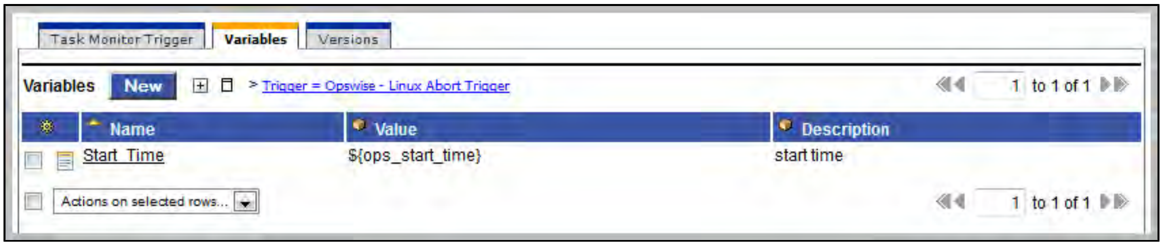
Enter Trigger Variables and click Submit.

Name: Start_Time Value: ${ops_start_time}

5. Enter the name and value for up to 10 variables that will be used when executing the task(s) named in the trigger. Any variables attached to the task are automatically displayed.
6. Click **Submit** to launch the tasks named in the trigger. The variable information in this window is used where referenced in the tasks. After launching the tasks, the system deletes the contents of the window.

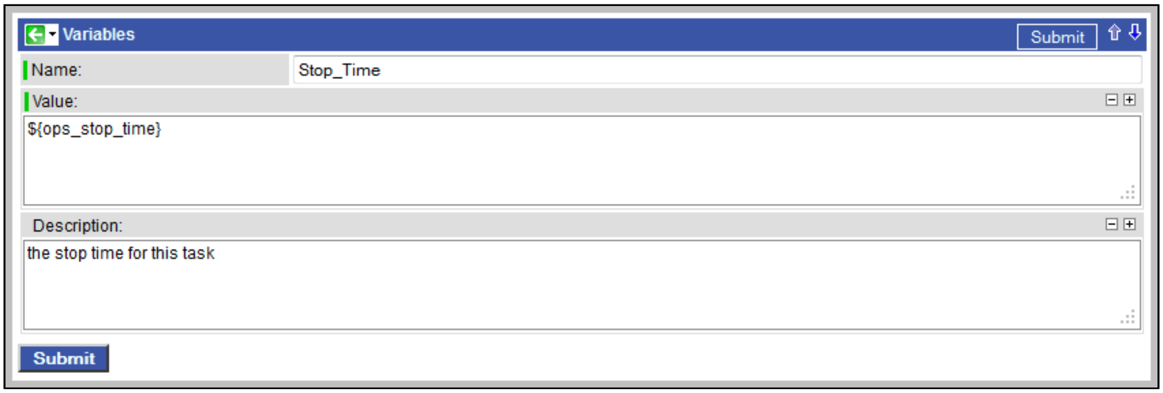
Using the Variables Tab Method

1. From the navigation pane, select **Automation Center > Triggers** and a trigger type.
2. Select the trigger whose tasks you want to launch.
3. Click the **Variables** tab. The Trigger Variables List screen displays.



Name	Value	Description
Start_Time	${ops_start_time}	start time

4. To add a variable:
   a. Click **New**. The Trigger Variables Definition screen displays.



Name: Stop_Time

Value: ${ops_stop_time}

Description: the stop time for this task

   b. Enter a name, value, and description for the variable and click **Submit**.
5. To update a variable:
   a. Click the variable name on the Trigger Variables List screen.
   b. Change the name, values, and/or description of the variable and click **Update**.
6. Click the **<trigger type> Trigger** tab.
7. Click **Trigger Now** to launch the tasks named in the trigger.

Displaying Trigger Forecast Information

Introduction

Several methods are available for displaying forecasting reports on time-based triggers and the tasks they launch. These are:

- The **Forecast calendar** displays a calendar showing tasks scheduled to run based on a Time, Cron or Temporary trigger. Where data is available, the estimated end time for each task is also calculated and displayed.
- The **Next Scheduled Time** field on the trigger screen displays the next time a given trigger will launch its task(s).
- **List Qualifying Times** is a button on the Time and Cron triggers screens. Clicking the button opens a new browser tab and displays a list of the next 20 qualifying dates and times.

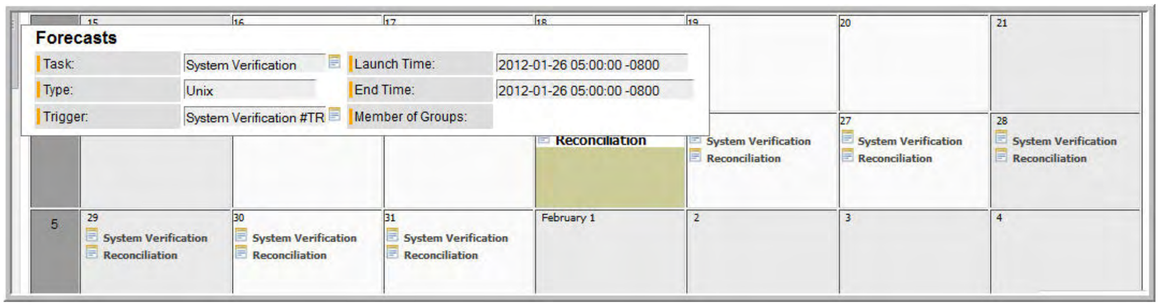
Each of these is described below.

Forecast Calendar

For enabled Time, Temporary, and Cron triggers where forecasting has been specified, Opswise writes an entry to the Forecasts calendar for each task instance that is scheduled to run within the next *N* days. The number of days included in the forecast is configurable.

To view the Forecast calendar, select **Automation Center > Triggers > Forecasts** from the Navigation Pane. Opswise displays a calendar showing each task instance that is forecast to be run in the next *N* days, where *N* is the forecast period you specified. For example, if a task will run ten times in the specified period, an entry displays for each of the ten task instances.

Shown below is a sample Forecast calendar, with two tasks running daily on a Time trigger. You can display the details of an entry, as shown, by hovering over the entry.



Forecasts

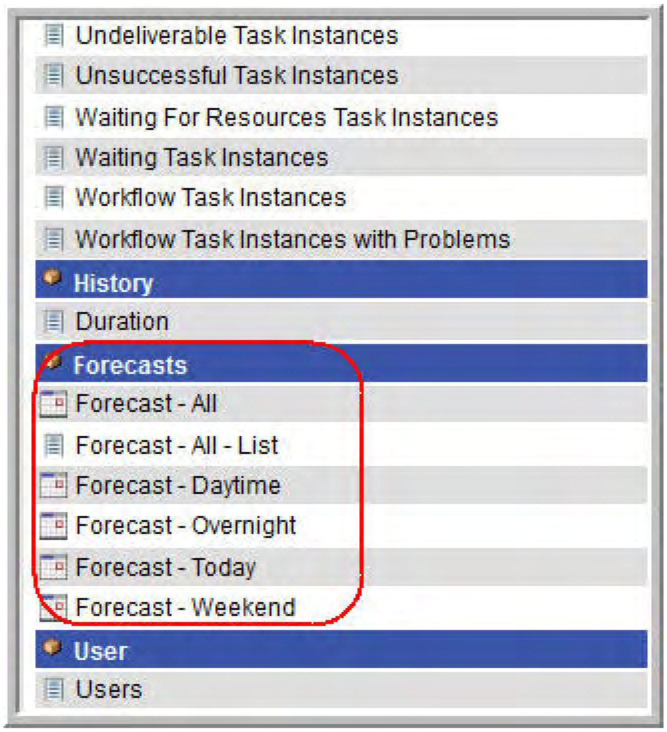
Task:	System Verification	Launch Time:	2012-01-26 05:00:00 -0800
Type:	Unix	End Time:	2012-01-26 05:00:00 -0800
Trigger:	System Verification #TR	Member of Groups:	

	15	16	17	18	19	20	21
				Reconciliation	System Verification Reconciliation	27 System Verification Reconciliation	28 System Verification Reconciliation
5	29 System Verification Reconciliation	30 System Verification Reconciliation	31 System Verification Reconciliation	February 1	2	3	4

As the tasks are run, the end time of each forecast entry is calculated by the system. The calculation is the average runtime, based on task instances that have already run. This information is updated each time you display the forecast.

Use the following points as a checklist when setting up forecasting:

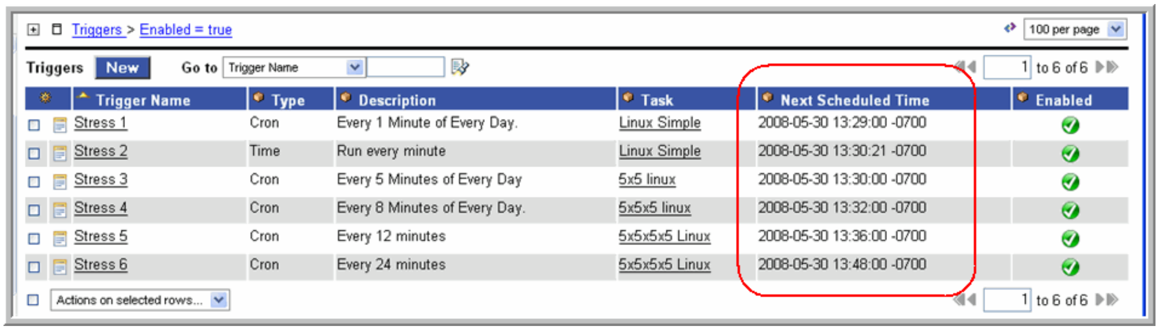
- Forecasting is supported in the following trigger types: Time, Temporary, and Cron.
- In the trigger, make sure the Forecast field is enabled.
- Make sure the trigger is enabled.
- Disabling the trigger removes all related entries from the Forecast calendar.
- To specify the number of days you want covered in your Forecasts:
  1. Select **Configuration > Properties**. You need administrative privileges to access this function.
  2. Open the property called **Forecast Period In Days**.
  3. The default number of days is 31. This means that the forecast calendar will display task instances that are scheduled within the next 31 days. Enter the number of days you want displayed in the Forecasts calendar and click **Update** to save.
- The Forecast calendar is generated by a report called Forecast - All. You can also navigate to the Reports menu and run several other pre-defined forecasting reports, shown below in the Reports list:



Undeliverable Task Instances
Unsuccessful Task Instances
Waiting For Resources Task Instances
Waiting Task Instances
Workflow Task Instances
Workflow Task Instances with Problems
History
Duration
Forecasts
Forecast - All
Forecast - All - List
Forecast - Daytime
Forecast - Overnight
Forecast - Today
Forecast - Weekend
User
Users

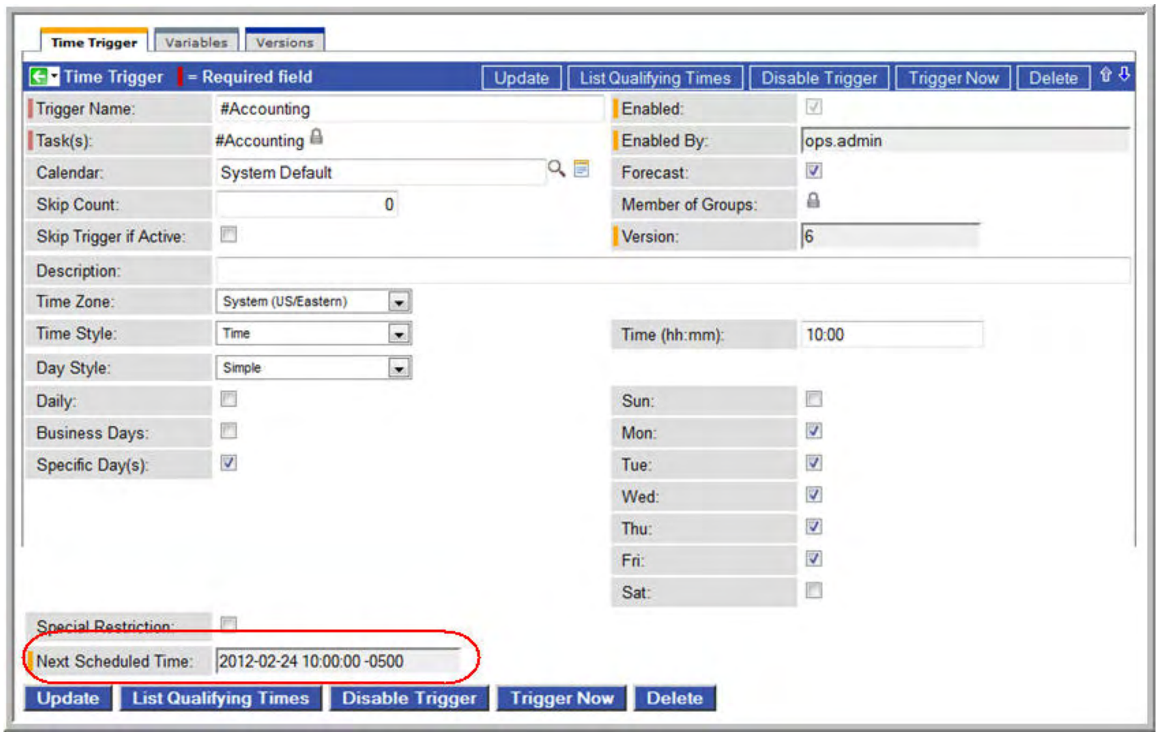
Next Scheduled Time

For enabled Time, Temporary and Cron triggers, Opswise calculates the next scheduled time and displays it on the Trigger list, as shown in the example below.



	Trigger Name	Type	Description	Task	Next Scheduled Time	Enabled
□	Stress 1	Cron	Every 1 Minute of Every Day.	Linux Simple	2008-05-30 13:29:00 -0700	✓
□	Stress 2	Time	Run every minute	Linux Simple	2008-05-30 13:30:21 -0700	✓
□	Stress 3	Cron	Every 5 Minutes of Every Day	5x5 linux	2008-05-30 13:30:00 -0700	✓
□	Stress 4	Cron	Every 8 Minutes of Every Day.	5x5x5 linux	2008-05-30 13:32:00 -0700	✓
□	Stress 5	Cron	Every 12 minutes	5x5x5x5 Linux	2008-05-30 13:36:00 -0700	✓
□	Stress 6	Cron	Every 24 minutes	5x5x5x5 Linux	2008-05-30 13:48:00 -0700	✓

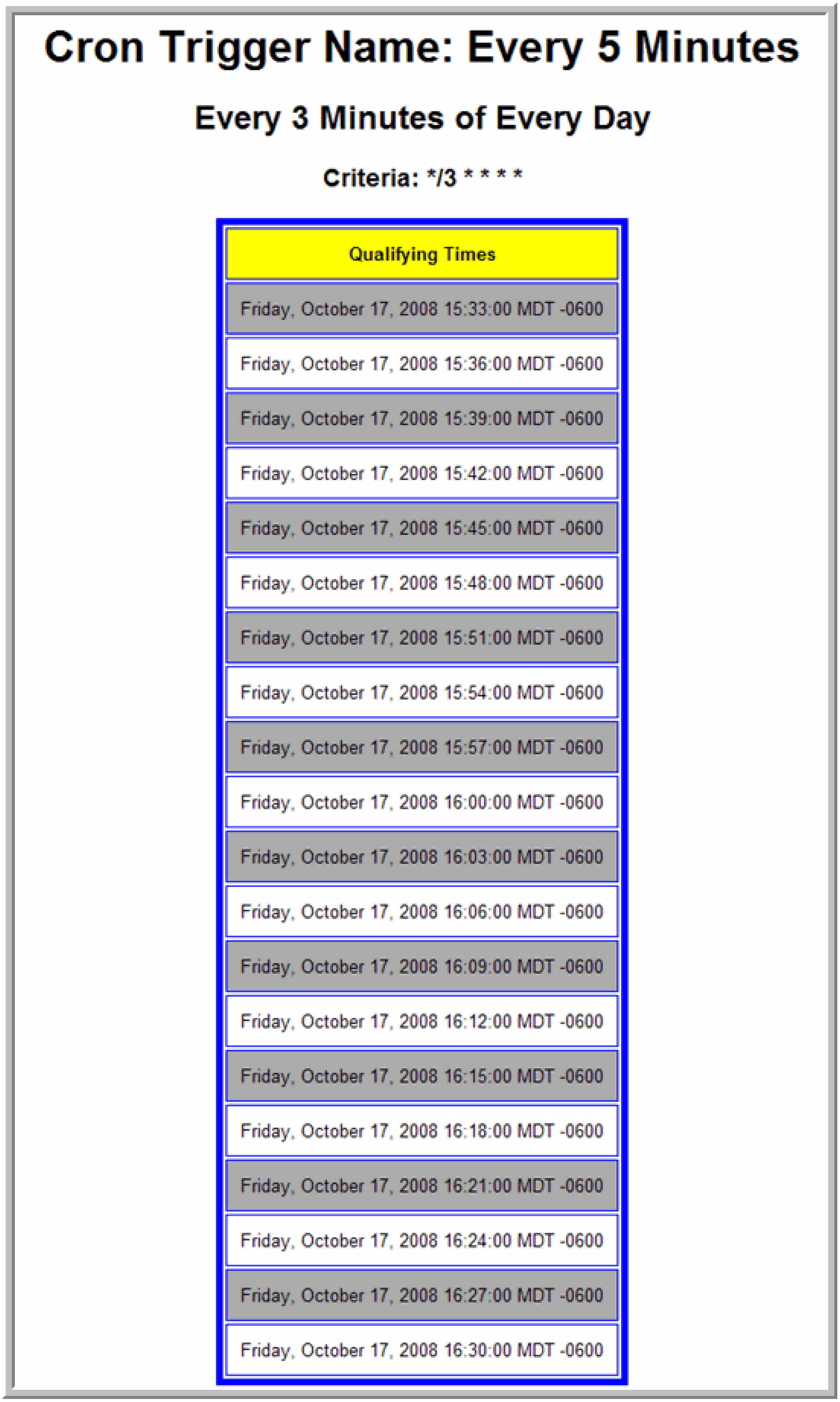
The next scheduled time also displays within the trigger record, as shown:



Trigger Name:	#Accounting	Enabled:	☑
Task(s):	#Accounting	Enabled By:	ops.admin
Calendar:	System Default	Forecast:	☑
Skip Count:	0	Member of Groups:	
Skip Trigger if Active:	□	Version:	6
Description:			
Time Zone:	System (US/Eastern)		
Time Style:	Time	Time (hh:mm):	10:00
Day Style:	Simple		
Daily:	□	Sun:	□
Business Days:	□	Mon:	☑
Specific Day(s):	☑	Tue:	☑
		Wed:	☑
		Thu:	☑
		Fri:	☑
		Sat:	□
Special Restriction:	□		
Next Scheduled Time:	2012-02-24 10:00:00 -0500		

List Qualifying Times

For Time, Temporary and Cron triggers, you can display a list of the next 20 dates and times when the trigger will be satisfied by clicking the **List Qualifying Times** button from the trigger record. Opswise displays the list in a new browser tab. Note that this display differs from the Forecast list, which shows scheduled task instances as opposed to qualifying times. The following example shows the next 20 dates and times when a Cron trigger will be satisfied.



Cron Trigger Name: Every 5 Minutes

Every 3 Minutes of Every Day

Criteria: */3 * * * *

Qualifying Times
Friday, October 17, 2008 15:33:00 MDT -0600
Friday, October 17, 2008 15:36:00 MDT -0600
Friday, October 17, 2008 15:39:00 MDT -0600
Friday, October 17, 2008 15:42:00 MDT -0600
Friday, October 17, 2008 15:45:00 MDT -0600
Friday, October 17, 2008 15:48:00 MDT -0600
Friday, October 17, 2008 15:51:00 MDT -0600
Friday, October 17, 2008 15:54:00 MDT -0600
Friday, October 17, 2008 15:57:00 MDT -0600
Friday, October 17, 2008 16:00:00 MDT -0600
Friday, October 17, 2008 16:03:00 MDT -0600
Friday, October 17, 2008 16:06:00 MDT -0600
Friday, October 17, 2008 16:09:00 MDT -0600
Friday, October 17, 2008 16:12:00 MDT -0600
Friday, October 17, 2008 16:15:00 MDT -0600
Friday, October 17, 2008 16:18:00 MDT -0600
Friday, October 17, 2008 16:21:00 MDT -0600
Friday, October 17, 2008 16:24:00 MDT -0600
Friday, October 17, 2008 16:27:00 MDT -0600
Friday, October 17, 2008 16:30:00 MDT -0600

Calendars

Calendars define business days, holidays, and other special days. They are used in conjunction with triggers to define when tasks are run. Specifically, calendars are used by the system to determine run dates as follows:

- If the user selects Business Days in a trigger, the calendar selected in the trigger defines the business days.
- If the user selects Day Style = Complex in the trigger, all the Custom Days attached to the calendar are written to the Date Noun drop-down menu.
- If the user selects Special Restriction in the trigger, the calendar defines what the Holidays or Non Business days are.

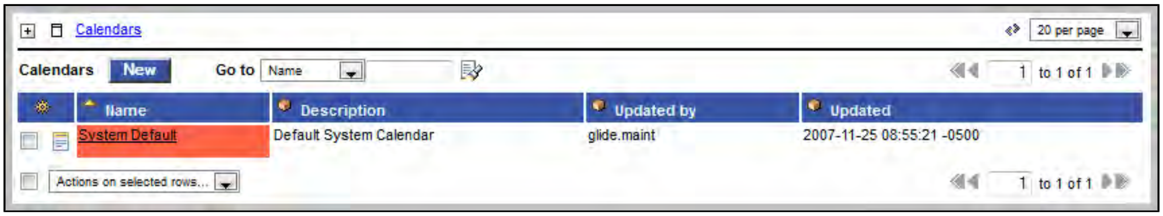
The process for setting up your calendars is:

1. Define the custom days you will need.
2. Create one or more calendars.
3. Assign the custom days to the calendar(s).

Creating Calendars

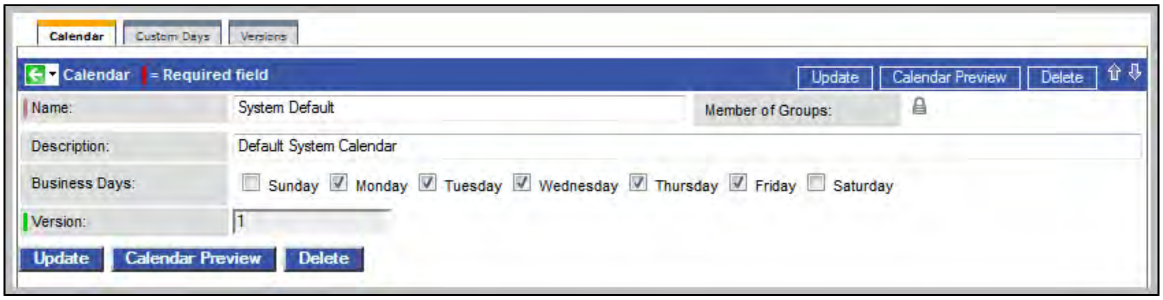
Creating a New Calendar

1. From the Navigation Pane, select **Automation Center > Calendars**. The Calendar list screen displays (shown below is the default list):



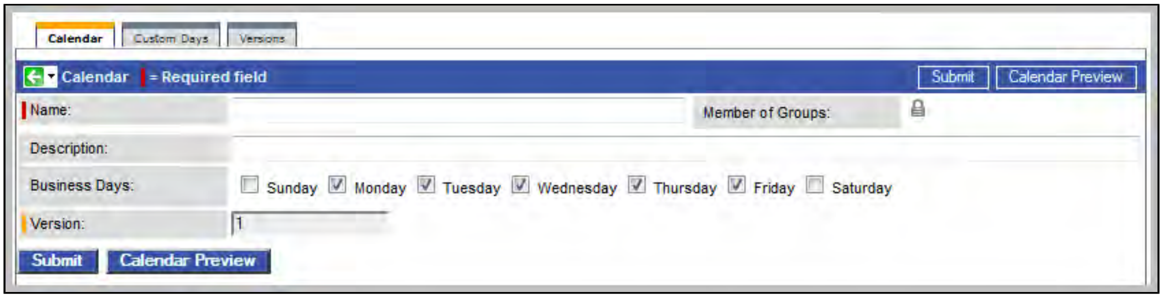
Name	Description	Updated by	Updated
System Default	Default System Calendar	glide.maint	2007-11-25 08:55:21 -0500

SystemDefault, the default system calendar, defines the work week.



Name:	System Default
Member of Groups:	
Description:	Default System Calendar
Business Days:	□ Sunday ☑ Monday ☑ Tuesday ☑ Wednesday ☑ Thursday ☑ Friday □ Saturday
Version:	1

2. Click **New**. A blank Calendar screen displays.



Name:	
Member of Groups:	
Description:	
Business Days:	□ Sunday ☑ Monday ☑ Tuesday ☑ Wednesday ☑ Thursday ☑ Friday □ Saturday
Version:	1

3. Give the calendar a name and description, and select the business days for this calendar. The default selection is Monday through Friday.
4. Right-click on the title bar and select **Save** to save the record and remain on the current display.
5. Follow the appropriate instructions below to assign existing custom days to this calendar or add new custom days to the calendar.
6. If appropriate, repeat these steps for any additional calendars you want to add.

Calendar Field Descriptions

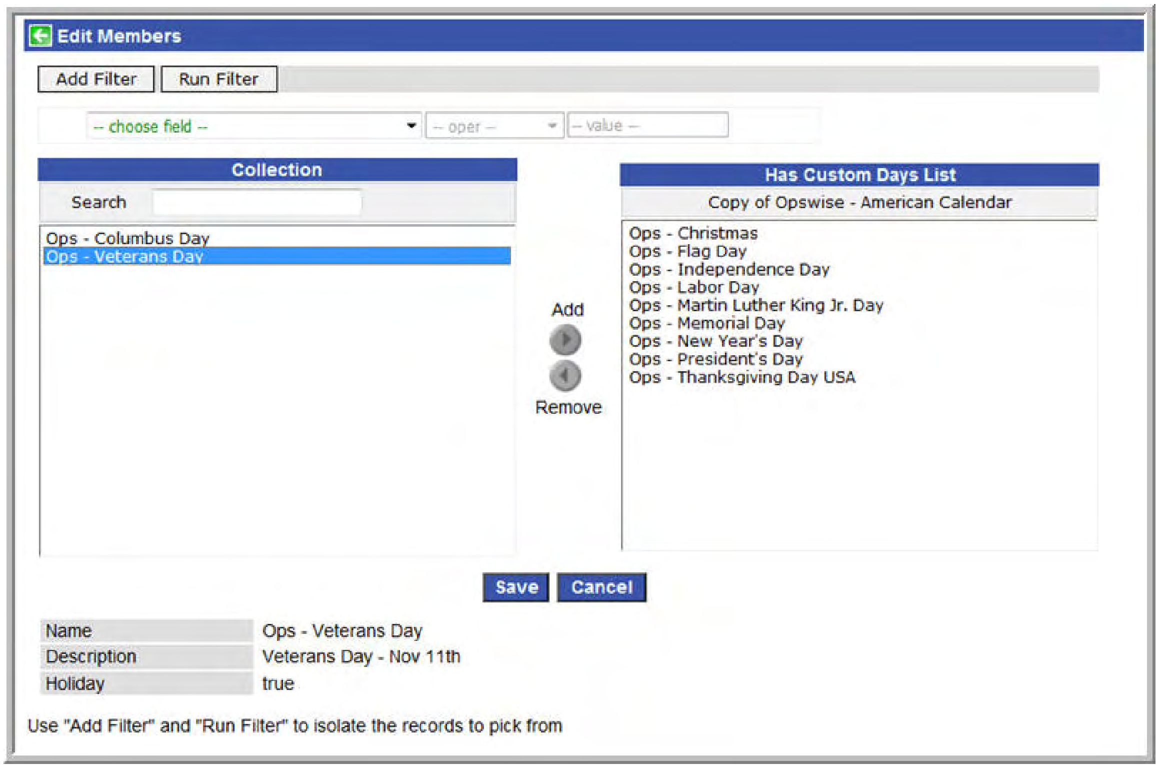
The following table provides field descriptions for the calendar form.

Field Name	Description

Name	User-defined. Name used within Opswise to identify this calendar. Up to 40 alphanumerics. It is the user's responsibility to develop a workable naming scheme for calendars.
Member of Groups	User-defined. Allows you to select one or more Opswise groups that this record definition belongs to. Click on the lock icon to unlock the field and select groups.
Description	User-defined. Provides a description for the calendar.
Business Days	User-defined. Allows the user to select which days of the week constitute business days for this calendar.
Version	Task definition only; system-supplied. The version number of the current record, which is incremented by the system every time a user updates a record. Click on the Versions tab to view previous versions. For details, see Record Versioning.
Submit button	Submits the new record to the database.
Update button	Saves updates to the record.
Calendar Preview button	Opens a new browser and displays a calendar for the next four years (including the current year). All the dates specified in this calendar are highlighted and identified.
Delete button	Deletes the current record.
Custom Days tab	Displays all custom days associated with this calendar.
Versions tab	Stores copies of all previous versions of the current record (see Record Versioning).

Assigning Existing Custom Days to the Calendar

1. Open the calendar to which you want to assign one or more existing custom days.
2. Click on the **Custom Days** tab or scroll down to that section.
3. Click the **Edit** button. The Edit Members form displays:



Copy of Opswise - American Calendar

Collection:

Ops - Columbus Day
Ops - Veterans Day

Has Custom Days List:

Ops - Christmas
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

Name	Ops - Veterans Day
Description	Veterans Day - Nov 11th
Holiday	true

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

4. The days listed under Collection are existing Custom Days that do not already belong to this calendar. The days listed under Has Custom Days List are days that belong to this calendar. You can filter the days listed under Collection as follows:
   a. Click **Add Filter**. See To create a filter for information about how to construct a filter.
   b. When you have defined the filter you want, click **Run Filter**. The Collection list now displays only those days that match the filter.
   c. To remove filtering, click the minus sign to the left of each condition, then click **Run Filter** again.
5. To add to or remove days from the Has Custom Days List:
   - To add a day to the list, double-click on the day in the Collection list.
   - To remove a day from the list, double-click on the name in the Has Custom Days List.
6. As you click on a day the system displays details about the day at the bottom of the form.
7. When you are finished, click **Save**.

Adding New Custom Days and Assigning Them to this Calendar

1. Open the calendar to which you want to add and assign custom days.
2. Click on the **Custom Days** tab or scroll down to that section.
3. Click **New**. The system displays the Custom Days screen.
4. Fill in the fields using Creating Custom Days for guidance.
5. When you click **Submit** from the calendar form, the custom day is automatically added to the Has Custom Days List.

Copying Calendars

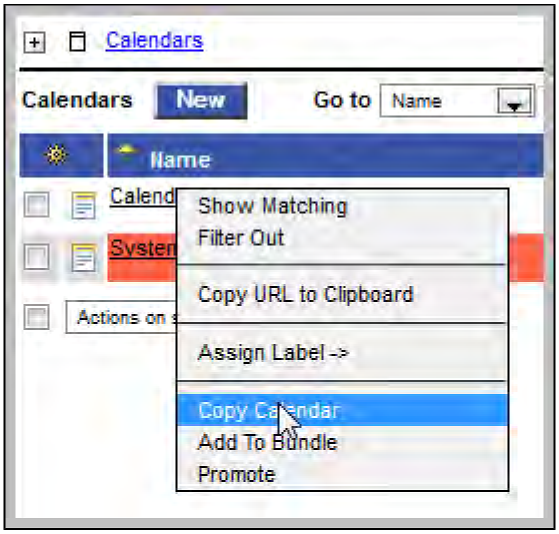
Overview

You can make copies of all Opswise records, including calendars, using the standard methods for Saving, Updating, Deleting, and Copying Records. However, the standard method does not make copies of other records that may be attached to the calendar, such as Notifications, Actions, Variables, and so on.

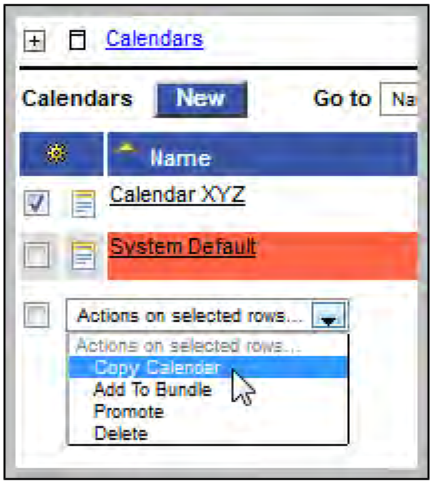
The Copy Calendar option allows you to make a complete copy of a calendar, including all of its associated records, such as variables and notes. It does not copy referenced records, such as virtual resources, but retains the relationship to these records for the copied calendar.

Copying One or More Calendars on the Calendar List

1. Display the calendar list by selecting **Calenders** from the navigation pane.
2. Locate the calendars you want to copy (see Searching for Records).
3. Two methods are available for copying calendars:
   a. To copy a single calendar, hover over the calendar name and right-click. From the pop-up menu, select **Copy Calendar**.



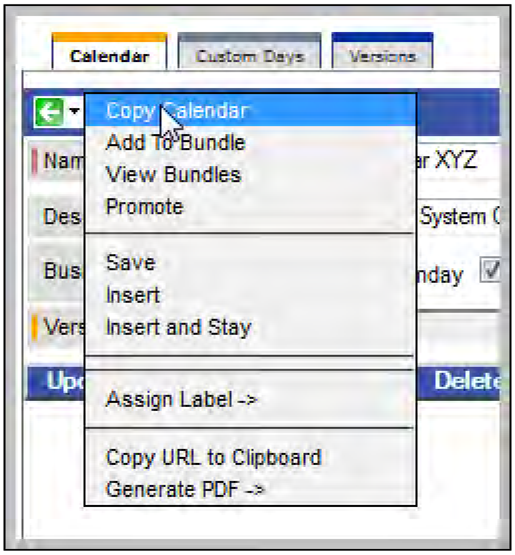
   b. To copy multiple calendars, click the box to the left of each calendar name. From the **Action on selected rows...** drop-down list at the bottom of the page, select **Copy Calendar**.



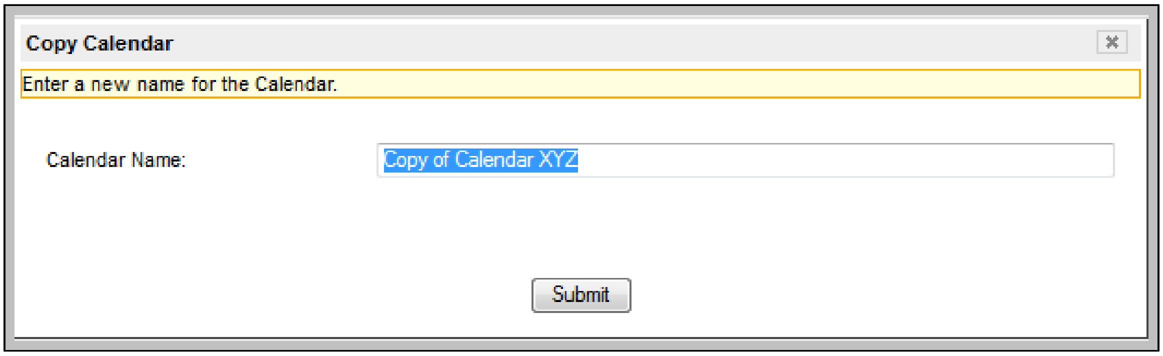
4. The system copies the calendar(s), automatically creating the new name by prepending the original name with "Copy of" (for example, "Copy of Calendar XYZ"), and adds it to the list. If the new name already exists, the system appends a counter to the name, such as "Copy of Calendar XYZ 1", "Copy of Calendar XYZ 2", and so on, until it finds a name that is available.
5. To modify the name or any other information in the calendar, open the new calendar, make your changes, and click **Update**.

Copying a Calendar on the Calendar Definition Form

1. Open the calendar you want to copy.
2. Hover over the down arrow above **Calendar Name:**. A pop-up menu displays:



3. Click **Copy Calendar**. Another window appears, prompting for a name for the new calendar. The default is the original calendar name, prepended with "Copy of," as shown in the following example:



Copy Calendar

Enter a new name for the Calendar.

Calendar Name: Copy of Calendar XYZ

4. Enter a new name for the calendar and click **Submit**. The system copies the calendar and all its attachments and saves it under the new

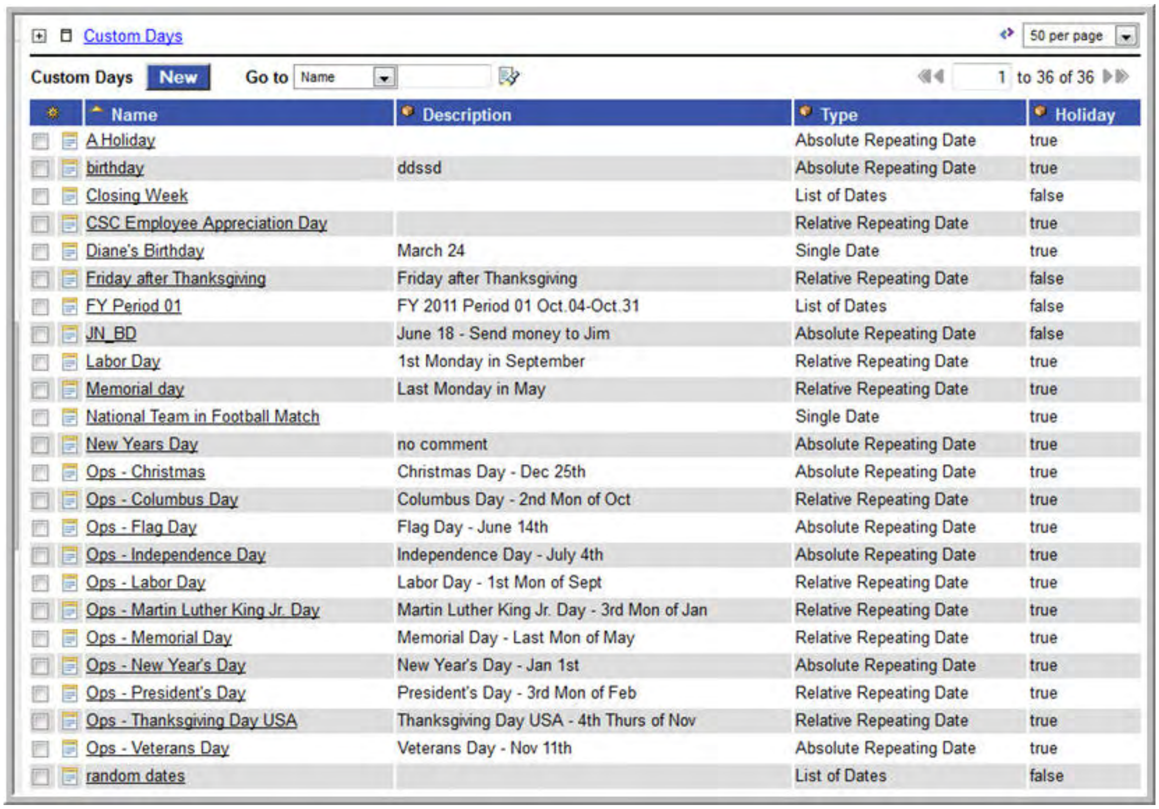
name. (If the new name already exists, the copy will fail.)

Creating Custom Days

Creating Custom Days

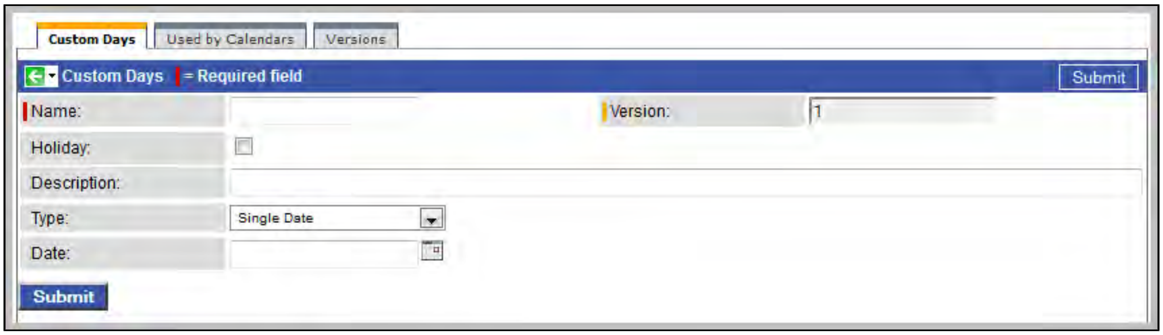
A custom days definition might define a single one-time date, a repeating date, or a list of dates. Custom days are attached to calendars.

1. From the Navigation pane, select **Automation Center > Custom Days**. The Custom Days List screen displays:



Name	Description	Type	Holiday
A Holiday		Absolute Repeating Date	true
birthday	ddssd	Absolute Repeating Date	true
Closing Week		List of Dates	false
CSC Employee Appreciation Day		Relative Repeating Date	true
Diane's Birthday	March 24	Single Date	true
Friday after Thanksgiving	Friday after Thanksgiving	Relative Repeating Date	false
FY Period 01	FY 2011 Period 01 Oct.04-Oct.31	List of Dates	false
JN_BD	June 18 - Send money to Jim	Absolute Repeating Date	false
Labor Day	1st Monday in September	Relative Repeating Date	true
Memorial day	Last Monday in May	Relative Repeating Date	true
National Team in Football Match		Single Date	true
New Years Day	no comment	Absolute Repeating Date	true
Ops - Christmas	Christmas Day - Dec 25th	Absolute Repeating Date	true
Ops - Columbus Day	Columbus Day - 2nd Mon of Oct	Relative Repeating Date	true
Ops - Flag Day	Flag Day - June 14th	Absolute Repeating Date	true
Ops - Independence Day	Independence Day - July 4th	Absolute Repeating Date	true
Ops - Labor Day	Labor Day - 1st Mon of Sept	Relative Repeating Date	true
Ops - Martin Luther King Jr. Day	Martin Luther King Jr. Day - 3rd Mon of Jan	Relative Repeating Date	true
Ops - Memorial Day	Memorial Day - Last Mon of May	Relative Repeating Date	true
Ops - New Year's Day	New Year's Day - Jan 1st	Absolute Repeating Date	true
Ops - President's Day	President's Day - 3rd Mon of Feb	Relative Repeating Date	true
Ops - Thanksgiving Day USA	Thanksgiving Day USA - 4th Thurs of Nov	Relative Repeating Date	true
Ops - Veterans Day	Veterans Day - Nov 11th	Absolute Repeating Date	true
random dates		List of Dates	false

2. Click **New**. The Custom Days Definition screen displays.



Name:		Version:	1
Holiday:	□		
Description:			
Type:	Single Date		
Date:			

3. Using the field descriptions provided below, fill in the fields.
4. Click the **Submit** button to save the record and return to the menu or right-click on the title bar and select **Save** to save the record and remain on the current display.

Custom Days Field Descriptions

The following table provides descriptions of the fields on the custom days form.

Field Name	Description
Name	Descriptive name for this Custom Day.
Version	System-supplied. The version number of the current record, which is incremented by the system every time a user updates a record. Click the Versions tab to view previous versions. For details, see Record Versioning.
Holiday	Indication of whether or not this Custom Days record is defining a holiday. Dates flagged as holidays come into play when the user enables Special Restriction on a trigger and selects a situation of On Holiday.
Description	Description of this day, which displays on the Custom Days list.
Type	Options: • Single Date - Any one-time date. • Relative Repeating Date - An annual (repeating) date that changes from year to year. For example, the U.S. Thanksgiving falls on the 4th Thursday of November, and is therefore on a different date every year. • Absolute Repeating Date - An annual (repeating) date that does not change from year to year. For example, the Canadian holiday Canada Day falls on July 1st of every year. • List of Dates - The dates are listed below.
When	If Type = Relative Repeating Date, the type of selection. Options: 1st, 2nd, 3rd, 4th, Last. Example: The fourth Thursday of November.
Day of Week	If Type = Relative Repeating Date, the day. Example: The fourth Thursday of November.
Month - Relative Repeating Date	If Type = Relative Repeating Date, the month of the year, or All. Example: The fourth Thursday of November.
Month - Absolute Repeating Date	If Type = Absolute Repeating Date, the month. Options: January through December, or All. Example: December 25.
Day - Absolute Repeating Date	If Type = Absolute Repeating Date, the day. Options 1 through 31. Example: December 25.
Day - Single Date	If type = Single Date, this field allows you to enter or select a single date.
Date - List of Dates	If Type = List of dates, this field allows you to manually add a series of dates by typing in each one and clicking the **Add** button.
Submit button	Submits the new record to the database.
Update button	Saves updates to the record.
List Qualifying Dates button	Opens a new browser tab and displays a list of the next 20 dates on which this day occurs.
Delete button	Deletes the current record.

Used by Calendars tab	Displays all calendars that use this custom day.
Versions tab	Stores copies of all previous versions of the current record (see Record Versioning).

Monitoring Task Activity

Opswise lets you perform the following actions related to the monitoring of tasks and Workflows:

- Monitoring Activity from the Activity Screen
- Monitoring Activity from the All Tasks Instances Screen
- Monitoring the Activity History Report
- Monitoring Workflows
- Displaying Task Status
- Retrieving Output

Monitoring Activity from the Activity Screen

Overview

The Activity screen is a real-time display of task status and the Opswise central console of activity. It displays a selected group (or all) task instances, controlled by the Activity Report selected in the drop-down menu. The selected report also defines what columns are displayed.

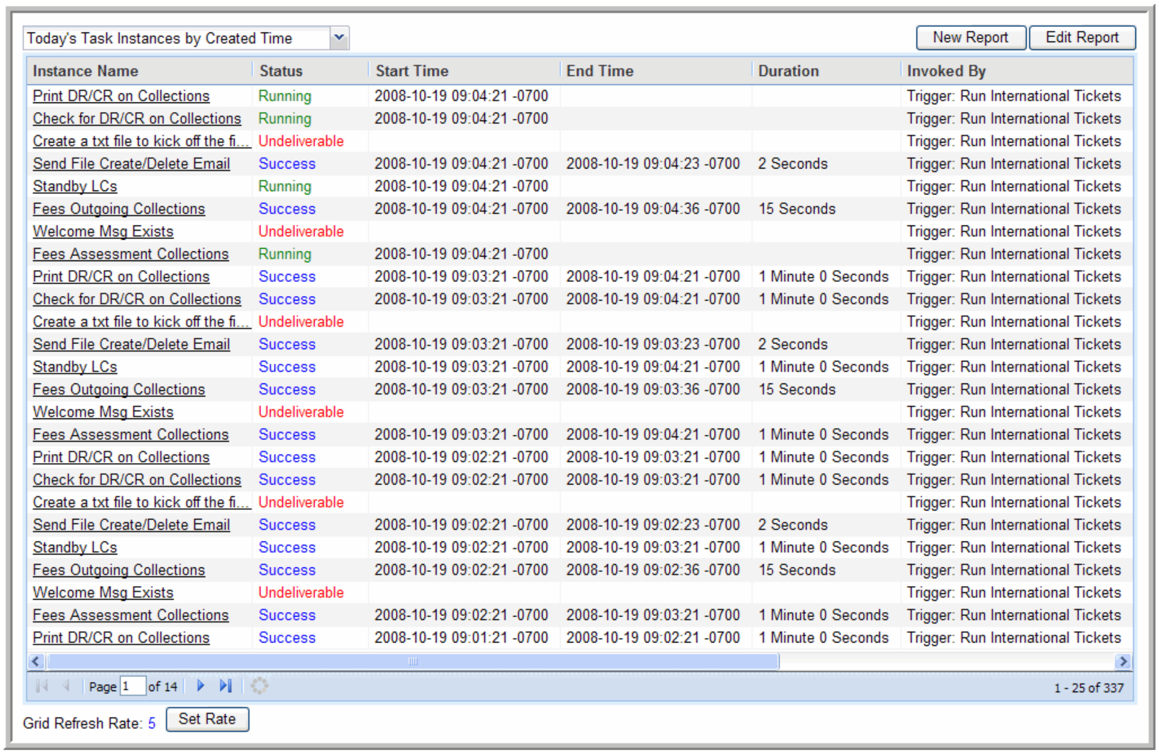
A task instance is the "run" version of a task. Each time a task runs, the system creates a task instance and monitors its activity in the Activity screen. Each task instance is a separate record.

The Activity screen display allows you to issue commands against task instances. You can also issue commands using All Task Instances. In cases where the task definition did not instruct the system to retrieve output automatically, you can retrieve output manually from any completed task.

Accessing the Activity Screen

From the navigation pane, select **Automation Center > Activity**. The Activity screen displays, based on the most recently selected report.

Below is a sample Activity screen display, using the report, "Today's Task Instances by Created Time."



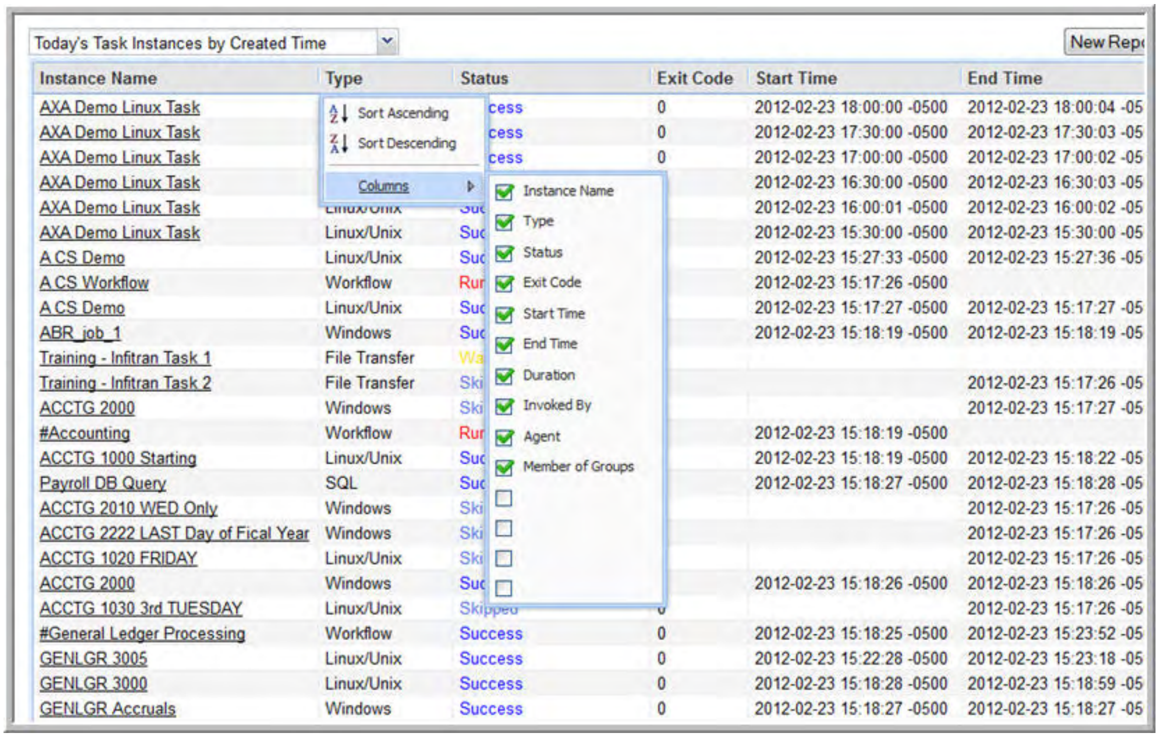
Instance Name	Status	Start Time	End Time	Duration	Invoked By
Print DR/CR on Collections	Running	2008-10-19 09:04:21 -0700			Trigger: Run International Tickets
Check for DR/CR on Collections	Running	2008-10-19 09:04:21 -0700			Trigger: Run International Tickets
Create a txt file to kick off the fi...	Undeliverable				Trigger: Run International Tickets
Send File Create/Delete Email	Success	2008-10-19 09:04:21 -0700	2008-10-19 09:04:23 -0700	2 Seconds	Trigger: Run International Tickets
Standby LCs	Running	2008-10-19 09:04:21 -0700			Trigger: Run International Tickets
Fees Outgoing Collections	Success	2008-10-19 09:04:21 -0700	2008-10-19 09:04:36 -0700	15 Seconds	Trigger: Run International Tickets
Welcome Msg Exists	Undeliverable				Trigger: Run International Tickets
Fees Assessment Collections	Running	2008-10-19 09:04:21 -0700			Trigger: Run International Tickets
Print DR/CR on Collections	Success	2008-10-19 09:03:21 -0700	2008-10-19 09:04:21 -0700	1 Minute 0 Seconds	Trigger: Run International Tickets
Check for DR/CR on Collections	Success	2008-10-19 09:03:21 -0700	2008-10-19 09:04:21 -0700	1 Minute 0 Seconds	Trigger: Run International Tickets
Create a txt file to kick off the fi...	Undeliverable				Trigger: Run International Tickets
Send File Create/Delete Email	Success	2008-10-19 09:03:21 -0700	2008-10-19 09:03:23 -0700	2 Seconds	Trigger: Run International Tickets
Standby LCs	Success	2008-10-19 09:03:21 -0700	2008-10-19 09:04:21 -0700	1 Minute 0 Seconds	Trigger: Run International Tickets
Fees Outgoing Collections	Success	2008-10-19 09:03:21 -0700	2008-10-19 09:03:36 -0700	15 Seconds	Trigger: Run International Tickets
Welcome Msg Exists	Undeliverable				Trigger: Run International Tickets
Fees Assessment Collections	Success	2008-10-19 09:03:21 -0700	2008-10-19 09:04:21 -0700	1 Minute 0 Seconds	Trigger: Run International Tickets
Print DR/CR on Collections	Success	2008-10-19 09:02:21 -0700	2008-10-19 09:03:21 -0700	1 Minute 0 Seconds	Trigger: Run International Tickets
Check for DR/CR on Collections	Success	2008-10-19 09:02:21 -0700	2008-10-19 09:03:21 -0700	1 Minute 0 Seconds	Trigger: Run International Tickets
Create a txt file to kick off the fi...	Undeliverable				Trigger: Run International Tickets
Send File Create/Delete Email	Success	2008-10-19 09:02:21 -0700	2008-10-19 09:02:23 -0700	2 Seconds	Trigger: Run International Tickets
Standby LCs	Success	2008-10-19 09:02:21 -0700	2008-10-19 09:03:21 -0700	1 Minute 0 Seconds	Trigger: Run International Tickets
Fees Outgoing Collections	Success	2008-10-19 09:02:21 -0700	2008-10-19 09:02:36 -0700	15 Seconds	Trigger: Run International Tickets
Welcome Msg Exists	Undeliverable				Trigger: Run International Tickets
Fees Assessment Collections	Success	2008-10-19 09:02:21 -0700	2008-10-19 09:03:21 -0700	1 Minute 0 Seconds	Trigger: Run International Tickets
Print DR/CR on Collections	Success	2008-10-19 09:01:21 -0700	2008-10-19 09:02:21 -0700	1 Minute 0 Seconds	Trigger: Run International Tickets

Grid Refresh Rate: 5

Changing the Sorting

You can change the sorting on the display and add or remove columns.

1. Right-click on the title of the column you want to sort by. A pop-up menu displays.



Instance Name	Type	Status	Exit Code	Start Time	End Time
AXA Demo Linux Task		cess	0	2012-02-23 18:00:00 -0500	2012-02-23 18:00:04 -05
AXA Demo Linux Task		cess	0	2012-02-23 17:30:00 -0500	2012-02-23 17:30:03 -05
AXA Demo Linux Task		cess	0	2012-02-23 17:00:00 -0500	2012-02-23 17:00:02 -05
AXA Demo Linux Task				2012-02-23 16:30:00 -0500	2012-02-23 16:30:03 -05
AXA Demo Linux Task	Linux/Unix	Suc		2012-02-23 16:00:01 -0500	2012-02-23 16:00:02 -05
AXA Demo Linux Task	Linux/Unix	Suc		2012-02-23 15:30:00 -0500	2012-02-23 15:30:00 -05
A CS Demo	Linux/Unix	Suc		2012-02-23 15:27:33 -0500	2012-02-23 15:27:36 -05
A CS Workflow	Workflow	Rur		2012-02-23 15:17:26 -0500	
A CS Demo	Linux/Unix	Suc		2012-02-23 15:17:27 -0500	2012-02-23 15:17:27 -05
ABR_job_1	Windows	Suc		2012-02-23 15:18:19 -0500	2012-02-23 15:18:19 -05
Training - Infitran Task 1	File Transfer	Wa			
Training - Infitran Task 2	File Transfer	Ski			2012-02-23 15:17:26 -05
ACCTG 2000	Windows	Ski			2012-02-23 15:17:27 -05
#Accounting	Workflow	Rur		2012-02-23 15:18:19 -0500	
ACCTG 1000 Starting	Linux/Unix	Suc		2012-02-23 15:18:19 -0500	2012-02-23 15:18:22 -05
Payroll DB Query	SQL	Suc		2012-02-23 15:18:27 -0500	2012-02-23 15:18:28 -05
ACCTG 2010 WED Only	Windows	Ski			2012-02-23 15:17:26 -05
ACCTG 2222 LAST Day of Fical Year	Windows	Ski			2012-02-23 15:17:26 -05
ACCTG 1020 FRIDAY	Linux/Unix	Ski			2012-02-23 15:17:26 -05
ACCTG 2000	Windows	Suc		2012-02-23 15:18:26 -0500	2012-02-23 15:18:26 -05
ACCTG 1030 3rd TUESDAY	Linux/Unix	Skipped	0		2012-02-23 15:17:26 -05
#General Ledger Processing	Workflow	Success	0	2012-02-23 15:18:25 -0500	2012-02-23 15:23:52 -05
GENLGR 3005	Linux/Unix	Success	0	2012-02-23 15:22:28 -0500	2012-02-23 15:23:18 -05
GENLGR 3000	Linux/Unix	Success	0	2012-02-23 15:18:28 -0500	2012-02-23 15:18:59 -05
GENLGR Accruals	Windows	Success	0	2012-02-23 15:18:27 -0500	2012-02-23 15:18:27 -05

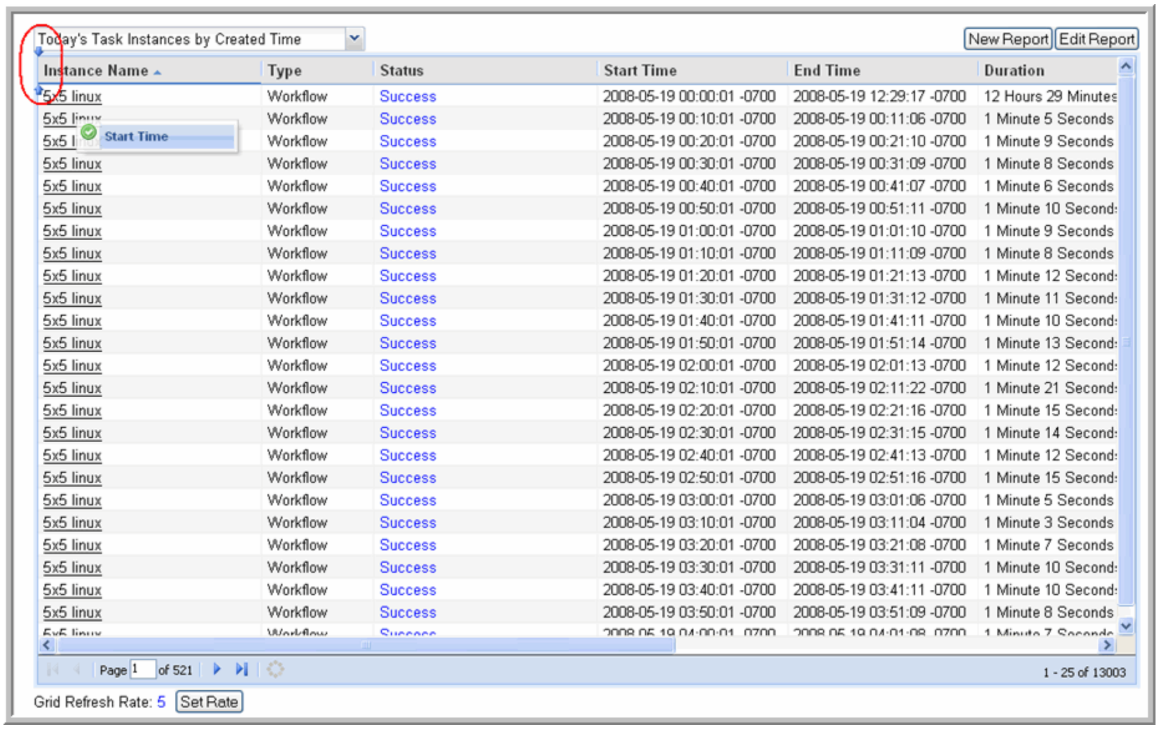
2. To change the sorting, click **Sort Ascending** or **Sort Descending**.

Changing the Columns

You can move columns around on the screen or add or remove columns.

Moving Columns

1. Highlight the column you want to move.
2. Drag it to a new location. The arrows indicate where the column will be placed, as shown.



Today's Task Instances by Created Time

Instance Name	Type	Status	Start Time	End Time	Duration
5x5 linux	Workflow	Success	2008-05-19 00:00:01 -0700	2008-05-19 12:29:17 -0700	12 Hours 29 Minutes
5x5 linux	Workflow	Success	2008-05-19 00:10:01 -0700	2008-05-19 00:11:06 -0700	1 Minute 5 Seconds
5x5 linux	Workflow	Success	2008-05-19 00:20:01 -0700	2008-05-19 00:21:10 -0700	1 Minute 9 Seconds
5x5 linux	Workflow	Success	2008-05-19 00:30:01 -0700	2008-05-19 00:31:09 -0700	1 Minute 8 Seconds
5x5 linux	Workflow	Success	2008-05-19 00:40:01 -0700	2008-05-19 00:41:07 -0700	1 Minute 6 Seconds
5x5 linux	Workflow	Success	2008-05-19 00:50:01 -0700	2008-05-19 00:51:11 -0700	1 Minute 10 Second:
5x5 linux	Workflow	Success	2008-05-19 01:00:01 -0700	2008-05-19 01:01:10 -0700	1 Minute 9 Seconds
5x5 linux	Workflow	Success	2008-05-19 01:10:01 -0700	2008-05-19 01:11:09 -0700	1 Minute 8 Seconds
5x5 linux	Workflow	Success	2008-05-19 01:20:01 -0700	2008-05-19 01:21:13 -0700	1 Minute 12 Second:
5x5 linux	Workflow	Success	2008-05-19 01:30:01 -0700	2008-05-19 01:31:12 -0700	1 Minute 11 Second:
5x5 linux	Workflow	Success	2008-05-19 01:40:01 -0700	2008-05-19 01:41:11 -0700	1 Minute 10 Second:
5x5 linux	Workflow	Success	2008-05-19 01:50:01 -0700	2008-05-19 01:51:14 -0700	1 Minute 13 Second:
5x5 linux	Workflow	Success	2008-05-19 02:00:01 -0700	2008-05-19 02:01:13 -0700	1 Minute 12 Second:
5x5 linux	Workflow	Success	2008-05-19 02:10:01 -0700	2008-05-19 02:11:22 -0700	1 Minute 21 Second:
5x5 linux	Workflow	Success	2008-05-19 02:20:01 -0700	2008-05-19 02:21:16 -0700	1 Minute 15 Second:
5x5 linux	Workflow	Success	2008-05-19 02:30:01 -0700	2008-05-19 02:31:15 -0700	1 Minute 14 Second:
5x5 linux	Workflow	Success	2008-05-19 02:40:01 -0700	2008-05-19 02:41:13 -0700	1 Minute 12 Second:
5x5 linux	Workflow	Success	2008-05-19 02:50:01 -0700	2008-05-19 02:51:16 -0700	1 Minute 15 Second:
5x5 linux	Workflow	Success	2008-05-19 03:00:01 -0700	2008-05-19 03:01:06 -0700	1 Minute 5 Seconds
5x5 linux	Workflow	Success	2008-05-19 03:10:01 -0700	2008-05-19 03:11:04 -0700	1 Minute 3 Seconds
5x5 linux	Workflow	Success	2008-05-19 03:20:01 -0700	2008-05-19 03:21:08 -0700	1 Minute 7 Seconds
5x5 linux	Workflow	Success	2008-05-19 03:30:01 -0700	2008-05-19 03:31:11 -0700	1 Minute 10 Second:
5x5 linux	Workflow	Success	2008-05-19 03:40:01 -0700	2008-05-19 03:41:11 -0700	1 Minute 10 Second:
5x5 linux	Workflow	Success	2008-05-19 03:50:01 -0700	2008-05-19 03:51:09 -0700	1 Minute 8 Seconds
5x5 linux	Workflow	Success	2008-05-19 04:00:01 -0700	2008-05-19 04:01:08 -0700	1 Minute 7 Seconds

Page 1 of 521

1 - 25 of 13003

Grid Refresh Rate: 5

3. Drop the column when the arrows are pointing to the location you want.

Adding or Removing Columns

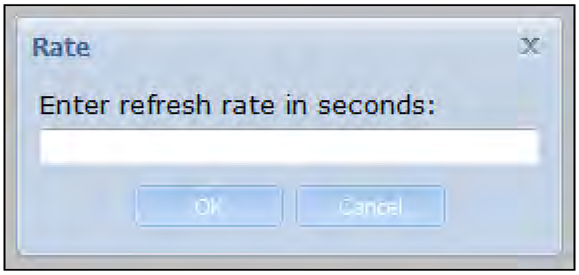
1. Right-click on any column title.
2. Click **Columns**.
3. Select or deselect the columns you want to add or remove, respectively.

To make more considerable changes to the column display, update the report.

Changing the Refresh Rate

The default refresh rate on the Activity display is every 5 seconds. To change the rate:

1. Click **Set Rate**.



Rate

Enter refresh rate in seconds:

2. Enter the new rate in seconds, and click **OK** to save or **Cancel** to cancel the change.

Column Descriptions

Described below are the columns on the Activity display.

Column Name	Description

<table>
<tr><td>Instance Name</td><td>Required. Name used within Opswise to identify this task. Up to 40 alphanumerics; variables supported. It is the user's responsibility to develop a workable naming scheme for tasks.</td></tr>
<tr><td>Status</td><td>Status of this task.
<table>
<tr><th>Status</th><th>Description</th></tr>
<tr><td><b>Defined</b></td><td>All task types. The new task instance has been created (the task has been launched). Not yet implemented.</td></tr>
<tr><td><b>Waiting</b></td><td>All task types. The task has been loaded by a workflow and is waiting to run.</td></tr>
<tr><td><b>Held</b></td><td>All task types. The task has been put on hold by a user.</td></tr>
<tr><td><b>Resource Requested</b></td><td>All tasks with a virtual resource defined go immediately to a status of Resource Requested. If the resource is available, the task then moves to the next appropriate processing status.</td></tr>
<tr><td><b>Resource Wait</b></td><td>All tasks with a virtual resource defined go immediately to a status of Resource Requested. If the resource is not available, the task goes to a status of Resource Wait. When the resource becomes available, the task moves to the next appropriate processing status</td></tr>
<tr><td><b>Undeliverable</b></td><td>Agent-based tasks. The agent is unavailable.</td></tr>
<tr><td><b>Queued</b></td><td>Agent-based tasks only. The task has been queued on a resource.</td></tr>
<tr><td><b>Submitted</b></td><td>z/OS only. The task has been submitted to the z/OS Job Entry subsystem and scheduled by the z/OS Job Scheduler.</td></tr>
<tr><td><b>Action Required</b></td><td>Manual tasks only. When a manual task launches, it goes into Action Required status, meaning a user must perform some manual activity. For details, see Manual task.</td></tr>
<tr><td><b>Started</b></td><td>Agent-based and Manual tasks only. The task has started. For agent-based tasks, this means the agent has received the task.</td></tr>
<tr><td><b>Running</b></td><td>All task types. The task is running. For agent-based tasks, the agent has started running the program.</td></tr>
<tr><td><b>Running Problems</b></td><td>Workflows only. One or more tasks within the workflow has one of the following statuses:<ul><li>Held</li><li>Undeliverable</li><li>Running Problems (for sub-workflows)</li><li>Cancel Pending</li><li>In Doubt</li><li>Start Failure</li><li>Cancelled</li></ul></td></tr>
<tr><td><b>In Doubt</b></td><td>Agent-based tasks only. The agent is "in doubt" about the current status of the task instance. This may occur if an agent or agent connection goes down. In this case, the agent restarts and reviews its data about tasks in progress. If the agent finds a task still running, it resumes normal monitoring. If the agent cannot find the task, this usually indicates that the task completed, but the agent considers the task status to be "in doubt."</td></tr>
<tr><td><b>Start Failure</b></td><td>All task types. The task was unable to start.</td></tr>
<tr><td><b>Confirmation Required</b></td><td>z/OS only. If you make JCL changes and restart a z/OS task, Opswise will put the task into Confirmation Required status and prompt you for a confirmation. For detailed processing steps, see Rerunning a z/OS Task.</td></tr>
<tr><td><b>Cancelled</b></td><td>All task types. The task was cancelled by a user.</td></tr>
<tr><td><b>Failed</b></td><td>All task types. The task ran to a failure status.</td></tr>
<tr><td><b>Skipped</b></td><td>All task types. The task was skipped by a user.</td></tr>
<tr><td><b>Finished</b></td><td>All task types. The task was forced by the user to finish. The user may do this in cases where the task had "Cancelled" or "Failed" status, and the user needed to release other task instances depending on the successful completion of this task instance in a workflow. For more information, see Force Finishing a Task.</td></tr>
<tr><td><b>Success</b></td><td>All task types. The task has completed successfully.</td></tr>
</table>
</td></tr>
</table>

Invoked By	Task instance only; system-supplied. How the task instance was launched. One of the following: • Trigger: (Trigger Name) - The instance was launched by the named trigger. • Workflow: (Workflow Name) - The instance was launched by the named workflow. • Manually Launched - The instance was launched by a user. To determine the name of the user: 1. From the Activity or All Task Instances screen, click the task instance name to open the record. 2. The Execution User field identifies the user who launched the task instance.
Start Time	Task instance only; system-supplied. The date and time the task started.
End Time	Task instance only; system-supplied. The date and time the task instance completed.
Type	Type of task instance.
Duration	Task instance only; system-supplied. The amount of time the task took to run.
Agent	Required. Name used within Opswise Automation Center to identify this resource. Up to 40 alphanumerics. It is the user's responsibility to develop a workable naming scheme for resources.
Member of Groups	User-defined. Allows you to select one or more Opswise groups that this record definition belongs to. Click on the lock icon to unlock the field and select groups.

Changing the Activity Report

You can change the selection in your Activity screen by selecting an existing Activity Report or creating a new one.

Selecting a Different Activity Report

To select a different Activity Report, click the Activity Report dropdown and click on the report you want to display. The following default reports are available:

Active Task Instances — Displays all instances with an active status, including: Defined, Waiting, Held, Waiting for Resources, Action Required, Queued, Started, Running, In Doubt.

Active Workflow Task Instances — Displays all workflow instances with an active status, including: Defined, Waiting, Held, Waiting for Resources, Action Required, Queued, Started, Running, In Doubt.

Active/Late Task Instances — Displays all instances with a started late flag set to true and an active status, including: Defined, Waiting, Held, Waiting for Resources, Action Required, Queued, Started, Running, In Doubt.

All Task Instance(s) by Status — Displays all instances, sorted by status.

Cancelled Task Instances — Displays all instances with a status of Cancelled.

Held Task Instance(s) — Displays all instances that were created on the current date and have a status of Held.

In Doubt Task Instances — Displays all task instances with a status of In Doubt.

Queued Task Instances — Displays all instances with a status of Queued.

Running Task Instances — Displays all instances with a status of Running.

Task Instances Due to Finish in the Next 3 Hours — For tasks where forecasting information is available, lists all tasks due to finish in the next three hours.

Task Instances Due to Finish in the Next Hour — For tasks where forecasting information is available, lists all tasks due to finish in the next hour.

Today's Failed Task Instances by Status — Displays all instances that were created on the current date and have one of the following statuses:

Failed, Cancelled, Start Failure.

Today's Successful Task Instances — Displays all instances that were created on the current date and have a status of Success.

Today's Task Instance(s) by Created Time — Displays all instances that were created on the current date, sorted by created time.

Today's Task Instance(s) by Type — Displays all instances that were created on the current date, sorted by type.

Undeliverable Task Instances — Displays all instances with a status of Undeliverable (agent not available).

Unsuccessful Task Instances — Displays all instances with a status of Failed, Cancelled, Start Failure, Finished.

Waiting for Resources Task Instances — Displays all instances with a status of Waiting for Resources.

Waiting Task Instances — Displays all instances with a status of Waiting.

Workflow Task Instances — Displays all workflows, not including the task instances within the workflows.

Workflow Task Instances with Problems — Displays all workflows with a status of Running Problems, not including the task instances within the workflows.

Creating a New Activity Report

Activity Reports select data about task instances from an Opswise table called the Activity ops_exec table. To create an Activity Report, you use the same form that is used to create reports. When you are generating normal reports, you can use any table from the Opswise database; when you create Activity Reports, you only use the Activity ops_exec table.

1. From the Activity Report screen, click **New Report**. The New Report form displays.
2. For an Activity Report, you must specify the following:
   - For Type, specify List.
   - For the Table field, select Activity ops_exec; otherwise, the saved report will not appear in the Activity Reports dropdown.
3. Specify the Activity Report parameters. For assistance in selecting columns, refer to the field descriptions provided with each task type.
4. To save the new Activity Report and add it to the drop-down, click **Save**. The report runs immediately and displays in the lower half of the screen. The new Activity Report is also added to the Activity Report drop-down menu.

Updating, Copying, Deleting Activity Reports

All Activity Reports are stored under **Automation Center > Reports**, under the **Activity** heading within one of the following sections:

- My Saved reports (if Visible to = Me)
- My Groups' reports (if Visible to = Automation Center_SysAdmin)
- Global reports (if Visible to = Everyone)
  To update, copy, or delete an Activity Report, select **Automation Center > Reports** and scroll to the Activity section where the report is stored, as described above.

Updating an Activity Report

1. Display the report you want to modify.
2. Make your changes and click **Update**.

Copying an Activity Report

1. Display the report you want to copy.
2. Give the report a new name, specify your changes, if any, and click **Insert**.

Deleting an Activity Report

1. Display the report you want to delete.
2. Click **Delete**.

Displaying Details about Task Instances

You can view details about task instances currently being displayed on the Activity screen. Click on the Instance Name.

- If the task is a workflow, the worklow monitor opens. If you want to view record details such as status description about the workflow, use Monitoring Activity from the All Tasks Instances Screen.
- If the task is not a workflow, the system displays the record details on the task screen.

From this screen, you can do the following:

- If the task was triggered by a workflow, you can view the parent workflow by clicking the **View Parent** button. This jumps to the workflow record. To display the workflow monitor, click **View Workflow**.
- Where applicable, you can manually intervene in processing by issuing a command.
- To show all the details of this task instance, click **Show Details**. A new browser tab opens and displays a report of the task instance. For a description of the data being displayed, see the documentation for that task type.
- If the task instance is complete and it generated any output, you can view it by clicking the Output tab.

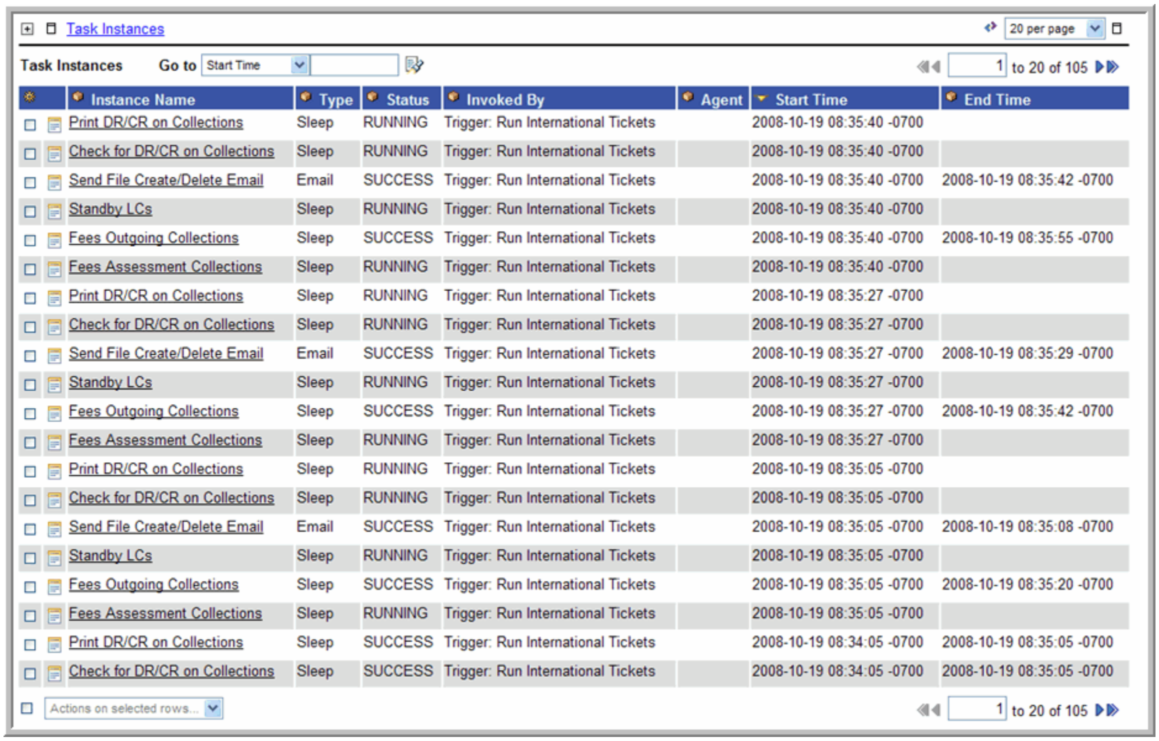
Monitoring Activity from the All Tasks Instances Screen

Overview

The All Task Instances screen displays the same information as the Activity screen, but the display is not automatically refreshed. This screen allows you to issue commands against multiple tasks and provides more extensive filtering capabilities. All Task Instances also allows you to view details about workflow instances – information that is not available from the Activity screen.

Accessing the All Task Instances Screen

From the navigation pane, select **Automation Center > All Task Instances**. Below is a sample All Task Instances screen display.

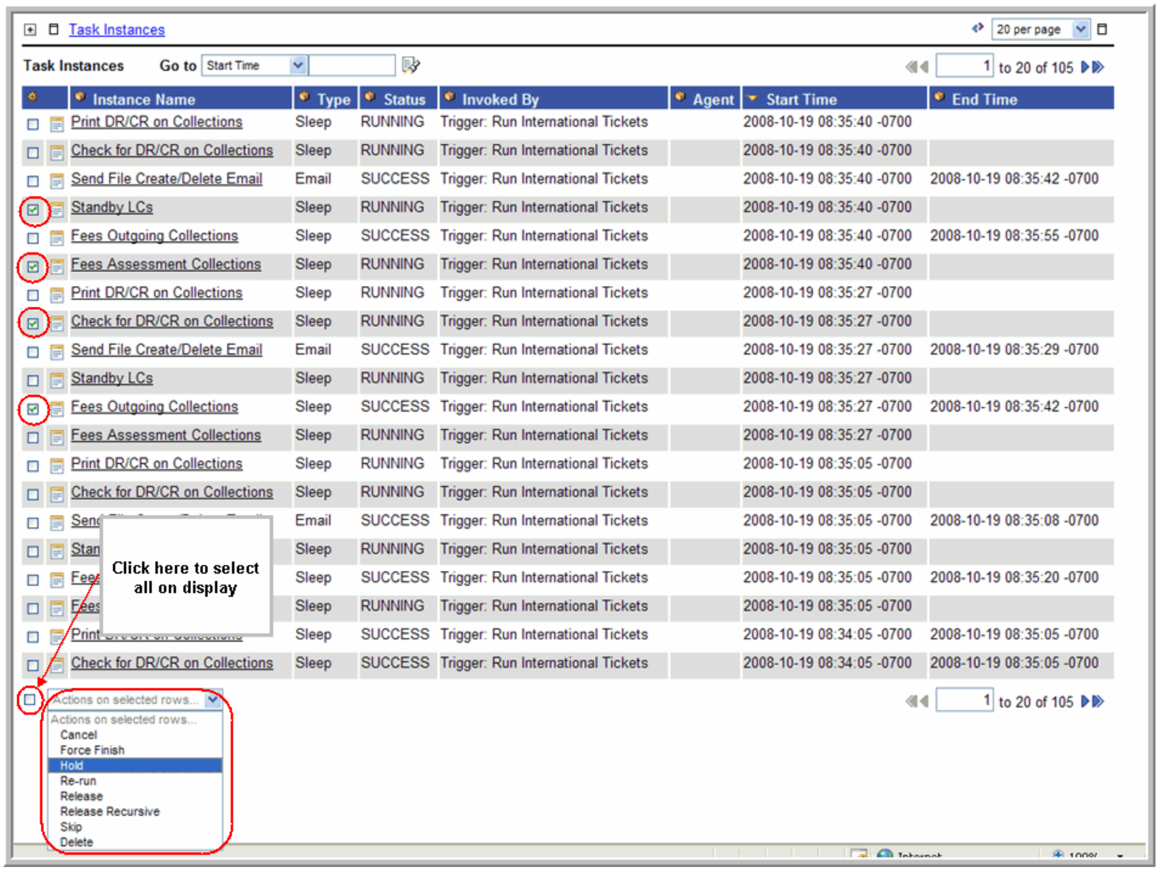


	Instance Name	Type	Status	Invoked By	Agent	Start Time	End Time
□	Print DR/CR on Collections	Sleep	RUNNING	Trigger: Run International Tickets		2008-10-19 08:35:40 -0700	
□	Check for DR/CR on Collections	Sleep	RUNNING	Trigger: Run International Tickets		2008-10-19 08:35:40 -0700	
□	Send File Create/Delete Email	Email	SUCCESS	Trigger: Run International Tickets		2008-10-19 08:35:40 -0700	2008-10-19 08:35:42 -0700
□	Standby LCs	Sleep	RUNNING	Trigger: Run International Tickets		2008-10-19 08:35:40 -0700	
□	Fees Outgoing Collections	Sleep	SUCCESS	Trigger: Run International Tickets		2008-10-19 08:35:40 -0700	2008-10-19 08:35:55 -0700
□	Fees Assessment Collections	Sleep	RUNNING	Trigger: Run International Tickets		2008-10-19 08:35:40 -0700	
□	Print DR/CR on Collections	Sleep	RUNNING	Trigger: Run International Tickets		2008-10-19 08:35:27 -0700	
□	Check for DR/CR on Collections	Sleep	RUNNING	Trigger: Run International Tickets		2008-10-19 08:35:27 -0700	
□	Send File Create/Delete Email	Email	SUCCESS	Trigger: Run International Tickets		2008-10-19 08:35:27 -0700	2008-10-19 08:35:29 -0700
□	Standby LCs	Sleep	RUNNING	Trigger: Run International Tickets		2008-10-19 08:35:27 -0700	
□	Fees Outgoing Collections	Sleep	SUCCESS	Trigger: Run International Tickets		2008-10-19 08:35:27 -0700	2008-10-19 08:35:42 -0700
□	Fees Assessment Collections	Sleep	RUNNING	Trigger: Run International Tickets		2008-10-19 08:35:27 -0700	
□	Print DR/CR on Collections	Sleep	RUNNING	Trigger: Run International Tickets		2008-10-19 08:35:05 -0700	
□	Check for DR/CR on Collections	Sleep	RUNNING	Trigger: Run International Tickets		2008-10-19 08:35:05 -0700	
□	Send File Create/Delete Email	Email	SUCCESS	Trigger: Run International Tickets		2008-10-19 08:35:05 -0700	2008-10-19 08:35:08 -0700
□	Standby LCs	Sleep	RUNNING	Trigger: Run International Tickets		2008-10-19 08:35:05 -0700	
□	Fees Outgoing Collections	Sleep	SUCCESS	Trigger: Run International Tickets		2008-10-19 08:35:05 -0700	2008-10-19 08:35:20 -0700
□	Fees Assessment Collections	Sleep	RUNNING	Trigger: Run International Tickets		2008-10-19 08:35:05 -0700	
□	Print DR/CR on Collections	Sleep	SUCCESS	Trigger: Run International Tickets		2008-10-19 08:34:05 -0700	2008-10-19 08:35:05 -0700
□	Check for DR/CR on Collections	Sleep	SUCCESS	Trigger: Run International Tickets		2008-10-19 08:34:05 -0700	2008-10-19 08:35:05 -0700

Issuing Commands Against Multiple Task Instances

You can issue commands against one or more task instances as follows:

- Click on the box associated with those records and select the command you want from the **Actions on selected rows...** menu.



	Instance Name	Type	Status	Invoked By	Agent	Start Time	End Time
□	Print DR/CR on Collections	Sleep	RUNNING	Trigger: Run International Tickets		2008-10-19 08:35:40 -0700	
□	Check for DR/CR on Collections	Sleep	RUNNING	Trigger: Run International Tickets		2008-10-19 08:35:40 -0700	
□	Send File Create/Delete Email	Email	SUCCESS	Trigger: Run International Tickets		2008-10-19 08:35:40 -0700	2008-10-19 08:35:42 -0700
☑	Standby LCs	Sleep	RUNNING	Trigger: Run International Tickets		2008-10-19 08:35:40 -0700	
□	Fees Outgoing Collections	Sleep	SUCCESS	Trigger: Run International Tickets		2008-10-19 08:35:40 -0700	2008-10-19 08:35:55 -0700
☑	Fees Assessment Collections	Sleep	RUNNING	Trigger: Run International Tickets		2008-10-19 08:35:40 -0700	
□	Print DR/CR on Collections	Sleep	RUNNING	Trigger: Run International Tickets		2008-10-19 08:35:27 -0700	
☑	Check for DR/CR on Collections	Sleep	RUNNING	Trigger: Run International Tickets		2008-10-19 08:35:27 -0700	
□	Send File Create/Delete Email	Email	SUCCESS	Trigger: Run International Tickets		2008-10-19 08:35:27 -0700	2008-10-19 08:35:29 -0700
□	Standby LCs	Sleep	RUNNING	Trigger: Run International Tickets		2008-10-19 08:35:27 -0700	
☑	Fees Outgoing Collections	Sleep	SUCCESS	Trigger: Run International Tickets		2008-10-19 08:35:27 -0700	2008-10-19 08:35:42 -0700
□	Fees Assessment Collections	Sleep	RUNNING	Trigger: Run International Tickets		2008-10-19 08:35:27 -0700	
□	Print DR/CR on Collections	Sleep	RUNNING	Trigger: Run International Tickets		2008-10-19 08:35:05 -0700	
□	Check for DR/CR on Collections	Sleep	RUNNING	Trigger: Run International Tickets		2008-10-19 08:35:05 -0700	
□	Send	Email	SUCCESS	Trigger: Run International Tickets		2008-10-19 08:35:05 -0700	2008-10-19 08:35:08 -0700
□	Stan	Sleep	RUNNING	Trigger: Run International Tickets		2008-10-19 08:35:05 -0700	
□	Fees	Sleep	SUCCESS	Trigger: Run International Tickets		2008-10-19 08:35:05 -0700	2008-10-19 08:35:20 -0700
□	Fees	Sleep	RUNNING	Trigger: Run International Tickets		2008-10-19 08:35:05 -0700	
□	Print	Sleep	SUCCESS	Trigger: Run International Tickets		2008-10-19 08:34:05 -0700	2008-10-19 08:35:05 -0700
□	Check for DR/CR on Collections	Sleep	SUCCESS	Trigger: Run International Tickets		2008-10-19 08:34:05 -0700	2008-10-19 08:35:05 -0700

- To issue a command against all records, click on the box next to the **Actions on selected rows**... menu and select the command you want to issue. For information about the commands available, see Manually Running and Controlling Tasks.

Column Descriptions

Described below are the columns on All Task Instances.

Field Name	Description
Instance Name	Required. Name used within Opswise to identify this task. Up to 40 alphanumerics; variables supported. It is the user's responsibility to develop a workable naming scheme for tasks.
Type	Type of task instance.

<table>
<tr><td>Status</td><td>
<table>
<tr><th>Status</th><th>Description</th></tr>
<tr><td><b>Defined</b></td><td>All task types. The new task instance has been created (the task has been launched). Not yet implemented.</td></tr>
<tr><td><b>Waiting</b></td><td>All task types. The task has been loaded by a workflow and is waiting to run.</td></tr>
<tr><td><b>Held</b></td><td>All task types. The task has been put on hold by a user.</td></tr>
<tr><td><b>Resource Requested</b></td><td>All tasks with a virtual resource defined go immediately to a status of Resource Requested. If the resource is available, the task then moves to the next appropriate processing status.</td></tr>
<tr><td><b>Resource Wait</b></td><td>All tasks with a virtual resource defined go immediately to a status of Resource Requested. If the resource is not available, the task goes to a status of Resource Wait. When the resource becomes available, the task moves to the next appropriate processing status</td></tr>
<tr><td><b>Undeliverable</b></td><td>Agent-based tasks. The agent is unavailable.</td></tr>
<tr><td><b>Queued</b></td><td>Agent-based tasks only. The task has been queued on a resource.</td></tr>
<tr><td><b>Submitted</b></td><td>z/OS only. The task has been submitted to the z/OS Job Entry subsystem and scheduled by the z/OS Job Scheduler.</td></tr>
<tr><td><b>Action Required</b></td><td>Manual tasks only. When a manual task launches, it goes into Action Required status, meaning a user must perform some manual activity. For details, see Manual task.</td></tr>
<tr><td><b>Started</b></td><td>Agent-based and Manual tasks only. The task has started. For agent-based tasks, this means the agent has received the task.</td></tr>
<tr><td><b>Running</b></td><td>All task types. The task is running. For agent-based tasks, the agent has started running the program.</td></tr>
<tr><td><b>Running Problems</b></td><td>Workflows only. One or more tasks within the workflow has one of the following statuses:
• Held
• Undeliverable
• Running Problems (for sub-workflows)
• Cancel Pending
• In Doubt
• Start Failure
• Cancelled</td></tr>
<tr><td><b>In Doubt</b></td><td>Agent-based tasks only. The agent is "in doubt" about the current status of the task instance. This may occur if an agent or agent connection goes down. In this case, the agent restarts and reviews its data about tasks in progress. If the agent finds a task still running, it resumes normal monitoring. If the agent cannot find the task, this usually indicates that the task completed, but the agent considers the task status to be "in doubt."</td></tr>
<tr><td><b>Start Failure</b></td><td>All task types. The task was unable to start.</td></tr>
<tr><td><b>Confirmation Required</b></td><td>z/OS only. If you make JCL changes and restart a z/OS task, Opswise will put the task into Confirmation Required status and prompt you for a confirmation. For detailed processing steps, see Rerunning a z/OS Task.</td></tr>
<tr><td><b>Cancelled</b></td><td>All task types. The task was cancelled by a user.</td></tr>
<tr><td><b>Failed</b></td><td>All task types. The task ran to a failure status.</td></tr>
<tr><td><b>Skipped</b></td><td>All task types. The task was skipped by a user.</td></tr>
<tr><td><b>Finished</b></td><td>All task types. The task was forced by the user to finish. The user may do this in cases where the task had "Cancelled" or "Failed" status, and the user needed to release other task instances depending on the successful completion of this task instance in a workflow. For more information, see Force Finishing a Task.</td></tr>
<tr><td><b>Success</b></td><td>All task types. The task has completed successfully.</td></tr>
</table>
</td></tr>
<tr><td>Invoked By</td><td>Task instance only; system-supplied. How the task instance was launched. One of the following:
• Trigger: (Trigger Name) - The instance was launched by the named trigger.
• Workflow: (Workflow Name) - The instance was launched by the named workflow.
• Manually Launched - The instance was launched by a user. To determine the name of the user:
1. From the Activity or All Task Instances screen, click the task instance name to open the record.
2. The Execution User field identifies the user who launched the task instance.</td></tr>
</table>

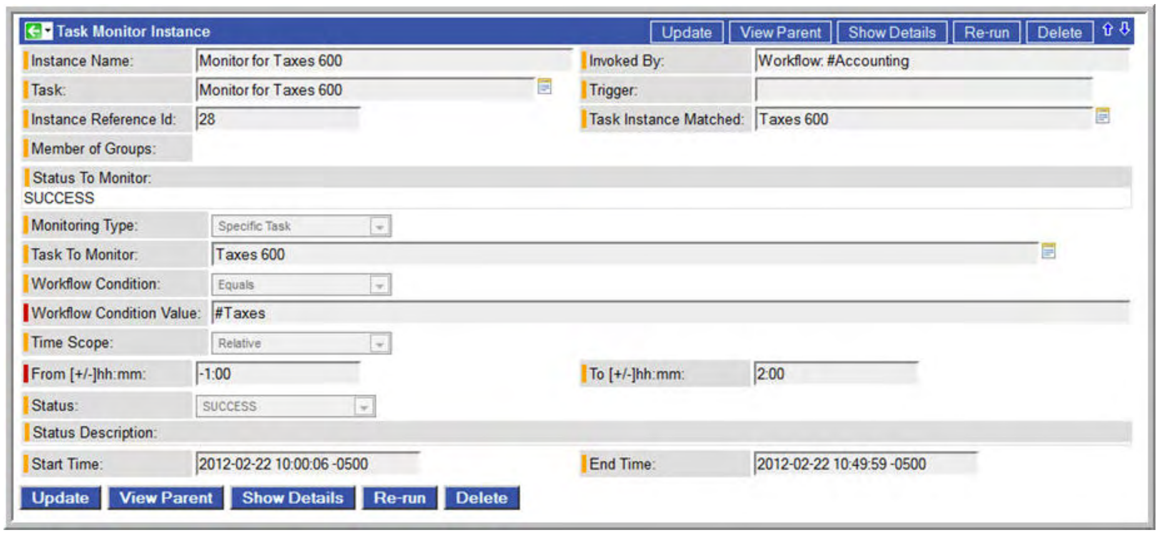
Agent	Required. Name used within Opswise Automation Center to identify this resource. Up to 40 alphanumerics. It is the user's responsibility to develop a workable naming scheme for resources.
Start Time	Task instance only; system-supplied. The date and time the task started.
End Time	Task instance only; system-supplied. The date and time the task instance completed.

Sorting and Filtering

For information about sorting, filtering, and other list options, see Using Lists.

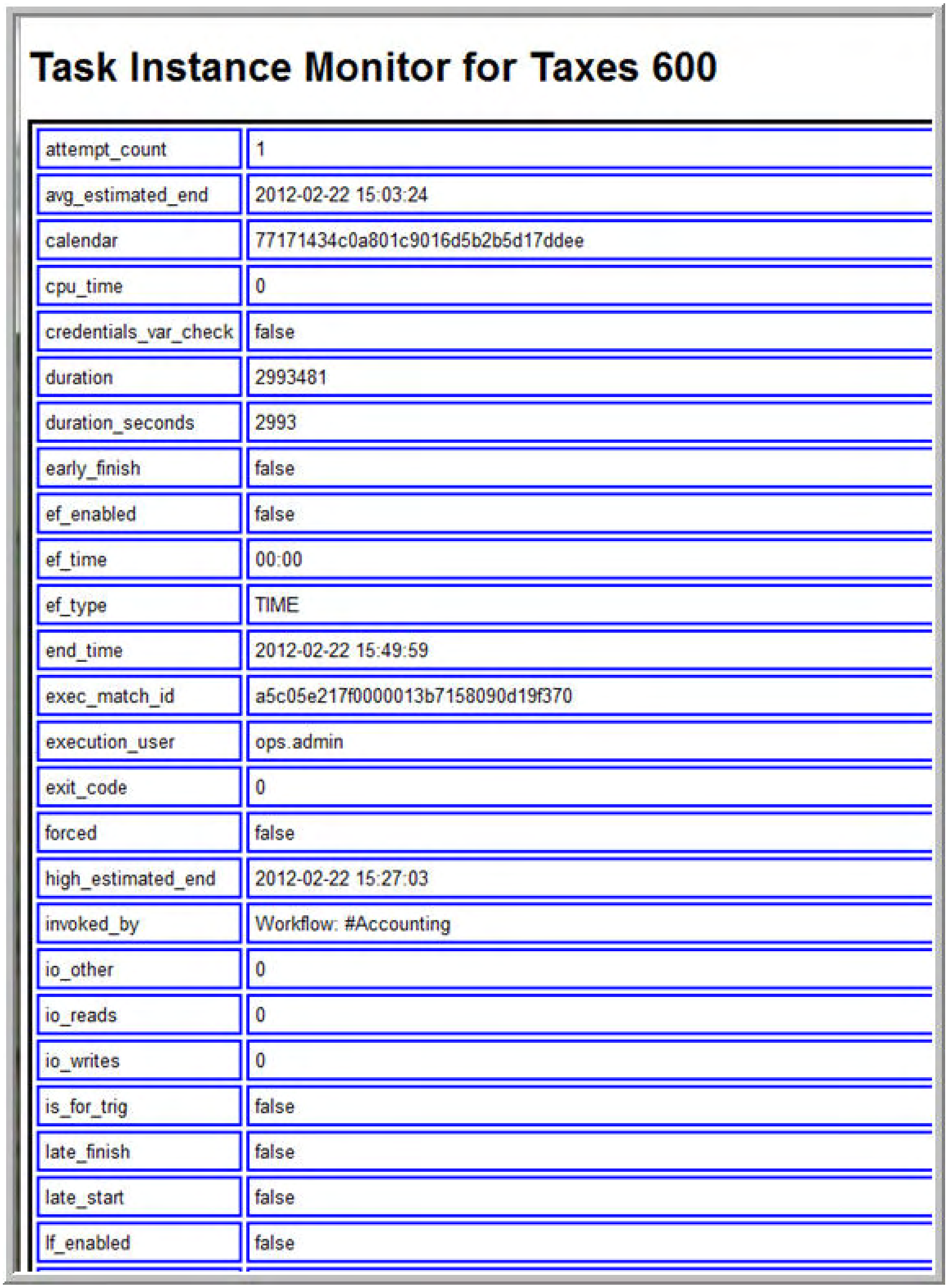
Displaying Details about Task Instances

Click on the underlined detail in the leftmost column (the default is Instance Name). Opswise displays additional details about the instance, as shown in the following example:



Instance Name: Monitor for Taxes 600
Invoked By: Workflow: #Accounting
Task: Monitor for Taxes 600
Trigger:
Instance Reference Id: 28
Task Instance Matched: Taxes 600
Member of Groups:
Status To Monitor:
SUCCESS
Monitoring Type: Specific Task
Task To Monitor: Taxes 600
Workflow Condition: Equals
Workflow Condition Value: #Taxes
Time Scope: Relative
From [+/-]hh:mm: -1:00
To [+/-]hh:mm: 2:00
Status: SUCCESS
Status Description:
Start Time: 2012-02-22 10:00:06 -0500
End Time: 2012-02-22 10:49:59 -0500

To show all details about the instance, click the **Show Details** button. Opswise opens a new browser and displays a table containing all the detailed elements for this task instance, as shown in the following example:



Task Instance Monitor for Taxes 600

attempt_count	1
avg_estimated_end	2012-02-22 15:03:24
calendar	77171434c0a801c9016d5b2b5d17ddee
cpu_time	0
credentials_var_check	false
duration	2993481
duration_seconds	2993
early_finish	false
ef_enabled	false
ef_time	00:00
ef_type	TIME
end_time	2012-02-22 15:49:59
exec_match_id	a5c05e217f0000013b7158090d19f370
execution_user	ops.admin
exit_code	0
forced	false
high_estimated_end	2012-02-22 15:27:03
invoked_by	Workflow: #Accounting
io_other	0
io_reads	0
io_writes	0
is_for_trig	false
late_finish	false
late_start	false
lf_enabled	false

Close the browser to return the the Task Instance details display.

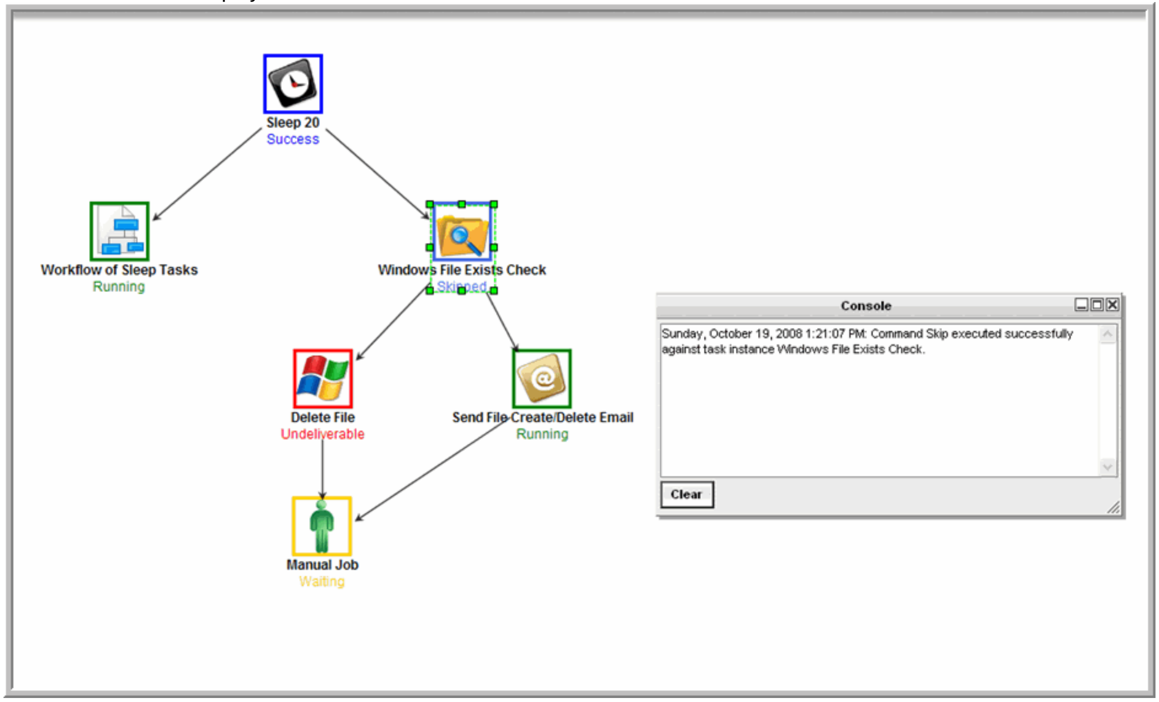
Monitoring Workflows

Overview

Opswise allows you to monitor running workflows in graphical format. As the workflow progresses, the display provides up-to-date textual and color-coded status information for each task. You can also intervene in processing where necessary.

Monitoring a Running Workflow

1. Select **Automation Center > Activity** from the navigation pane.
2. Locate the workflow you want to monitor. A workflow must already be running in order for you to view its status.
3. Click on the underlined Instance Name of the workflow. The workflow monitor opens, as shown in the following example. Click the Console icon to display the console.



Sunday, October 19, 2008 1:21:07 PM: Command Skip executed successfully against task instance Windows File Exists Check.

Status Color Coding

For each task, the display provides its current status with an associated color. You can customize the colors used for each status.

The following table shows the default color for each status, along with the status value you will need if you change the color.

Status	Status Value	Default Color
Defined	0	Grey
Waiting	10	Gold

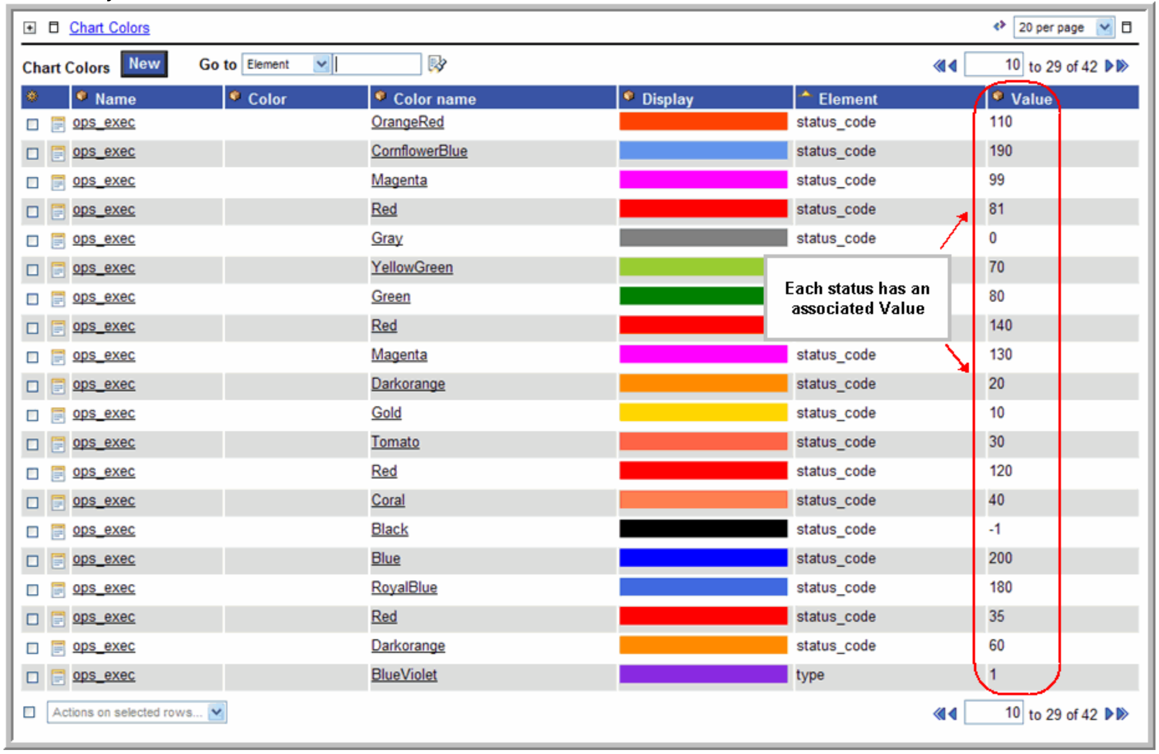
Held	20	Dark Orange
Resource Wait	30	Tomato
Action Required	60	Dark Orange
Queued	40	Coral
Started	70	Yellow-Green
Running	80	Green
Running Problems	81	Red
Cancel Pending	99	Magenta
In Doubt	110	Orange-red
Success	200	Blue
Finished	190	Cornflower Blue
Failed	140	Red
Cancelled	130	Magenta
Start Failure	120	Red
Skipped	180	Royal Blue

Changing a Status Color

Perform the following steps to change a status color.

> ⚠ You may need administrative privileges to perform these steps.

1. Select **Configuration > Chart Colors**.
2. Use the **Go To** field to skip to the Element "status_code". Each entry represents a color specification for a particular status. The status is identified by the number in the Value field. The table above identifies the Status Value for each status.

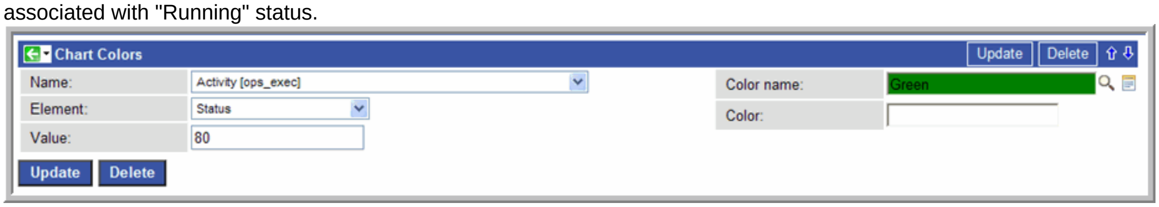


	Name	Color	Color name	Display	Element	Value
□	ops_exec		OrangeRed	■	status_code	110
□	ops_exec		CornflowerBlue	■	status_code	190
□	ops_exec		Magenta	■	status_code	99
□	ops_exec		Red	■	status_code	81
□	ops_exec		Gray	■	status_code	0
□	ops_exec		YellowGreen	■		70
□	ops_exec		Green	■		80
□	ops_exec		Red	■		140
□	ops_exec		Magenta	■	status_code	130
□	ops_exec		Darkorange	■	status_code	20
□	ops_exec		Gold	■	status_code	10
□	ops_exec		Tomato	■	status_code	30
□	ops_exec		Red	■	status_code	120
□	ops_exec		Coral	■	status_code	40
□	ops_exec		Black	■	status_code	-1
□	ops_exec		Blue	■	status_code	200
□	ops_exec		RoyalBlue	■	status_code	180
□	ops_exec		Red	■	status_code	35
□	ops_exec		Darkorange	■	status_code	60
□	ops_exec		BlueViolet	■	type	1

Each status has an associated Value

3. Locate the entry for the status you want to change. For example, to change the color associated with "Running" status, locate the entry with an Element of "status_code" and a Value of 80 (from the table above).

4. Open the record by clicking on the Name (ops_exec) in the leftmost column. The example below shows the entry for value "80", which is associated with "Running" status.



Name: Activity [ops_exec]
Element: Status
Value: 80
Color name: Green
Color:

5. Click on the magnifying glass to select a color or add a new color definition. Or, use the Color field to enter a hex representation for a color.
6. Click **Update**.

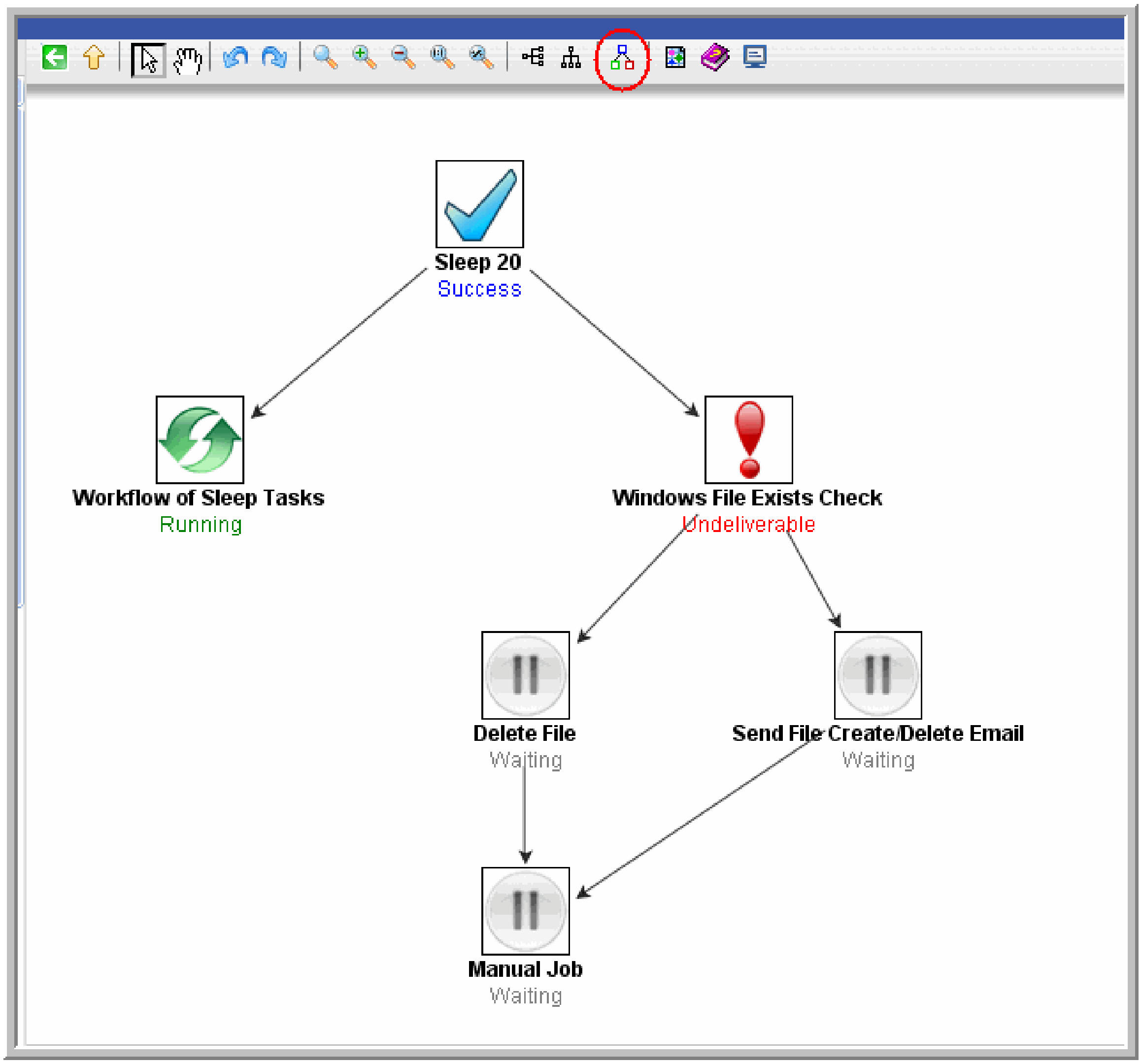
Task Type Icons

Each task type has an icon associated with it, along with the name of the task below it. Shown below are the icons associated with each task type.

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

Toggling the Workflow Display Mode

The default display mode for running workflows is to show icons that represent the task types (described in the above table). If you prefer, you can switch the mode of your running workflow so that the icons represent status instead of task type, as shown in the example below. To switch modes, click the **Toggle Vertex Style** icon .



Each status has an icon associated with it, as described below:

⟳ Running	✓ Complete	⏸ Not Running	! Problem	? Status Unknown

Manually Intervening in a Workflow

You can take action on a task in a running workflow by right-clicking the task and selecting a command from the command list. For more information, see Manually Running and Controlling Tasks. When you issue a command from the command list, the Console window automatically opens and displays processing information.

Displaying Processing Information

To display details about task instance processing while you monitor the workflow, click the Console 🖥 icon.

Displaying Task Status

You can display the status of one or more tasks from the Activity screen or from the command line interface.

To view status from the Activity screen:

Click here to view an image of the Activity screen, which displays task status.

To view status from the command line interface:

Click here for the command format.

Status	Description
Defined	All task types. The new task instance has been created (the task has been launched). Not yet implemented.
Waiting	All task types. The task has been loaded by a workflow and is waiting to run.
Held	All task types. The task has been put on hold by a user.
Resource Requested	All tasks with a virtual resource defined go immediately to a status of Resource Requested. If the resource is available, the task then moves to the next appropriate processing status.
Resource Wait	All tasks with a virtual resource defined go immediately to a status of Resource Requested. If the resource is not available, the task goes to a status of Resource Wait. When the resource becomes available, the task moves to the next appropriate processing status
Undeliverable	Agent-based tasks. The agent is unavailable.
Queued	Agent-based tasks only. The task has been queued on a resource.
Submitted	z/OS only. The task has been submitted to the z/OS Job Entry subsystem and scheduled by the z/OS Job Scheduler.
Action Required	Manual tasks only. When a manual task launches, it goes into Action Required status, meaning a user must perform some manual activity. For details, see Manual task.
Started	Agent-based and Manual tasks only. The task has started. For agent-based tasks, this means the agent has received the task.
Running	All task types. The task is running. For agent-based tasks, the agent has started running the program.
Running Problems	Workflows only. One or more tasks within the workflow has one of the following statuses: • Held • Undeliverable • Running Problems (for sub-workflows) • Cancel Pending • In Doubt • Start Failure • Cancelled
In Doubt	Agent-based tasks only. The agent is "in doubt" about the current status of the task instance. This may occur if an agent or agent connection goes down. In this case, the agent restarts and reviews its data about tasks in progress. If the agent finds a task still running, it resumes normal monitoring. If the agent cannot find the task, this usually indicates that the task completed, but the agent considers the task status to be "in doubt."
Start Failure	All task types. The task was unable to start.
Confirmation Required	z/OS only. If you make JCL changes and restart a z/OS task, Opswise will put the task into Confirmation Required status and prompt you for a confirmation. For detailed processing steps, see Rerunning a z/OS Task.
Cancelled	All task types. The task was cancelled by a user.
Failed	All task types. The task ran to a failure status.
Skipped	All task types. The task was skipped by a user.
Finished	All task types. The task was forced by the user to finish. The user may do this in cases where the task had "Cancelled" or "Failed" status, and the user needed to release other task instances depending on the successful completion of this task instance in a workflow. For more information, see Force Finishing a Task.
Success	All task types. The task has completed successfully.

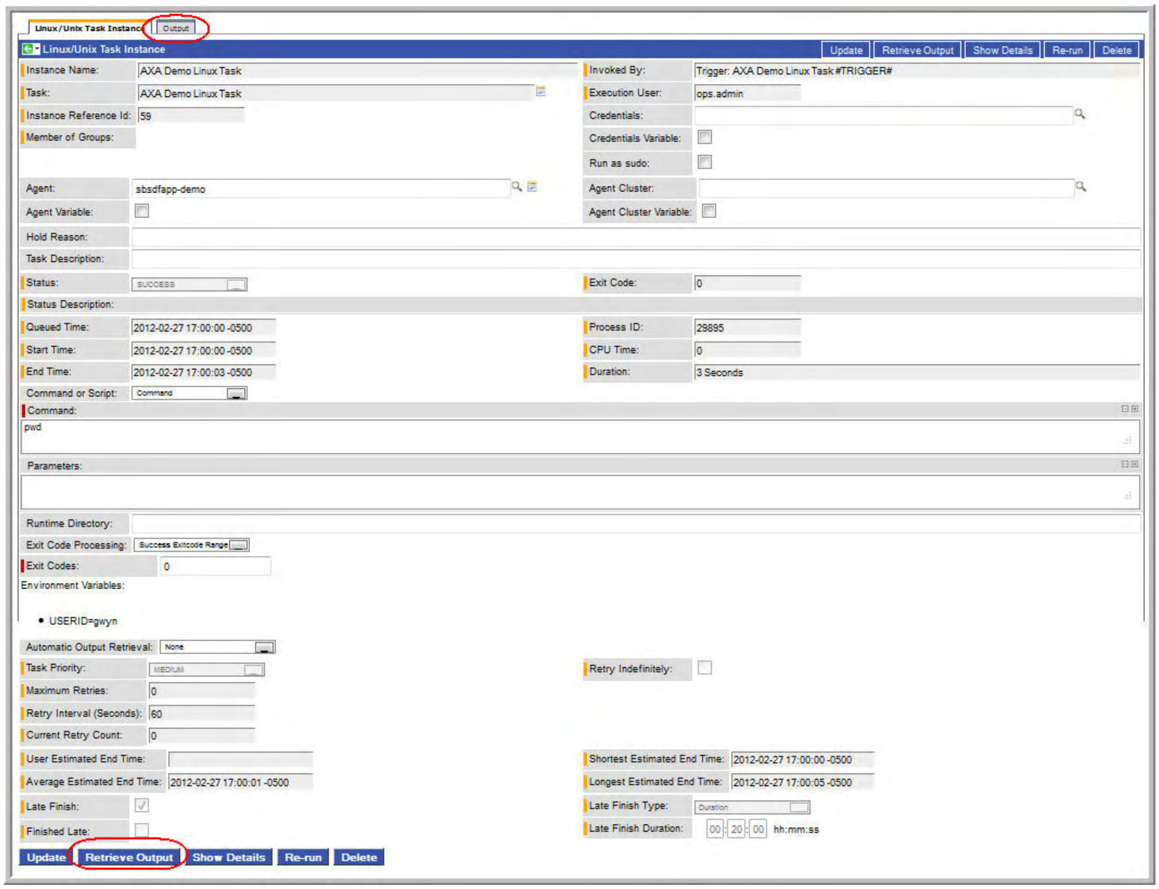
Retrieving Output

Overview

For any completed task where output has been generated and you did not specify that output be automatically retrieved, you can instruct the system to retrieve the output.

Procedure

1. From the navigation pane, select **Automation Center > Task Instances > Activity**. The Activity screen displays.
2. Click the task whose output you want to retrieve. The system displays the task instance record, as shown in the following example of a Linux/Unix task instance.



Instance Name: AXA Demo Linux Task
Invoked By: Trigger: AXA Demo Linux Task #TRIGGER#
Task: AXA Demo Linux Task
Execution User: ops.admin
Instance Reference Id: 59
Credentials:
Member of Groups:
Credentials Variable: □
Run as sudo: □
Agent: sbsdfapp-demo
Agent Cluster:
Agent Variable: □
Agent Cluster Variable: □
Hold Reason:
Task Description:
Status: SUCCESS
Exit Code: 0
Status Description:
Queued Time: 2012-02-27 17:00:00 -0500
Process ID: 29895
Start Time: 2012-02-27 17:00:00 -0500
CPU Time: 0
End Time: 2012-02-27 17:00:03 -0500
Duration: 3 Seconds
Command or Script: Command
Command:

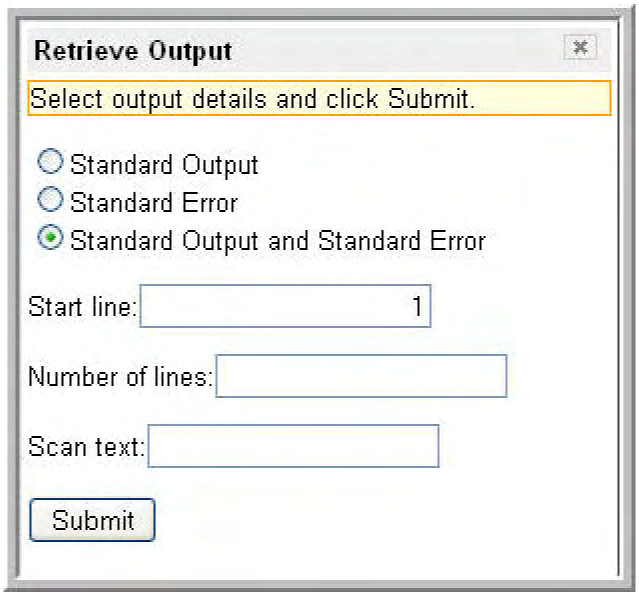
```
pwd
```

Parameters:
Runtime Directory:
Exit Code Processing: Success Exitcode Range
Exit Codes: 0
Environment Variables:

- USERID=gwyn

Automatic Output Retrieval: None
Task Priority: MEDIUM
Retry Indefinitely: □
Maximum Retries: 0
Retry Interval (Seconds): 60
Current Retry Count: 0
User Estimated End Time:
Shortest Estimated End Time: 2012-02-27 17:00:00 -0500
Average Estimated End Time: 2012-02-27 17:00:01 -0500
Longest Estimated End Time: 2012-02-27 17:00:05 -0500
Late Finish: ☑
Late Finish Type: Duration
Finished Late: □
Late Finish Duration: 00 : 20 : 00 hh:mm:ss

3. Click the **Retrieve Output** button. If output is available, the Retrieve Output window displays.



Retrieve Output

Select output details and click Submit.

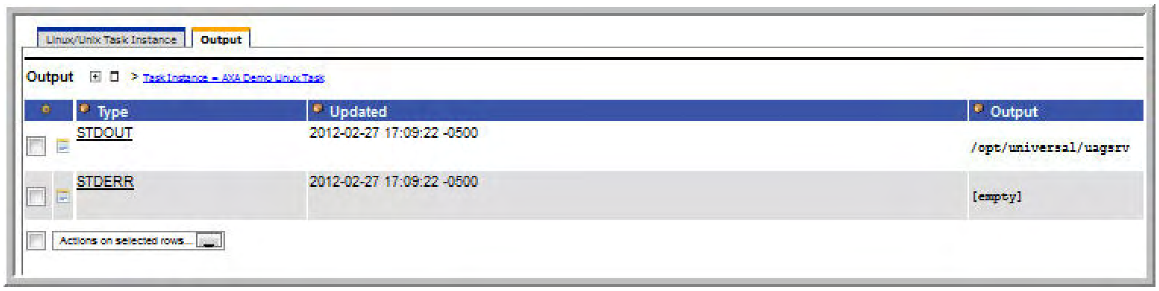
○ Standard Output
○ Standard Error
● Standard Output and Standard Error

Start line: 1

Number of lines:

Scan text:

4. Using the field descriptions provided below, make your selection. Positioning can be accomplished by using Start Line (use -1 to retrieve the last "Number of Lines") or "Scan Text" to see the "Number of lines" both before and after the first match of text.
5. Click the **Submit** button. The system retrieves the output you specified and writes the record to the Output tab, shown in the following example.

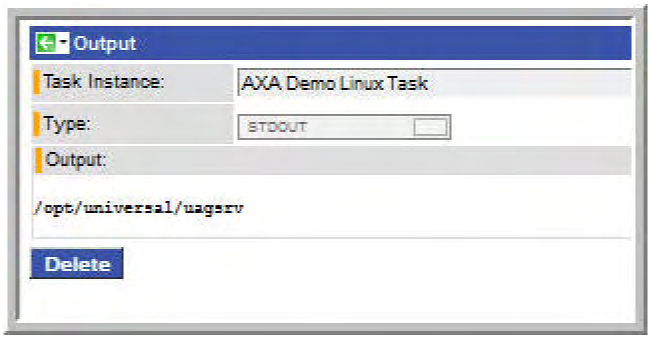


Linux/Unix Task Instance
Output
Output
> Task Instance = AXA Demo Linux Task

	Type	Updated	Output
□	STDOUT	2012-02-27 17:09:22 -0500	/opt/universal/uagsrv
□	STDERR	2012-02-27 17:09:22 -0500	[empty]

□ Actions on selected rows...

6. To display the output, open the Output tab and click on the record. The output record displays, as shown in the following example.



Output

Task Instance: AXA Demo Linux Task

Type: STDOUT

Output:

```
/opt/universal/uagsrv
```

Field Descriptions

Column Name	Description
Standard Output	Retrieve standard output returned by the program.
Standard Error	Retrieve standard error information returned by the program.
Standard Output and Standard Error	Retrieve both standard output and standard error information.
Start Line	Retrieve data beginning at the line indicated.

Number of Lines	Limit the retrieved data to the number of lines specified. The default is 100. You can change the default in the record **Configuration > Properties > Retrieve Output Default Maximum Lines**.
Scan text:	Scan the data for the text specified and retrieve only that.

Monitoring the Activity History Report

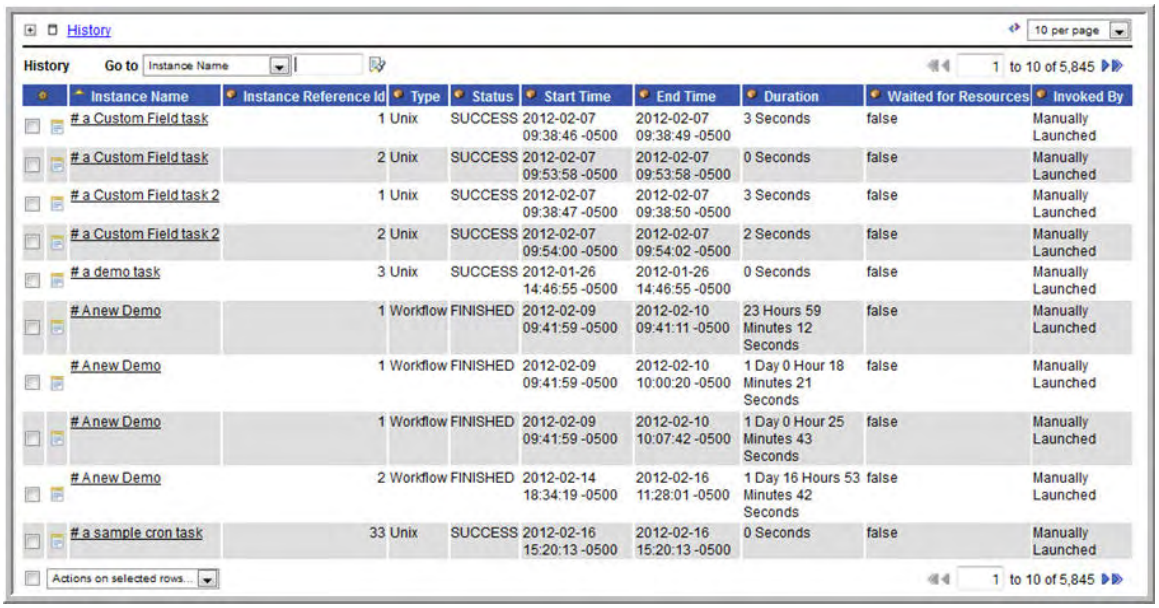
Overview

The Activity History report provides an historical display of all completed task activity. Only tasks with a status in an "end state" (SUCCESS, FINISHED, FAILED, CANCELLED, START FAILURE, SKIPPED) display in the Activity History report. This allows you to track information about a specific task or tasks, including multiple runs. For example, Task A may have failed and was then re-run by a user. This task will display twice in the Activity History report, first for the time that it ran and failed and again for the time it was re-run to success.

See Using Lists for information about sorting, filtering, and other record list features.

Displaying the Activity History Report

1. From the navigation pane, select **Automation Center > Task Instances > History**. Opswise displays the Activity History report.

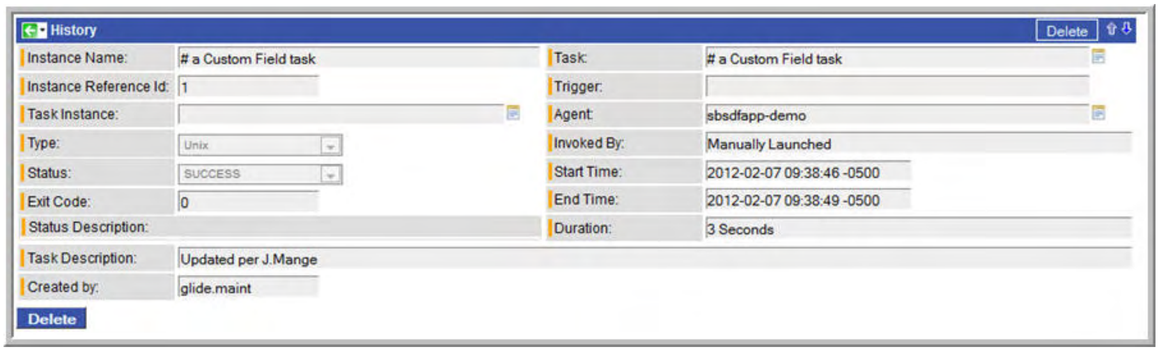


	Instance Name	Instance Reference Id	Type	Status	Start Time	End Time	Duration	Waited for Resources	Invoked By
□	# a Custom Field task	1	Unix	SUCCESS	2012-02-07 09:38:46 -0500	2012-02-07 09:38:49 -0500	3 Seconds	false	Manually Launched
□	# a Custom Field task	2	Unix	SUCCESS	2012-02-07 09:53:58 -0500	2012-02-07 09:53:58 -0500	0 Seconds	false	Manually Launched
□	# a Custom Field task 2	1	Unix	SUCCESS	2012-02-07 09:38:47 -0500	2012-02-07 09:38:50 -0500	3 Seconds	false	Manually Launched
□	# a Custom Field task 2	2	Unix	SUCCESS	2012-02-07 09:54:00 -0500	2012-02-07 09:54:02 -0500	2 Seconds	false	Manually Launched
□	# a demo task	3	Unix	SUCCESS	2012-01-26 14:46:55 -0500	2012-01-26 14:46:55 -0500	0 Seconds	false	Manually Launched
□	# Anew Demo	1	Workflow	FINISHED	2012-02-09 09:41:59 -0500	2012-02-10 09:41:11 -0500	23 Hours 59 Minutes 12 Seconds	false	Manually Launched
□	# Anew Demo	1	Workflow	FINISHED	2012-02-09 09:41:59 -0500	2012-02-10 10:00:20 -0500	1 Day 0 Hour 18 Minutes 21 Seconds	false	Manually Launched
□	# Anew Demo	1	Workflow	FINISHED	2012-02-09 09:41:59 -0500	2012-02-10 10:07:42 -0500	1 Day 0 Hour 25 Minutes 43 Seconds	false	Manually Launched
□	# Anew Demo	2	Workflow	FINISHED	2012-02-14 18:34:19 -0500	2012-02-16 11:28:01 -0500	1 Day 16 Hours 53 Minutes 42 Seconds	false	Manually Launched
□	# a sample cron task	33	Unix	SUCCESS	2012-02-16 15:20:13 -0500	2012-02-16 15:20:13 -0500	0 Seconds	false	Manually Launched

2. You can modify the display by filtering, sorting, adding, and removing columns, and so on. For more information, see Using Lists.

Displaying Details from the History Report

To display execution details about a task instance in the Activity History report, click the underlined identifier in the leftmost column. Opswise displays the execution details.



Instance Name:	# a Custom Field task	Task:	# a Custom Field task
Instance Reference Id:	1	Trigger:	
Task Instance:		Agent:	sbsdfapp-demo
Type:	Unix	Invoked By:	Manually Launched
Status:	SUCCESS	Start Time:	2012-02-07 09:38:46 -0500
Exit Code:	0	End Time:	2012-02-07 09:38:49 -0500
Status Description:		Duration:	3 Seconds
Task Description:	Updated per J.Mange		
Created by:	glide.maint		

Manually Running and Controlling Tasks

Overview

A number of options are available on the Activity screen that allow you to intervene in task processing where needed. Some commands are applicable only to certain task types and others are appropriate only when the task is in a particular status. In addition, commands require appropriate permissions.

Selecting One or More Task Instances

You can issue commands against a single task instance by right-clicking and selecting the option from the menu.

You also can select multiple task instances from the Activity screen:

- Press **CTRL/left-click** on each task instance. On the Activity screen, you can click in any column; make sure you don't click directly on the Name, which is a hyperlink and opens the record in a new tab.
- When you have selected all the records you want, right-click to display the menu. The menu displays only those options that apply to all selected task instances.

Supported Commands Listed by Task Status

The following table provides a quick reference of task statuses and the commands that you can issue against a task in each status. For details and instructions on each command type, see the specific section.

Status	Supported Commands

Defined	• Clear Dependencies • Force Finish • Hold • Skip • Recursive Release - Workflow tasks only.
Waiting	• Clear Dependencies • Force Finish • Hold • Skip • Recursive Release - Workflow tasks only.
Held	• Clear Dependencies • Force Finish • Release • Skip • Recursive Release - Workflow tasks only.
Resource Requested (tasks using Virtual Resources only)	• Force Finish • Hold • Skip
Resource Wait (tasks using Virtual Resources only)	• Force Finish • Hold • Skip • Recursive Release - Workflow tasks only.
Undeliverable (agent-based tasks only)	• Force Finish • Hold • Skip
Queued (agent-based tasks only)	• Cancel • Force Finish • Hold • Skip
Step Restarted (z/OS tasks only)	
Action Required (Manual tasks only)	• Cancel • Force Finish • Set Started • Set Completed
Started (agent-based and Manual tasks only)	• Cancel • Force Finish • Set Completed - Manual tasks only.
Running	• Cancel • Force Finish • Recursive Release - Workflow tasks only.

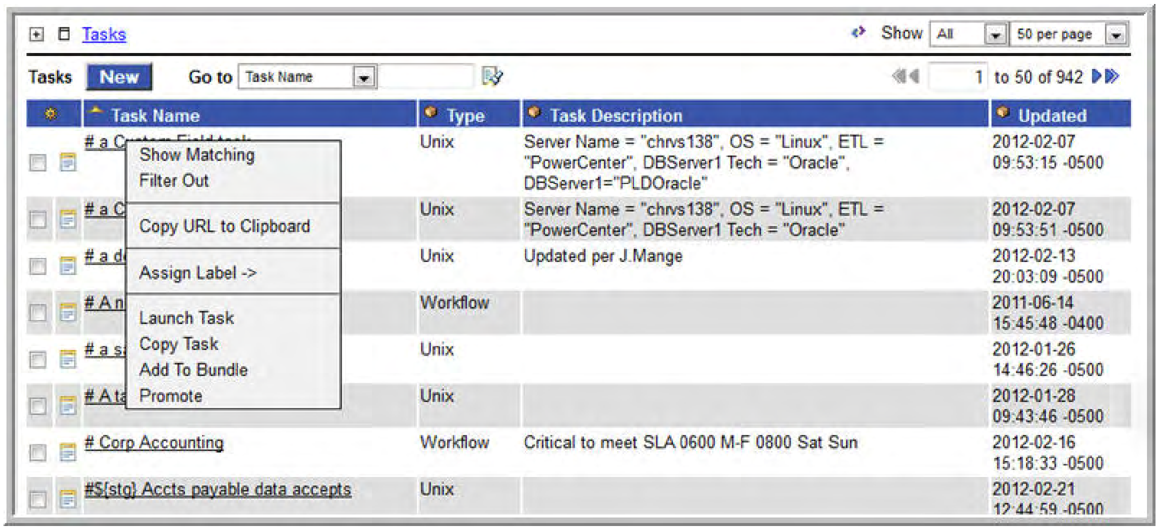
Running Problems (Workflows tasks only)	- Cancel - Force Finish - Hold - Release Recursive - Workflow tasks only.
Cancel Pending	- Force Finish
In Doubt (agent-based tasks only)	- Force Finish
Start Failure	- Force Finish - Re-run - Not applicable for Workflow.
Confirmation Required (z/OS only)	
Cancelled	- Force Finish - Re-run - Not applicable for Workflow.
Failed	- Force Finish - Re-run - Not applicable for workflow.
Skipped	(None)
Finished	- Re-run - Not applicable for workflow.
Success	- Re-run - Not applicable for workflow.

Manually Launching a Task

You can manually launch one or more tasks from the task list or display a task and launch it from there.

To launch one or more tasks from the task list

1. Select **Automation Center > Tasks** and follow one of the procedures below:
   To launch a single task:
   a. Right-click on the task you want to run. The right-click menu displays.

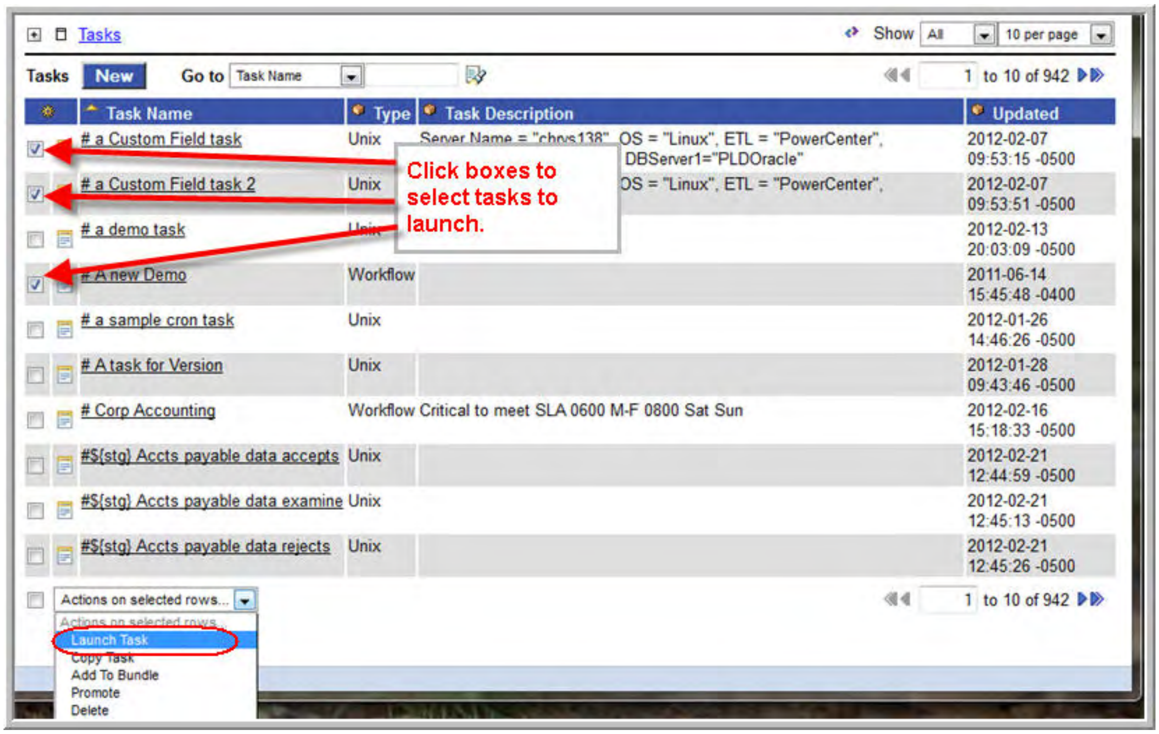


Task Name	Type	Task Description	Updated
# a C	Unix	Server Name = "chrvs138", OS = "Linux", ETL = "PowerCenter", DBServer1 Tech = "Oracle", DBServer1="PLDOracle"	2012-02-07 09:53:15 -0500
# a C	Unix	Server Name = "chrvs138", OS = "Linux", ETL = "PowerCenter", DBServer1 Tech = "Oracle"	2012-02-07 09:53:51 -0500
# a d	Unix	Updated per J.Mange	2012-02-13 20:03:09 -0500
# A n	Workflow		2011-06-14 15:45:48 -0400
# a s	Unix		2012-01-26 14:46:26 -0500
# A ta	Unix		2012-01-28 09:43:46 -0500
# Corp Accounting	Workflow	Critical to meet SLA 0600 M-F 0800 Sat Sun	2012-02-16 15:18:33 -0500
#${stg} Accts payable data accepts	Unix		2012-02-21 12:44:59 -0500

b. Select **Launch Task.** The system creates an instance of the task and runs it.

To launch one or more tasks:

a. For each task you want to launch, click the box to the left of the task.
b. From the **Actions on selected rows...** menu at the bottom of the list, select **Launch Task.**

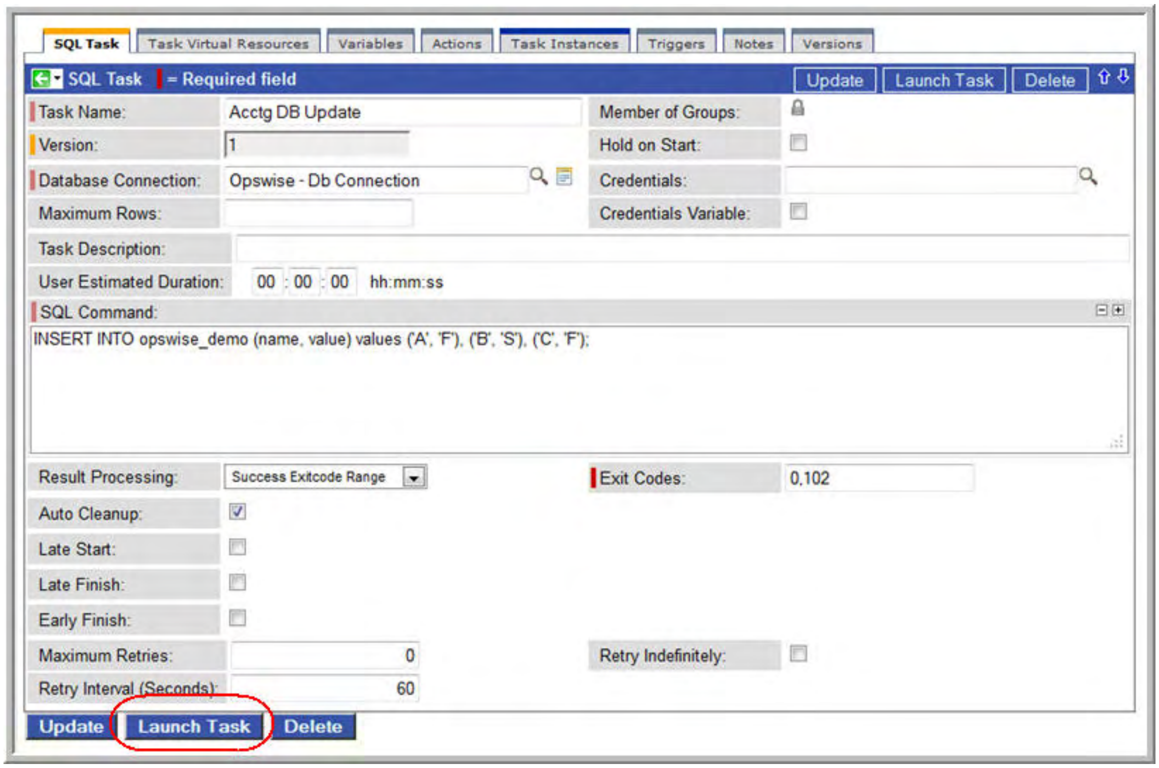

Tasks
Show All
10 per page
Tasks New Go to Task Name
1 to 10 of 942
Click boxes to select tasks to launch.
Actions on selected rows...
Actions on selected rows...
Launch Task
Copy Task
Add To Bundle
Promote
Delete
1 to 10 of 942

	Task Name	Type	Task Description	Updated
☑	# a Custom Field task	Unix	Server Name = "chrvs138" OS = "Linux", ETL = "PowerCenter", DBServer1="PLDOracle"	2012-02-07 09:53:15 -0500
☑	# a Custom Field task 2	Unix	OS = "Linux", ETL = "PowerCenter",	2012-02-07 09:53:51 -0500
□	# a demo task	Unix		2012-02-13 20:03:09 -0500
☑	# A new Demo	Workflow		2011-06-14 15:45:48 -0400
□	# a sample cron task	Unix		2012-01-26 14:46:26 -0500
□	# A task for Version	Unix		2012-01-28 09:43:46 -0500
□	# Corp Accounting	Workflow	Critical to meet SLA 0600 M-F 0800 Sat Sun	2012-02-16 15:18:33 -0500
□	#${stg} Accts payable data accepts	Unix		2012-02-21 12:44:59 -0500
□	#${stg} Accts payable data examine	Unix		2012-02-21 12:45:13 -0500
□	#${stg} Accts payable data rejects	Unix		2012-02-21 12:45:26 -0500

c. To view details about running task instances, select **Automation Center > Activity**and click on the task instance.

To launch a task from the task screen

1. Display the task you want to launch.
2. Click the **Launch Task** button, as shown in the example below.



Task Name: Acctg DB Update

Version: 1

Database Connection: Opswise - Db Connection

Maximum Rows:

Member of Groups:

Hold on Start: □

Credentials:

Credentials Variable: □

Task Description:

User Estimated Duration: 00 : 00 : 00 hh:mm:ss

SQL Command:

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

Result Processing: Success Exitcode Range

Exit Codes: 0,102

Auto Cleanup: ☑

Late Start: □

Late Finish: □

Early Finish: □

Maximum Retries: 0

Retry Indefinitely: □

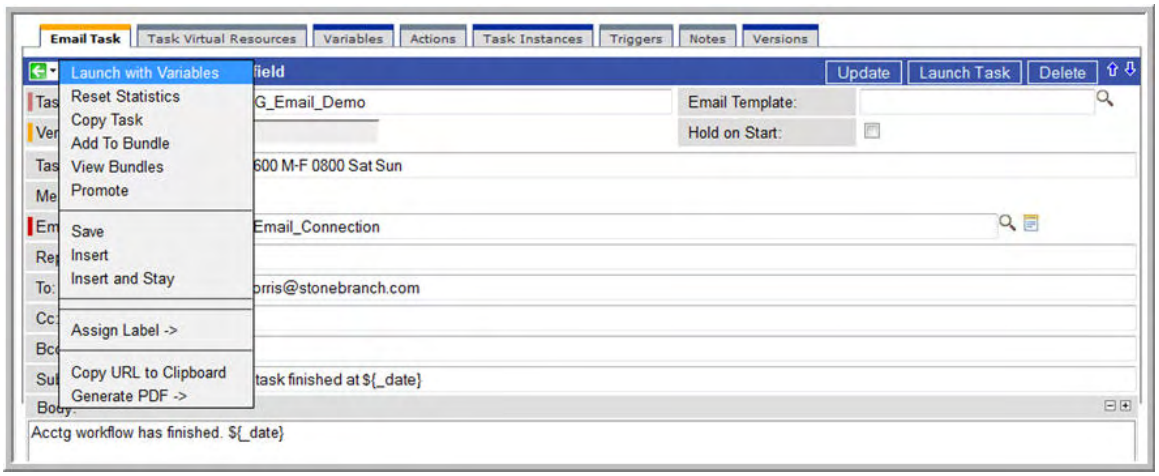
Retry Interval (Seconds): 60

To provide temporary variable values and launch a task manually

Opswise supports a "Launch with Variables" feature that allows you to quickly provide values for the variables specified in the task and launch it. All task types support the "Launch with Variables" feature.

To launch a task using "Launch with Variables:"

1. Display the task you want to launch. This procedure assumes the task is already set up with variables where required.
2. Open the task menu by clicking the down arrow or right-clicking in the title bar. The following menu appears.



G_Email_Demo

Email Template:

Hold on Start: □

600 M-F 0800 Sat Sun

Email_Connection

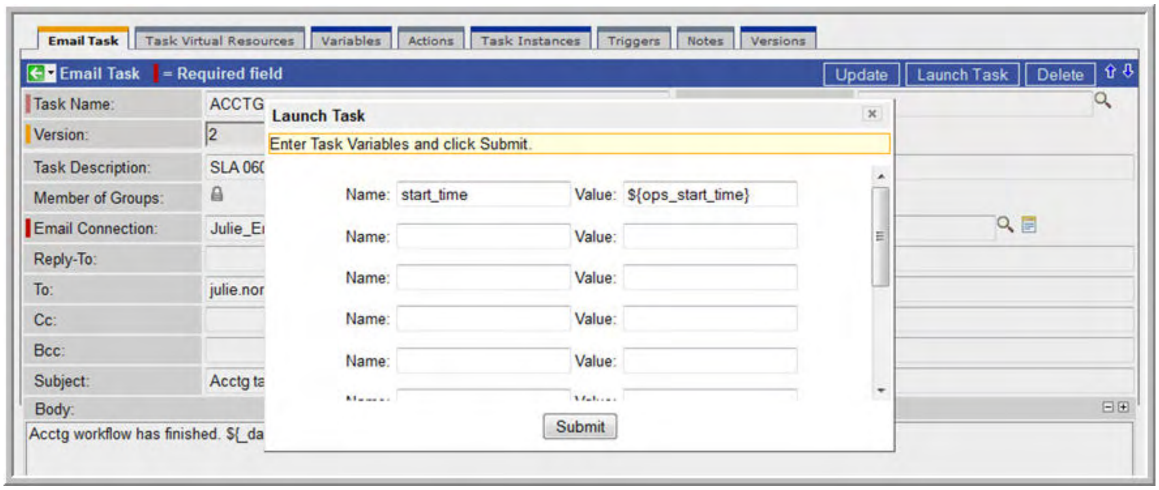
orris@stonebranch.com

task finished at ${_date}

Body:

Acctg workflow has finished. ${_date}

3. Select **Launch with Variables**. A pop-up window appears. Any variables attached to this task are displayed in the window.



Task Name: ACCTG

Version: 2

Task Description: SLA 060

Member of Groups:

Email Connection: Julie_E

Reply-To:

To: julie.nor

Cc:

Bcc:

Subject: Acctg ta

Body:

Acctg workflow has finished. ${_da

Launch Task

Enter Task Variables and click Submit.

Name: start_time Value: ${ops_start_time}

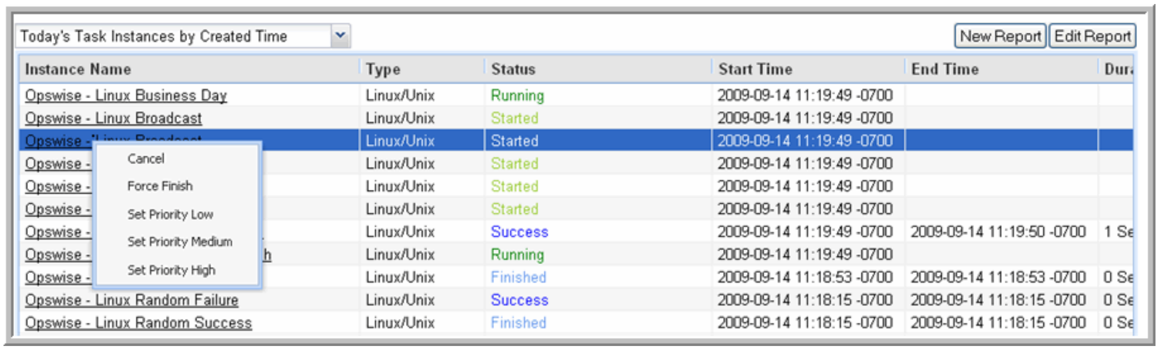
4. As needed, set the variable values or add new variables. The window allows you to specify up to six variable and value pairs.
5. When you are finished, click **Submit**. The system populates the variables with the values you supplied and launches the task.

Changing the Priority of a Task Instance

For Windows, Linux/Unix, or z/OS tasks in a status of Started, you can change the priority so that they will run sooner or later, as described below. The priority specified here is meaningful only in relation to the priority setting of other tasks sent to that agent from the same Opswise instance.

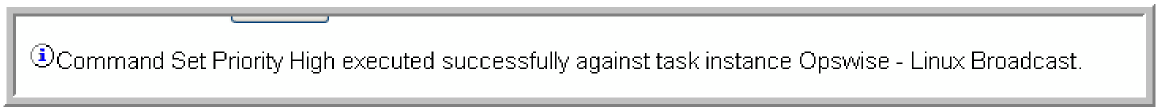
To set priority on a task instance from the Activity or All Task Instances display

1. Select the task instance(s) for which you want to set priority. All task instances must be in Started status.
2. Select the priority level you want this task to have.



Instance Name	Type	Status	Start Time	End Time	Dur
Opswise - Linux Business Day	Linux/Unix	Running	2009-09-14 11:19:49 -0700		
Opswise - Linux Broadcast	Linux/Unix	Started	2009-09-14 11:19:49 -0700		
Opswise -	Linux/Unix	Started	2009-09-14 11:19:49 -0700		
Opswise -	Linux/Unix	Started	2009-09-14 11:19:49 -0700		
Opswise -	Linux/Unix	Started	2009-09-14 11:19:49 -0700		
Opswise -	Linux/Unix	Started	2009-09-14 11:19:49 -0700		
Opswise -	Linux/Unix	Success	2009-09-14 11:19:49 -0700	2009-09-14 11:19:50 -0700	1 Se
Opswise -	Linux/Unix	Running	2009-09-14 11:19:49 -0700		
Opswise -	Linux/Unix	Finished	2009-09-14 11:18:53 -0700	2009-09-14 11:18:53 -0700	0 Se
Opswise - Linux Random Failure	Linux/Unix	Success	2009-09-14 11:18:15 -0700	2009-09-14 11:18:15 -0700	0 Se
Opswise - Linux Random Success	Linux/Unix	Finished	2009-09-14 11:18:15 -0700	2009-09-14 11:18:15 -0700	0 Se

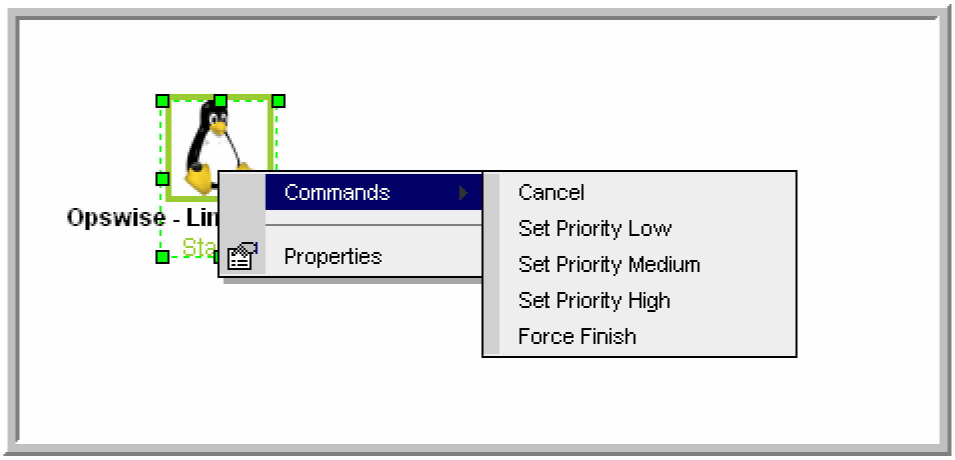
3. Once the set priority command has been executed, the system displays the following message at the bottom of the screen.



Command Set Priority High executed successfully against task instance Opswise - Linux Broadcast.

To set priority on a task instance from the workflow monitor

1. View the workflow that contains the task instance for which you want to set priority.
2. Select the task instance(s) for which you want to set priority. The task(s) must be in Started status.
3. Select **Commands**.



4. Select the priority you want to give it.

Re-running a Task

You can re-run a task while it is in any of the following statuses: Success, Start Failure, Failed, Cancelled, Finished. If a task instance is part of a workflow, you can only re-run it as long as the workflow has not completed. If the task instance is not part of a workflow, you can re-run it as long as it has not been manually purged from the Activity display. You cannot re-run a workflow task.

When you re-run a task, the system uses the same task instance. That is, the new task instance has the same sys_id. However, you can view the two task instances distinctly in the Activity History (one for each time it ran).

You can re-run a task instance from the Activity or All Task Instances display. If the task instance is running as part of a workflow, you can also re-run it from the workflow monitor.

To re-run a task instance from the Activity or All Task Instances display

1. Select the task instance(s) you want to re-run.
2. Click **Re-run**. The task status changes to the next appropriate status as though it had just been launched.

To re-run a task instance from the task instance screen

1. From the Activity screen, display the task instance you want to re-run.
2. Click the **Re-run** button. The task status changes to the next appropriate status as though it had just been launched.

To re-run a task instance from the workflow monitor

1. View the workflow that contains the task instance you want to re-run.
2. Select the task instance(s) you want to re-run.
3. Select **Commands**.
4. Select **Re-run**. The task status changes to the next appropriate status as though it had just been launched, and the Console opens to display information about the re-run.

Cancelling a Task Run

The Cancel command cancels a running task. For tasks that run on agents, including Windows, Linux, Unix, z/OS, FTP, File Monitor, and Indesca tasks, the Cancel command is sent to the agent. If the task has not been launched yet, it does not launch. If the task has already been launched, the agent cancels it, if possible. If the task is a workflow, any tasks in the workflow go to Cancelled status; the workflow goes to Running/Problems status.

You can cancel a task instance while it is in any of the following statuses: Queued, Action Required, Started, Running.

You can cancel a task instance from the Activity or All Task Instances display. If the task instance is running as part of a workflow, you can also cancel it from the workflow monitor.

To cancel a task instance from the Activity or All Task Instances display

1. Select the task instance(s) you want to cancel.
2. Click **Cancel**. The task status changes to Cancelled.

To cancel a task instance from the workflow monitor

1. Open the workflow editor for the workflow that contains the task instance you want to cancel.
2. Select the task instance(s).
3. Select **Commands**.
4. Select **Cancel**. The task status changes to Cancelled and the Console opens to display information about the cancellation.

Force Finishing a Task

A Force Finish puts a task into Finished status, regardless of what the task is doing.

One purpose of Force Finish is to allow downstream tasks in a workflow to launch without waiting for the current task to complete. You also may want to force finish a stand-alone task; for example, you may want to mark a failed job as Finished, rather than rerunning the job.

If a task is running when the user issues a Force Finish, the system marks the task instance as Finished even though the actual process continues running. Two exceptions are the File Monitor and FTP File Monitor; for these task types, the monitoring processes are aborted by a Force Finish command. Assuming they have no other dependencies, all downstream task instances waiting for successful completion of this task instance will start.

When you issue a Force Finish against a workflow, only the tasks in the workflow go to Finished status. The workflow will then go to Success.

You can force finish a task while it is in any of the following statuses: Defined, Waiting, Held, Resource Wait, Queued, Action Required, Started, Running, Cancel Pending, In Doubt, Failure to Start, Cancelled, Failed.

You can force finish a task instance from the Activity or All Task Instances display. If the task instance is running as part of a workflow, you can also force finish it from the workflow monitor.

To force finish a task instance from the Activity or All Task Instances display

1. Select the task instance(s) you want to force finish.
2. Click **Force Finish**. The task status changes to Finished.

To force finish a task instance from the workflow monitor

1. View the workflow that contains the task instance you want to force finish.
2. Select the task instance(s).
3. Select **Commands**.
4. Select **Force Finish**. The task status changes to Finished and the Console opens to display information about the force finish.

Putting a Task on Hold

You can put a task on hold while it is in any of the following statuses: Defined, Waiting, Resource Wait, Queued.

If you put a workflow on hold that has not yet started, the workflow and all the tasks in it are put on hold. If you put a workflow on hold when it is in running status, all the tasks within the workflow that have not yet started are put on hold; however, the workflow itself does not go to Hold status because it has already started.

To release the workflow and all its tasks that are on hold, issue the Recursive Release command against the workflow. To release the workflow but keep the tasks on hold until you release them one by one, use Release on the workflow first, then use Release on each task.

To hold a task instance from the Activity or All Task Instances display

1. Select the task instance(s) you want to put on hold.
2. Click **Hold**. The task status changes to Held.

To hold a task instance from the workflow monitor

1. View the workflow that contains the task instance you want to put on hold.
2. Select the task instance(s).
3. Select **Commands**.
4. Select **Hold**. The task status changes to Held and the Console opens to display information about the hold.

Releasing a Task from Hold

You can release a non-workflow task from hold from the Activity or All Task Instances display while it is in Held status.

For workflows, if the user held a workflow that was already running, only the tasks within the workflow that had not started yet are put into Held status. In this case, the workflow itself does not go to Held status. To release the workflow, use one of the following methods:

- To release the entire held workflow and its tasks, use **Release Recursive**.
- To release a workflow that is not in held status but has tasks that are in Held status, use **Release Recursive**. In this case, you can issue a Release Recursive on a workflow in any of the following statuses: Defined, Waiting, Held, Resource Wait, Running.
- To release the workflow but keep the tasks inside on hold so that you can release them one by one, use **Release**. In this case, release the workflow first, then release each child task instance manually.

To release a held task instance from the Activity or All Task Instances display

1. Select the task instance(s) you want to release from hold.
2. Click **Release** or **Release Recursive**. The task status changes to the next appropriate status according to where it was in processing at the time it was put on hold.

To release a held task instance from the workflow monitor

1. View the workflow that contains the task instance you want to release.
2. Select the task instance(s).
3. Select **Commands**.
4. Select **Release**. The task status changes to the next appropriate status according to where it was in processing, and the Console opens to display information about the release.

Skipping a Task

Two methods are available for specifying that you want the system to skip a task:

1. Instruct the system to skip a task instance in the Activity display or workflow monitor.
2. Modify a trigger to instruct the system to skip launching the next N number of task instances. See the Skip Count field in Creating Triggers.

A task or group of tasks may also go into Skipped status when you specify conditional paths in a workflow.

You can issue a Skip command on a task as long as it has not yet started running. This includes the following statuses: Defined, Waiting, Held, Resource Requested, Resource Wait, Queued.

If you skip a workflow, all the tasks within the workflow are also skipped, along with any nested workflows.

Once a task has been skipped, you cannot re-run it or issue any other commands against it.

You can skip a task instance from the Activity or All Task Instances display. If the task instance is running as part of a workflow, you can also skip it from the workflow monitor.

To skip a task instance from the Activity or All Task Instances display

1. Select the task instance(s) you want to skip.
2. Click **Skip**. The task status changes to Skipped.

To skip a task instance from the workflow monitor

1. View the workflow that contains the task instance you want to skip.
2. Select the task instance(s).
3. Select **Commands**.

4. Select **Skip**. The task status changes to Skipped and the Console opens to display information about the skip.

Clearing All Dependencies for a Task Instance

For tasks running inside of a workflow, you can clear all upstream dependencies to allow the task to run. Clearing a dependency has the same result as satisfying a dependency. You can clear dependencies on tasks in the following status: Defined, Waiting, Held.

To clear dependencies on a task instance from the Activity or All Task Instances display

1. Select the task instance(s) whose dependencies you want to satisfy.
2. Click **Clear Dependencies**. The task is launched normally.

To clear dependencies on a task instance from the workflow monitor

1. View the workflow that contains the task instance whose dependencies you want to satisfy.
2. Select the task instance(s) for which you want to clear upstream dependencies.
3. Select **Commands**.
4. Select **Clear Dependencies**. The task is launched normally.

Marking a Specific Dependency as Satisfied

For tasks running inside of a workflow, you can clear a single upstream dependency to allow the task to run. Clearing a dependency has the same result as satisfying a dependency. You can clear a
dependency on tasks in the following status: Defined, Waiting, Held.

1. View the workflow that contains the task instance whose dependencies you want to satisfy.
2. Locate and right-click on the task dependency (the connector line between two tasks).
3. Select **Commands**.
4. Select **Mark as Satisfied**. If all other dependencies are satisfied, the task is launched normally.

Monitoring and Controlling Applications

Introduction

The Opswise Application Monitoring and Control features allow you to use Opswise Automation Center as a network control and monitoring tool. You can use these resources to start, stop, and query any application running on any machine where you have an Opswise agent installed and running.

You will use the following three components to monitor your applications:

- The Application Resource record defines the name and location of the application, along with the necessary control commands. The list of application records displays a status for each application.
- Three Application Control tasks are automatically generated when you save an Application Resource record to the Opswise database: one each for executing a start, stop, and query command against the application. You can use these control tasks to schedule the Start, Stop, and Query commands in Workflows and triggers. You also can create customized Application Control tasks where necessary.

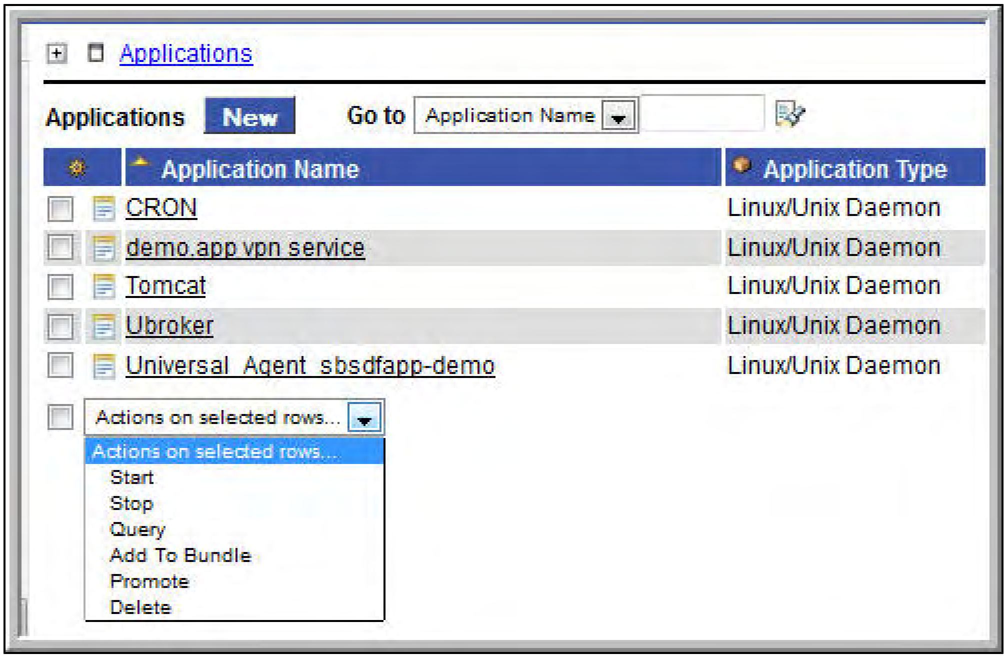
> **Note**
> You can manually run an Application Control task to execute a command specified in an Application resource record, but it is simpler to just execute the command from the Applications list or Application resource record.

- Optional Application Monitor Triggers allow you to launch other tasks based on the status of the Application Resource record being monitored.

Processing Flow

The following steps show a sample process flow for the manual monitoring (that is, not via a trigger or Workflow) of an application:

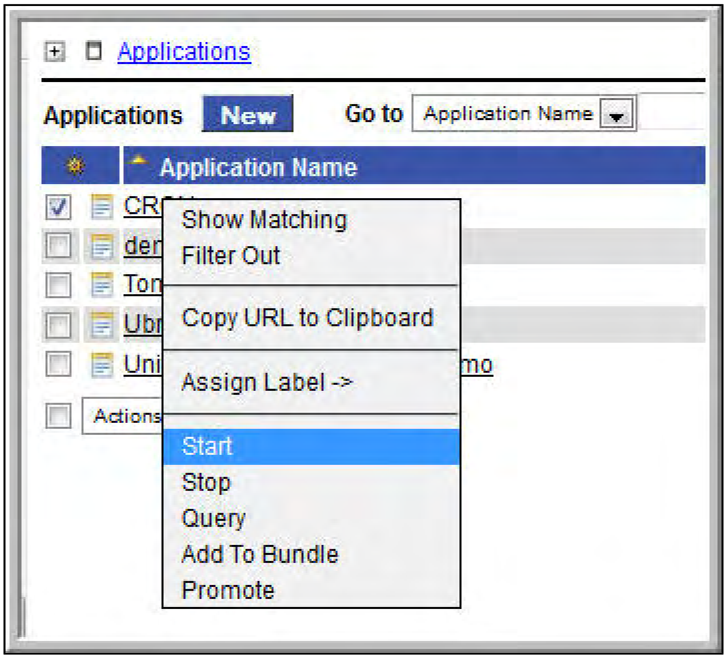
1. Use the Application Resource screen to create an Application resource record, specifying the name of an application and the start, stop, and query commands to control it. The Controller will automatically create three Application Control tasks that you can use in Workflows and triggers for starting, stopping, and querying the application.
2. Start the application using either of the following methods:
    a. Click the box to the left of the Application name to select it. You can select as many Applications as you want.
    b. From the Actions on selected rows... menu, select **Start**.



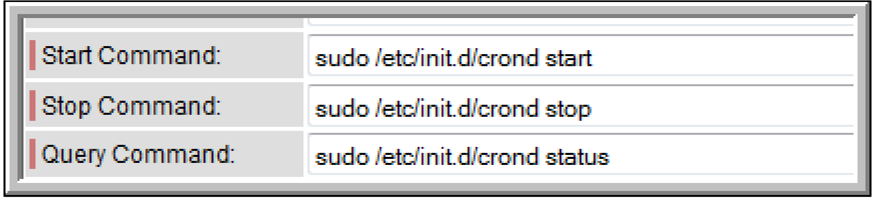
Application Name	Application Type
CRON	Linux/Unix Daemon
demo.app vpn service	Linux/Unix Daemon
Tomcat	Linux/Unix Daemon
Ubroker	Linux/Unix Daemon
Universal_Agent_sbsdfapp-demo	Linux/Unix Daemon

OR

1. To start a single application, right-click the Application Name and select **Start** from the pop-up menu.

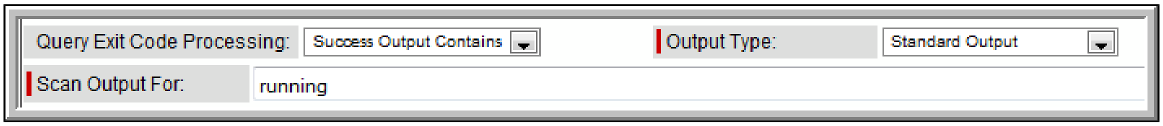


The system executes the Start command, as provided by the user in the Application Resource definition. It puts the Application into Starting status, and saves the Start Time. The Start command has two functions: 1) it starts the application and 2) it starts the query process that monitors the application.
After 30 seconds, the system automatically executes a Query command, as provided by the user in the Application resource definition, to determine the status of the application. The system continues executing the Query command every 120 seconds thereafter until the user stops the monitoring by issuing a Stop command from Opswise.



Start Command:	sudo /etc/init.d/crond start
Stop Command:	sudo /etc/init.d/crond stop
Query Command:	sudo /etc/init.d/crond status

The purpose of the Query is to determine whether or not the Application is Active. The system uses the specifications provided by the user in the Exit Code Processing fields (example shown below) to make this determination.

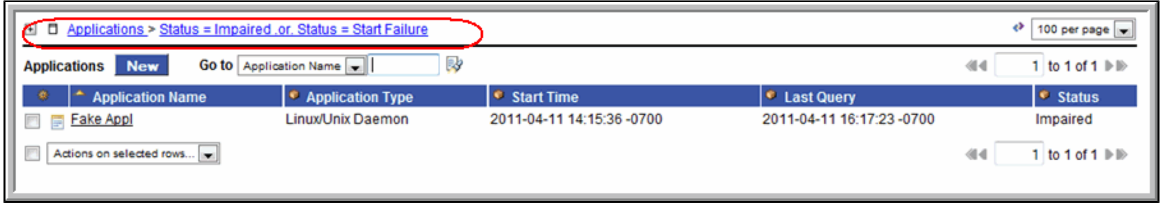


Query Exit Code Processing:	Success Output Contains	Output Type:	Standard Output
Scan Output For:	running		

- If the response from the application indicates a successful startup, the system puts the Application in Active status.
- If the response indicates the Application hasn't started, the system continues executing the Query (keeping track in the Startup Query Attempts field) until it reaches the maximum attempts specified by the user in the Startup Query Maximum field. If the maximum number is reached before achieving an Active status, the system puts the Application into Impaired status. However, the system continues monitoring. If the appropriate exit code parameters are eventually returned, the application will go to Active status. The purpose of the Startup Query Attempts field is to avoid having the application go right into Impaired status if it takes awhile to start.

The system writes any Exit Code captured by the Query in the Query Exit Code field.
After starting the application, Opswise continues monitoring by sending out the Query commands every 120 seconds.

- If Opswise detects a problem based on the Exit Code parameters, it puts the Application into Impaired status. If this occurs, you have several options for handling the problem, with increasing levels of automation:
  - The Application list displays the status of all Applications being monitored. You could create a filter for the Application list that displays only those with status of Impaired (or other), as shown in the example below. When you see a problem, troubleshoot the issue and restart the Application from outside Opswise.

    Applications > Status = Impaired .or. Status = Start Failure
100 per page
Applications
New
Go to
Application Name
1 to 1 of 1
Actions on selected rows...

Application Name	Application Type	Start Time	Last Query	Status
Fake Appl	Linux/Unix Daemon	2011-04-11 14:15:36 -0700	2011-04-11 16:17:23 -0700	Impaired

  - Set up an Application Monitor trigger that monitors this application for Impaired and other problem statuses. When the trigger is satisfied, it launches an email task that sends emails to support personnel, notifying them of the problem. Several built-in variables are supported that allow you to pass required data into the email message: the application name, type, and status.
  - You could also create a workflow launched by an Application Monitor trigger looking for Impaired or other problem statuses. The workflow can include Application Control tasks that attempt to resolve the problem by stopping and then re-starting the application. You could also include any other tasks that are specific to troubleshooting the application.
- If Opswise fails to get a response to a Query for three minutes, it puts the Application into <status>Query Overdue status, where <status>is the last known query status. This can be any of the following: Starting/Query Overdue, Active/Query Overdue, and Impaired/Query Overdue. For example, you may see this status if the agent went down or there was some other problem on the machine unrelated to the application itself. If this occurs, you should troubleshoot the issue. When you have fixed the problem, the continued queries from Opswise will then return an Active status for the application.

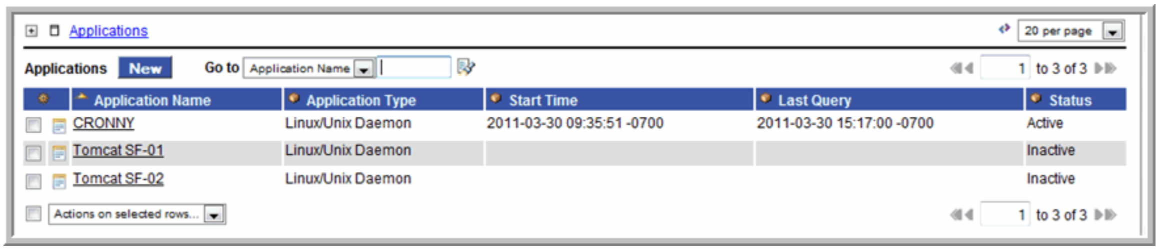
To stop monitoring an Application, issue the Stop command against it. Opswise stops the Application and puts it into Inactive status, which means it is no longer monitoring.

Applications

Overview

Application Resource records are the core component of the Opswise Application Monitoring and Control function. These records define the names of the Applications being monitored, the name and location of the machines where they are running, and the start, stop, and query commands needed to perform the monitoring and control functions.

Shown below is a sample Applications resource list, which displays all the Applications you have set up to be monitored. You must manually refresh this screen to fetch the latest status information.



Application Name	Application Type	Start Time	Last Query	Status
CRONNY	Linux/Unix Daemon	2011-03-30 09:35:51 -0700	2011-03-30 15:17:00 -0700	Active
Tomcat SF-01	Linux/Unix Daemon			Inactive
Tomcat SF-02	Linux/Unix Daemon			Inactive

You can also use Application resources and their associated Application Control tasks to start, stop, and query applications as part of your scheduling processes. You can execute Application Control tasks as you would execute any other task and include them in workflows where applicable. In addition, you can define Application Monitor triggers to automatically launch one or more tasks of any type, depending on the status of one or more Application resources. For example, you might set up an Application Monitor trigger that sends an email to Windows tech support personnel if any Windows application goes to Impaired or Inactive status.

In order for Opswise to access the application, the application must be installed on a machine where an Opswise Windows, Linux/Unix, or z/OS agent is running.

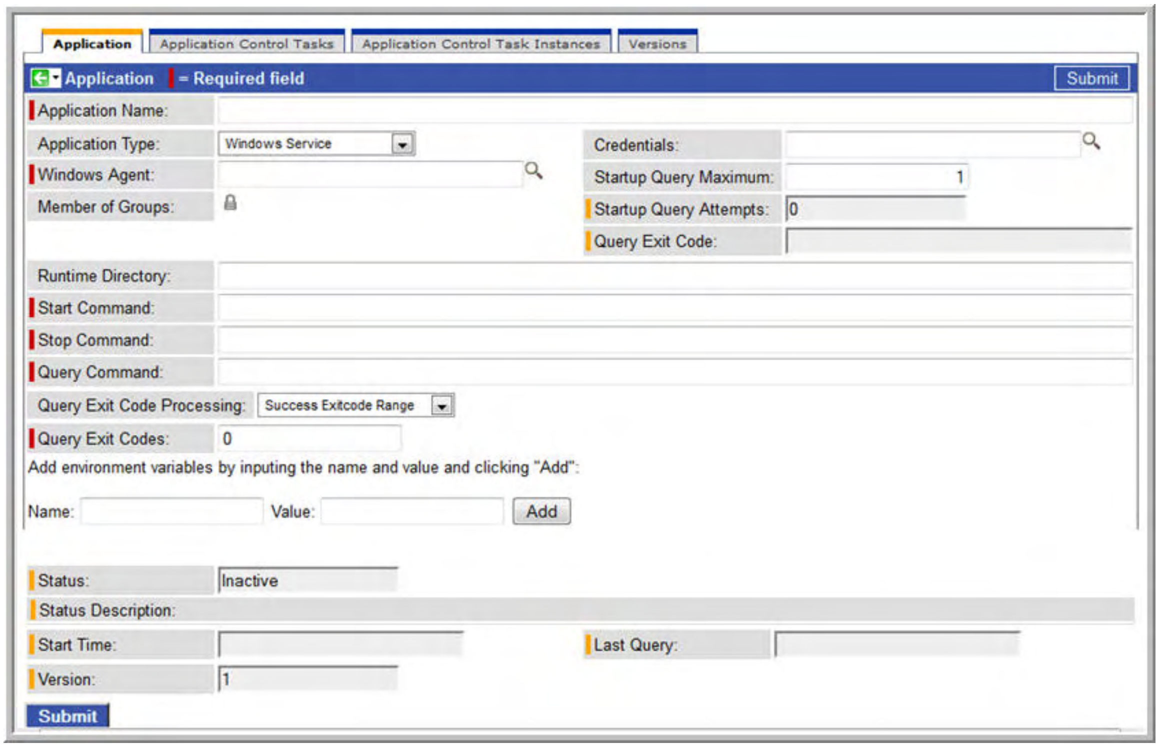
If you set up Opswise to monitor your applications, you should always start and stop the applications from within Opswise. If you stop an application outside of Opswise, you must also restart it from outside of Opswise. If Opswise detects a problem with an application (the application goes to Impaired status), you should troubleshoot the problem and restart the application outside of Opswise. Opswise will continue monitoring and when it detects that the application is back up, it will put the application back into Active status.

Built-In Variables

Several built-in variables are provided to pass information about the Application being monitored into the task or tasks being launched by the trigger. You can pass the information into the launched tasks by including the variables in a text field in the task definition. See Application Monitor Variables for details.

Creating a New Application Resource Record

1. From the navigation pane, select **Automation Center Resources > Applications**.
2. Click **New**. The Application screen displays.



Application Name:

Application Type: Windows Service

Startup Query Maximum: 1

Startup Query Attempts: 0

Query Exit Code Processing: Success Exitcode Range

Query Exit Codes: 0

Add environment variables by inputing the name and value and clicking "Add":

Status: Inactive

Version: 1

3. Using the field descriptions provided below as a guide, complete the fields as needed.
4. Click the **Submit** button to save the record and return to the menu, or right-click on the title bar and select **Save** to save the record and remain on the current display. When you save the new Application Resource record, Opswise also automatically creates three related tasks, one each for starting, stopping, and querying the application.
5. If appropriate, repeat these steps for any additional Applications you want to add.

Application Field Descriptions

Field Name	Description
Application Name	Required. Name used within Opswise to identify this resource. Up to 40 alphanumerics. It is the user's responsibility to develop a workable naming scheme for resources.
Application Type	User-defined. The type of application. Options: • Windows Service • Linux/Unix Daemon • z/OS Started Task
Credentials	Optional. The login credentials Opswise will use to access the remote machine. See Credentials. For z/OS application resources, make sure the credentials are in upper case.
Agent	Required. The name of the Windows, Linux/Unix, or z/OS agent resource that describes the machine where the application will run.
Run as sudo	Optional; Linux/Unix only. Run the command as Sudo (superuser do).
Member of Groups	User-defined. Allows you to select one or more Opswise groups that this task definition belongs to. Click on the lock icon to unlock the field and select groups.
Startup Query Maximum	Required; user-defined. The default is 1. Allows you to specify the maximum number of times the system should query for Active status during startup before it puts the application into Impaired status. For applications that take awhile to start, you should specify a higher number. When you issue a Start command, the system issues the Start, waits 30 seconds, then executes the Query command. It continues executing the Query command every 120 seconds thereafter (until you issue a Stop command). If the startup takes longer than the maximum queries specified, the application goes to Impaired status. However, the system continues to query. If the required exit codes are eventually returned, the application then goes to Active status.

Startup Query Attempts	System-supplied. The number of queries that were executed before the Application went into Active or Impaired status.
Query Exit Code	System-supplied. The most recent exit code returned by the application in response to a query.
Runtime Directory	Optional. The directory where the application executes. Variables supported.
Start Command	Required. The command used to start the application. This can be any process or command that starts the application. If you try to start an application monitor that is already started, you will see the message: Application already monitored with <status> status.
Stop Command	Required. The command used to stop the application. This can be any process or command that stops the application.
Query Command	Required. The command used to query the application. This can be any process or command that queries the application. You must first start the application monitor from Opswise before you can query the application.
Exit Code Processing	Specifies how Opswise should determine whether the application is running or not. Options: • Success Exitcode Range - The application goes to or remains in Active status if its exitcode falls within the range specified in the Query Exit Codes field (see below). Otherwise it has Impaired status. • Failure Exitcode Range - The application goes to or remains in Impaired status if its exitcode falls within the range specified in the Exit Codes field (see below). Otherwise it has Active status. • Success Output Contains - The application goes to or remains in Active status if its output contains the text specified in the Scan Output For field (see below). Otherwise it has Impaired status. • Failure Output Contains - The application goes to or remains in Impaired status if its output contains the text specified in the Scan Output For field (see below). Otherwise it has Active status.
Query Exit Codes	If Exit Code Processing = Success Exitcode Range or Failure Exit Code Range, this field specifies the range. Format: Numeric. Use commas to list a series of exitcodes; use hyphens to specify a range. Example: 1,5, 22-30.
Output Type	If Exit Code Processing = Success Output Contains or Failure Output Contains, this field specifies the type of output. Options: Standard Output (STDOUT) Standard Error (STDERR) File
Scan Output For	Required if output is being scanned. Specifies the string that the system should scan for in the output.
Output File	Required if Output type=File. The path and name of the file.
Status	System-supplied. Indicates the current status of the application. One of the following: • Inactive - The application is not being monitored by Opswise. • Start Failure - The application failed to start. This may occur, for example, if you have problems with credentials or the start command itself is incorrect. When this occurs, Opswise is not monitoring the application. You should troubleshoot the problem and restart the application from Opswise. • Starting - The Start command has been executed. • Active - The application has successfully started and is running, based on the parameters specified in the Exit Code processing fields. • Impaired - An application that is being monitored returned a response that, based on the specified exit code parameters, indicates that it is not running. If this occurs, you should troubleshoot the problem and restart the application from outside Opswise. Unless you issue a stop command, Opswise continues monitoring during this process. When the application comes back up, the query process will recognize this and return the application to Active status.
Status Description	System-supplied. A more detailed status message describing why a status change occurred, in the format: "Query exit code <in-not in> <success-failure> exit code range. Query <success-failure> output not found."
Start Time	System-supplied. The date and time that the application was last started by Opswise.
Last Query	System-supplied. The date and time of the last query response received from the application.
Version	Version number of the current record, which is incremented by the system every time a user updates a record. Click on the Versions tab to view previous versions. For details, see Record Versioning.
Application Control Tasks tab	Lists all Application Control tasks associated with this Application resource.

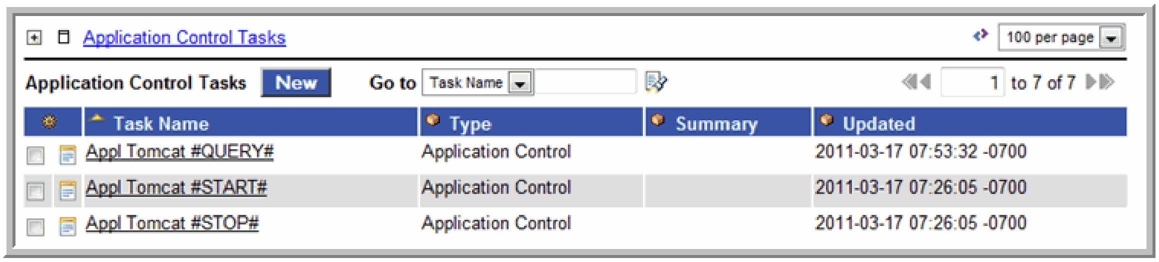
Application Control Task Instances tab	Lists all Application Control task instances associated with this Application resource.
Versions tab	Stores copies of all previous versions of the current record (see Record Versioning).
Update button	Updates this record with any changes.
Start button	Executes the Start command associated with this Application resource and begins querying.
Query button	Executes the Query command associated with this Application resource. This allows you to get immediate status of the application instead of waiting for the next automated query.
Stop button	Executes the Stop command associated with this Application resource. Opswise stops the application and stops querying (monitoring).
Delete button	Deletes this Application Resource record.

Application Control Tasks

Overview

Application Control tasks allow you to execute a start, stop, or query command against an application in the Opswise network. Three Application Control tasks are created automatically when you define and save an Application Resource record – one each for starting, stopping, and querying the application.

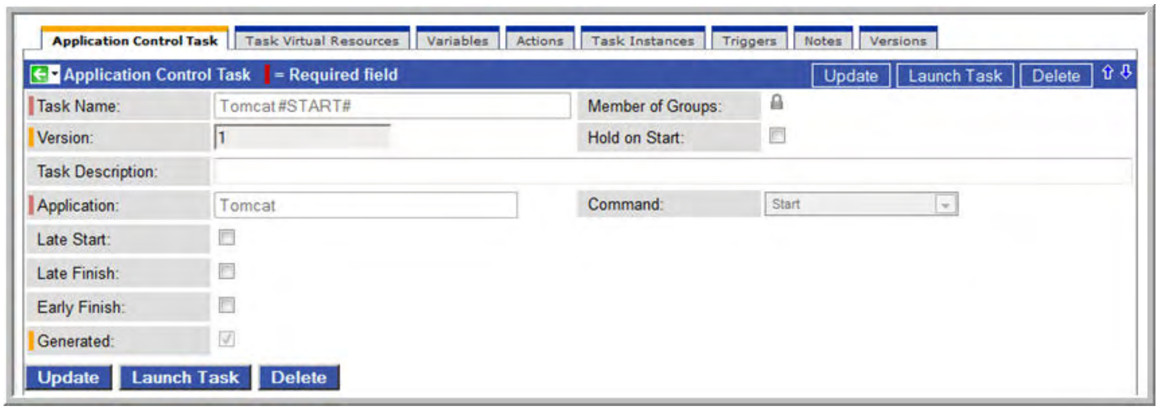
The following shows three Application Control tasks created when an Application resource record called "Appl Tomcat" was saved to the Opswise database:



	Task Name	Type	Summary	Updated
□	Appl Tomcat #QUERY#	Application Control		2011-03-17 07:53:32 -0700
□	Appl Tomcat #START#	Application Control		2011-03-17 07:26:05 -0700
□	Appl Tomcat #STOP#	Application Control		2011-03-17 07:26:05 -0700

Each of these auto-created records is stored as a separate task record and can be executed independently or added to a workflow like any other task.

The following shows the task record for the Appl Tomcat#START# task that was automatically created when the Appl Tomcat Application resource record was submitted to the database. Note that many of the fields are protected and the Generated field at the bottom is selected, indicating that this task was generated automatically. You cannot delete an auto-generated Application Control task.



Task Name:	Tomcat#START#	Member of Groups:	
Version:	1	Hold on Start:	□
Task Description:			
Application:	Tomcat	Command:	Start
Late Start:	□		
Late Finish:	□		
Early Finish:	□		
Generated:	☑		

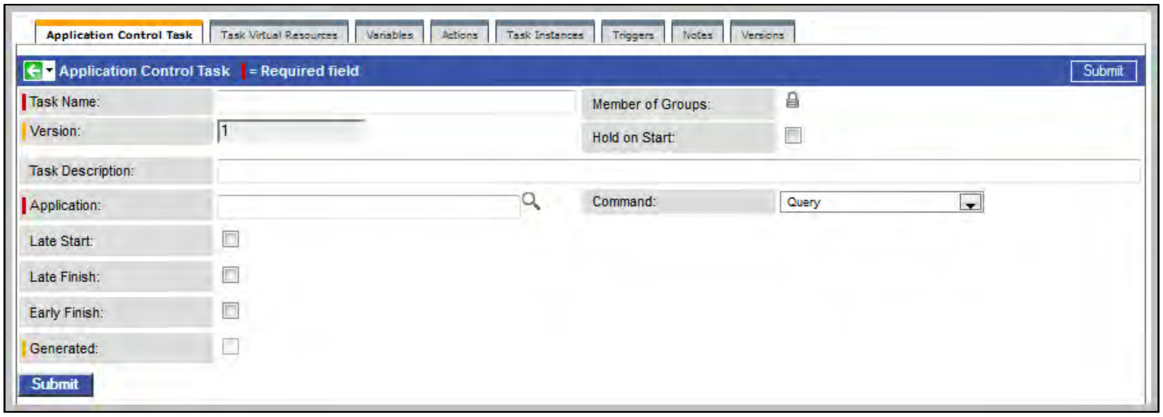
Built-In Variables

Several built-in variables are provided to pass information about the Application being monitored into the task or tasks being launched by the trigger. You can pass the information into the launched tasks by including the variables in a text field in the task definition. See Application Monitor Variables for details.

Creating a New Application Control Task

You can also manually create Application Control tasks. You might want to do so if you need to set different variables or conditions for different uses of the same task.

1. From the Navigation Pane, select **Application Control Tasks**. The Application Control Tasks List screen displays.
2. Click **New**. The Application Control Task screen displays.



3. Using the field descriptions provided below as a guide, complete the fields as needed.
4. Click the **Submit** button to save the record and return to the menu, or, right-click on the title bar and select **Save** to save the record and remain on the current display.
5. If appropriate, repeat these steps for any additional tasks you want to add.

Application Control Task Field Descriptions

The table below describes the fields, buttons, and tabs on the Application Control task definition and task instance screens. Most fields appear on both screens; however they do not always appear at the same spot. In the latter case, the table provides a field description at the location found on the task definition screen. Some fields appear only on one of the screens.

Field Name	Description
Task/Instance Name	Required. Name used within Opswise to identify this task. Up to 40 alphanumerics; variables supported. It is the user's responsibility to develop a workable naming scheme for tasks.
Summary	User-supplied description of this record.
Application	Required; protected if auto-generated. The name of the Application resource defined using the Application resource screen. The Application resource defines where the software application is running; it also defines the start, stop, and query commands for the application. Type in a name, or click the magnifying glass to browse to an existing Application resource record or create a new one.
Command	Required; protected if auto-generated. The command this task is executing against the software application. One of the following: • Query • Start • Stop
Hold on Start	If enabled, when the task is launched it appears in the Activity display with a status of Held. The task runs when the user releases it.
Member of Groups	User-defined. Allows you to select one or more Opswise groups that this record definition belongs to. Click on the lock icon to unlock the field and select groups.
Version	Task definition only; system-supplied. The version number of the current record, which is incremented by the system every time a user updates a record. Click on the Versions tab to view previous versions. For details, see Record Versioning.

Late Start	If enabled, and if the task instance starts after the time specified in the Late Start Time field (see below), the task instance is flagged as late. To determine whether a task instance started late, open the task instance and locate the Started Late field; the field is checked if the instance started late.
Late Start Type	Required if Late Start is enabled. Options are: - **Time** - Flag the task if it starts after the specified time. - **Duration** - Flag the task if it starts a certain amount of time after the programmed start time. The task must have a specific start time.
Late Start Time	Time after which the task start time is considered late. Use hh:mm, 24-hour time
Late Start Duration	Duration (amount of relative time) after which the task is considered to have started late. For a task within a workflow, the duration is the period between the time the workflow starts and the time the task itself starts. For example, a task might have a Late Start Duration of 60 minutes. If the workflow starts at 9:00 a.m. but the task itself does not start until 10:30, the task has started late. For a task that is not within a workflow, Late Start Duration has meaning only if the task has been held upon starting. For example, if a task has a Late Start Duration of 60 minutes and the Hold on Start field is enabled, if the task is not released from hold within the amount of time specified in the Late Start Duration field, the task has started late.
Started Late	Task instance only; system-supplied. This field is flagged if the task started later than the time specified in the Late Start fields.
Late Finish	If enabled, and if the task instance finishes after the time or period specified, the task instance is flagged as late. You can specify a time or duration to determine a late finish (see Late Finish Type below). To determine whether a task instance finished late, open the task instance and locate the Finished Late field; the field is checked if the instance finished after the specified time or lasted longer than expected. This field only appears on the task instance if the user specified a Late Finish in the task definition.
Late Finish Type	If Late Finish is enabled, use this field to specify whether you want to use time (see Late Finish Time below) or duration (see Late Finish Duration below) to determine whether the task instance finished late.
Late Finish Time	If Late Finish Type is Time, use this to specify the time after which the task finish time is considered late. Use hh:mm, 24-hour time.
Late Finish Duration	If Late Finish Type is Duration, use this to specify the longest amount of time this task instance should take to run. You can specify any combination of hours, minutes, and seconds.
Finished Late	Task instance only; system-supplied. This field is flagged if the task finished later than the time or duration specified in the Late Finish fields.
Early Finish	If enabled, and if the task instance finishes before the time or period specified, the task instance is flagged as early. You can specify a time or duration to determine an early finish (see Early Finish Type below). To determine whether a task instance finished early, open the task instance and locate the Finished Early field; the field is checked if the instance finished before the specified time or did not last as long as expected. This field only appears on the task instance if the user added Early Finish specifications to the task definition.
Early Finish Type	If Early Finish is enabled, use this field to specify whether you want to use time (see Early Finish Time below) or duration (see Early Finish Duration below) to determine whether the task instance finished early.

Early Finish Time	If Early Finish Type is Time, use this to specify the time before which the task finish time is considered early. That is, enter a time at which the task should still be running. Use hh:mm, 24-hour time.
Early Finish Duration	If Early Finish Type is Duration, use this to specify the shortest amount of time this task instance should take to run. You can specify any combination of hours, minutes, and seconds.
Finished Early	Task instance only; system-supplied. This field is flagged if the task finished earlier than the time specified in the Early Finish fields.
First Time Ran	Task definition only; system-supplied. Displays after the first time the task runs. The date and time this task first ran.
Lowest Instance Time	Task definition only; system-supplied. Displays after the first time the task runs. The shortest amount of time this task has taken to run.
Last Time Ran	Task definition only; system-supplied. Displays after the first time the task runs. The most recent date and time the task ran.
Average Instance Time	Task definition only; system-supplied. Displays after the first time the task runs. Shows the average amount of time this task takes to run.
Number of Instances	Task definition only; system-supplied. Displays after the first time the task runs. Shows the number of times this task has run.
Highest Instance Time	Task definition only; system-supplied. Displays after the first time the task runs. The longest amount of time this task has taken to run.
Generated	System-supplied; protected. If selected, indicates that this Application Control task was generated automatically when the Application resource record was submitted.
Submit button	Submits the new record to the database.
Update button	Saves updates to the record.
Launch Task button	Manually launches the task.
Delete button	Deletes the current record.
View Parent button	Task instance only; Displays this task's parent task (workflow), if any.
Show Details button	Task instance only; displays detailed information about this task instance.
Skip button	Task instance only; for tasks loaded into the schedule that haven't run yet. Allows you to tell the system to skip this task. See Skipping a Task.

Hold button	Task instance only; see Putting a Task on Hold.
Force Finish button	Task instance only; see Force Finishing a Task.
Output tab	Task instance only. Displays output generated from the process, if any, based on specifications provided by the user in the Automatic Output Retrieval fields in the task definition.
Task Virtual Resources tab	Lists Virtual Resources to which this task is assigned. See Creating Virtual Resources.
Variables tab	Displays all variables associated with this record.
Actions tab	Allows you to specify actions that the system will take automatically based on events that occur during the execution of this task. Events are task instance status, exit codes, late start, late finish, and early finish. Actions are: • **Abort Action** - Abort the task if certain events occur. For details, see Setting Up Abort Actions. • **Email Notification** - Send an email if certain events occur. For details, see Creating Email Notifications. • **SNMP Notification** - Send an email if certain events occur. For details, see Creating SNMP Notifications. • **Set Variable** - Used in workflows to set a variable based on the occurrence of certain events. For details, see Setting Variables in a Workflow.
Task Instances tab	System-supplied. Displays a list of all task instances generated by this task definition.
Triggers tab	Displays a list of all triggers that have been defined to launch this task. Also allows you to add new triggers. If you add a new trigger from this location, the system automatically constructs a default trigger name as follows: <current task name>#TRIGGER#. You can change the default name if desired. For instructions on creating triggers, see Creating Triggers.
Notes tab	Displays all notes associated with this task. See Runbook Notes.
Versions Tab	Stores copies of all previous versions of the current record (see Record Versioning).

Specifying When a Task Runs

You can run the task as part of a workflow, specify triggers that run the task automatically based on times or events, or run the task manually.

Monitoring Task Execution

You can monitor all system activity from the Activity screen and can view activity history from the Activity History screen.

Home Page, Dashboard, and Gauges

The Home Page is associated with the login ID and is customizable by each user. You access the Home Page by clicking the Home 🏠 icon.

The Dashboard also is customizable and is accessed by selecting **Automation Center > Dashboard** from the navigation pane.

Both the Home Page and the Dashboard use Gauges to display data. You can add or delete gauges from the Home Page and Dashboard and you can create your own gauges.

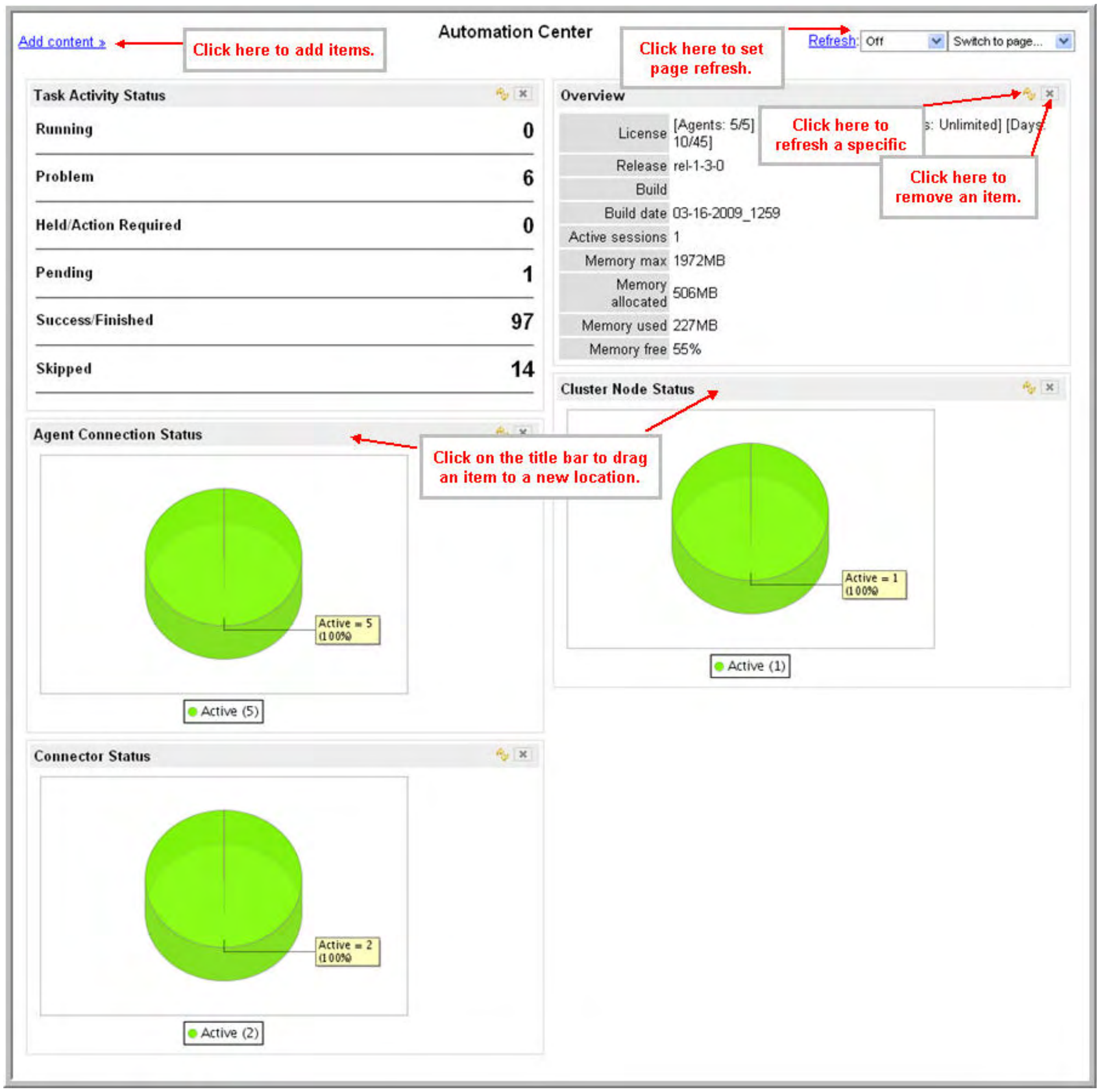
Customizing Your Home Page

Overview

The home page is the page that displays when you log on to Opswise. Its content consists of a number of windows. Each window contains a specific set of information, often displayed in graphical format. Once you navigate away from the home page, you can return to it by clicking the Home 🏠 icon.

A sample home page is shown below. You can click on the gauges to display detailed information.

- The Task Activity Status gauge provides a summary of task instances, sorted by status. You can click on any status for detailed information.
- The Overview displays system configuration information.
- The Agent Connection status gauge shows the status of all connected agents.
- The Cluster Node gauge shows the status of defined cluster nodes. Cluster nodes are created automatically.
- The Connector Status gauge shows defined connectors. The connector is a communications link between the core processor and Opswise agents. These are created automatically.



Add content »

Click here to add items.

Automation Center

Click here to set page refresh.

Refresh: Off

Switch to page...

Task Activity Status

Running	0
Problem	6
Held/Action Required	0
Pending	1
Success/Finished	97
Skipped	14

Overview

Click here to refresh a specific

Click here to remove an item.

License	[Agents: 5/5] 10/45] s: Unlimited] [Days
Release	rel-1-3-0
Build	
Build date	03-16-2009_1259
Active sessions	1
Memory max	1972MB
Memory allocated	506MB
Memory used	227MB
Memory free	55%

Agent Connection Status

Click on the title bar to drag an item to a new location.

Cluster Node Status

Connector Status

Moving Items

Each item on the home page has a grey title bar. To move an item, grab the title bar and drag the item to its new position.

Removing Items

To remove an item, click the **X** in the top right corner of the item's header.

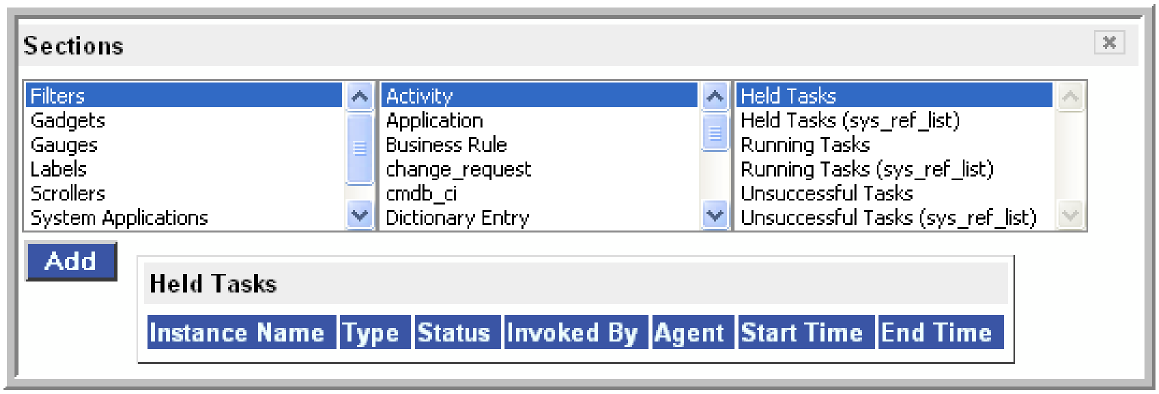
Refreshing

You can specify a refresh time on the home page of 5, 15, or 30 minutes, 1 hour, or you can specify Off (no refresh), which is the default. To manually refresh the page, click **Refresh**. To refresh a specific item on the page, click the item's Refresh 🔄 icon.

Adding Items

Many items are available to add to your home page. Other items listed in the drop-down menus refer to program internals and are not applicable to your installation. These items are omitted from the table descriptions provided below.

1. Click **Add Content**. The system displays a list of options as shown below.



2. Using the descriptions provided in the table below, make a selection in each of the columns to define the content you want to add.
3. Click **Add**. The content you specified is added as a new section to your home page.

Item Descriptions

The following table provides a description for all items that are applicable to an Opswise installation.

First Column	Second Column	Third Column
Filters – A filter is a list of records whose content is defined by the filter.	Type of filter. The content of this list is defined by users. When you create and save a filter on any Opswise list, such as Triggers, Tasks, or other types of record lists, the type of filter is added to this list (Triggers, Tasks, and so on). For example, if you create and save a filter on the Task list called "Windows Tasks," the second column in this dialog displays "Tasks" and the third column displays "Windows Tasks." There may also be some items in this list that are created by the interface platform and are not of interest.	This column displays this list of filters for the type selected in the second column.
Gadgets-- A gadget is a hard-coded feature. Two types are currently supported: sticky notes and system information.	**Sticky Note**. Allows users to type informational messages into the home page.	
	System Information. The Overview option provides information about the system, such as build date, active sessions, number of transactions, and so on.	
Gauges. A gauge is a report that has been defined and packaged into a gauge. See Reports for instructions about how to create a new report and package it into a gauge, or how to package an existing report into a gauge.	This column provides a list of gauge types that have been defined. The gauge type corresponds to the table that was selected when the report was defined, such as Activity or Audit Records.	This column provides a list of reports that have been defined using the table (gauge type) shown in the second column. For example, a number of Activity reports have been defined, such as "Active Task Instances By Status." Those reports display in this list.
Labels. Labels allow you to customize the navigation pane with business-centric options. For example, you might want to add the option "Fee-Related Tasks" to the navigation pane, then add your fee-related tasks to that menu option. For more information, see Customizing the Navigation Pane.	This list shows all labels that have been added to the navigation pane.	This list shows records that have been assigned to the label.
Scrollers	Not implemented.	
System Applications	Allows you to insert links to some sections of the navigation pane.	

Using the Dashboard

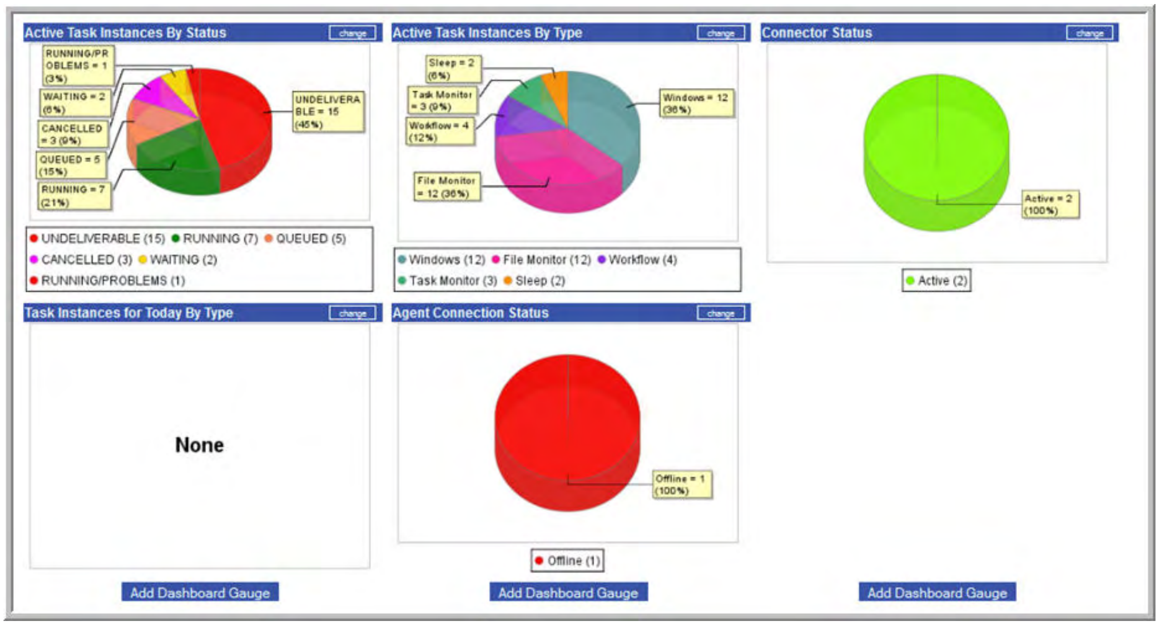
Overview

A dashboard allows you to set up a display of information that users commonly refer to throughout the day. The information is extracted from the database by way of one or more gauges. The dashboard is accessed by clicking **Automation Center > Dashboard** from the navigation pane.

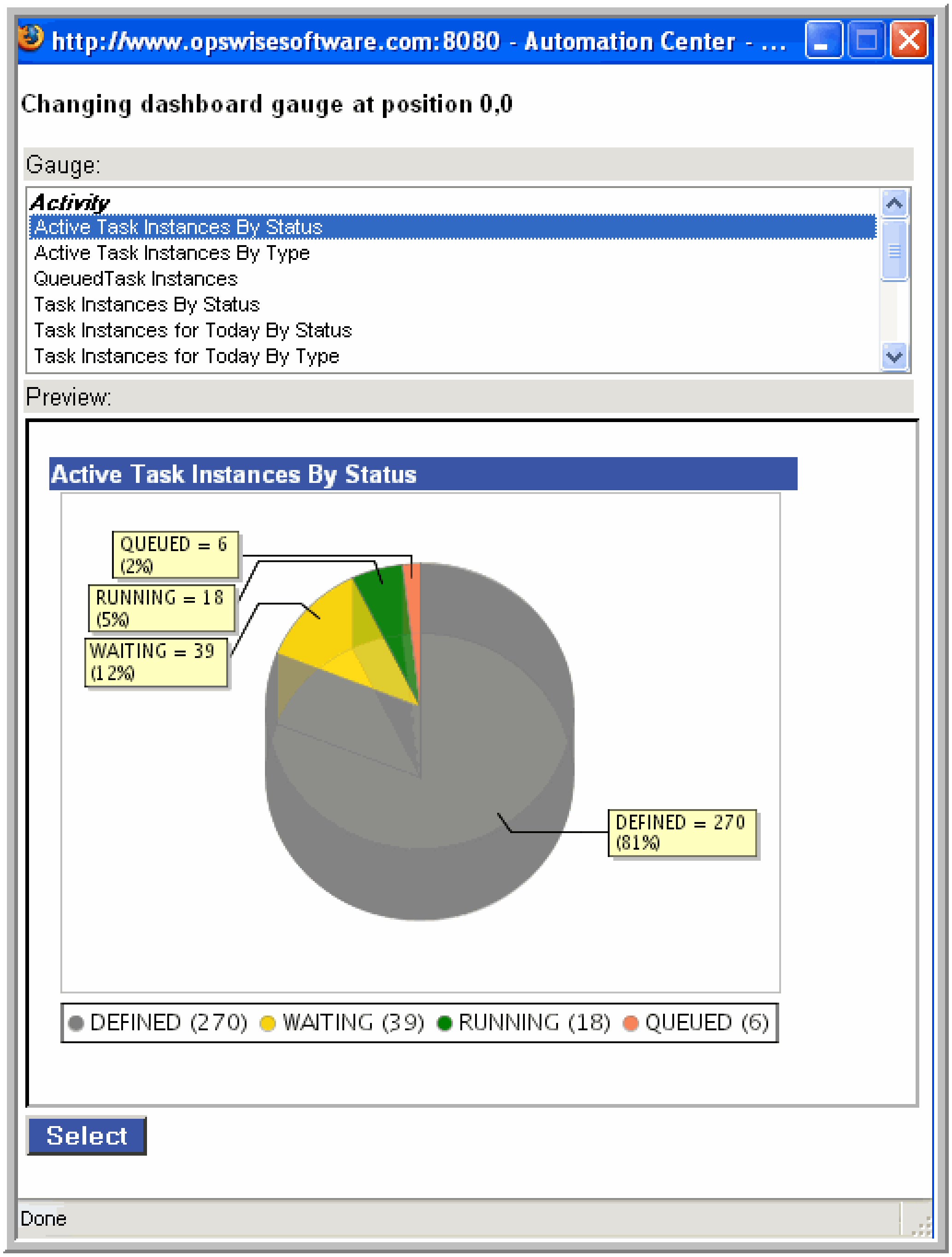
Customizing the Dashboard

The dashboard provides slots for up to nine gauges. Opswise is distributed with a set of default gauges and a default dashboard configuration.

1. If the gauge doesn't yet exist, follow the instructions in Adding and Deleting Gauges to create the new gauge.
2. Select **Automation Center > Dashboard** from the navigation pane. A default dashboard displays, as shown in the example below.



3. To add a gauge to an empty slot, click **Add Dashboard Gauge**. Or, to replace a gauge with a different one, click the **change** button on the gauge you want to change. The system displays a list of available gauges.



http://www.opswisesoftware.com:8080 - Automation Center - ...

Changing dashboard gauge at position 0,0

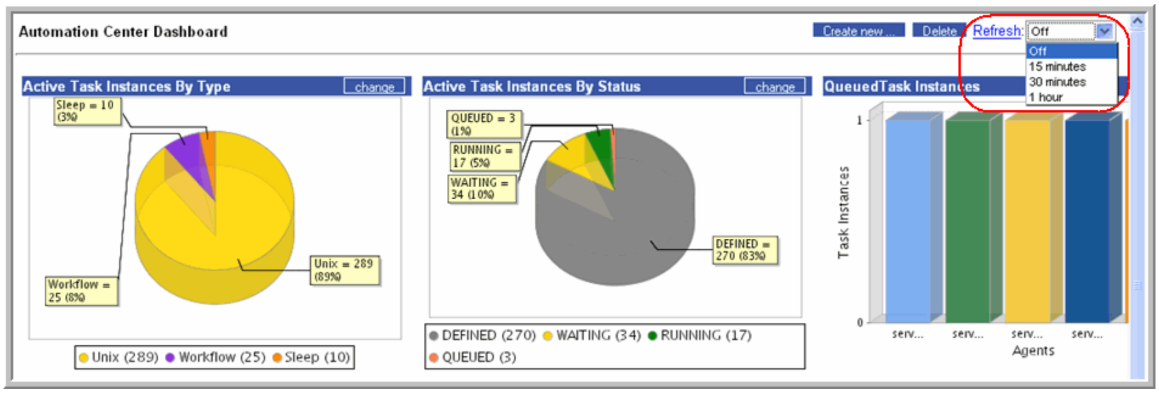
Gauge: Active Task Instances By Status

Gauge:
Activity
Active Task Instances By Status
Active Task Instances By Type
QueuedTask Instances
Task Instances By Status
Task Instances for Today By Status
Task Instances for Today By Type
Preview:
Active Task Instances By Status
QUEUED = 6 (2%)
RUNNING = 18 (5%)
WAITING = 39 (12%)
DEFINED = 270 (81%)
● DEFINED (270) ● WAITING (39) ● RUNNING (18) ● QUEUED (6)
Select
Done

4. Select the new gauge from the list. The system replaces the old gauge with the new one you selected.
5. Repeat the above procedure for any other changes you want to make.

Refreshing Dashboard Data

You can manually refresh the data displayed on your dashboard and set an automatic refresh rate.



Manually Refreshing Dashboard Data

Click **Refresh**. The system repaints the screen with the latest data.

Setting the Dashboard Refresh Rate

1. Click on the down arrow (see above illustration) to display refresh rate options.
2. Select the refresh rate you want. Options are: Off; 5, 10, 15, or 30 minutes; 1 hour.

Adding and Deleting Gauges

Adding Gauges

See Creating a Gauge from a Report.

Deleting Gauges

To remove a gauge from the dashboard or home page, you must delete the gauge. Deleting a gauge will remove it from any dashboard or home page where it is displayed.

1. From the navigation pane, select **Automation Center Administration > Configuration > Gauges**.
2. Select the gauge you want to delete.
3. Select **Delete** from the **Actions on selected rows ...** menu. See Deleting Records for detailed instructions.

Reports

Introduction

The Opswise installation includes a number of predefined reports, as described below. You can also create, save, and run your own reports as needed. The Activity Screen also uses reports created using this feature to define what task instances are displayed. When you create a report for the Activity Screen, you select records only from a specific table called the Activity table. When you save the report, it appears automatically in the drop-down menu on the Activity screen. For normal reports, the report appears on the Reports menu when you save it.

When you create reports, you need to know the table names of the data you want to include. See Opswise Report Tables.

Predefined Reports

The following reports have been predefined within Opswise:

Agent Connection

- Agent Connection Status
- All Agents

Application

- Application Status

Connector

- All Connectors

Activity

- Active Task Instances
- Active Workflow Task Instances
- Active/Late Task Instances
- All Task Instances by Status
- Cancelled Task Instances
- Held Task Instances
- In Doubt Task Instances
- Queued Task Instances
- Running Task Instances
- Task Instances Due to Finish in the Next 3 Hours
- Task Instances Due to Finish in the Next Hour
- Today's Failed Task Instances by Status
- Today's Successful Task Instances
- Today's Task Instances by Created Time
- Today's Task Instances by Type
- Undeliverable Task Instances
- Unsuccessful Task Instances
- Waiting For Resources Task Instances

- Waiting Task Instances
- Workflow Task Instances
- Workflow Task Instances with Problems

History

- Duration

Forecasts

- Forecast - All
- Forecast - All - List
- Forecast - Daytime
- Forecast - Overnight
- Forecast - Today
- Forecast - Weekend

User

- Users

Group Member

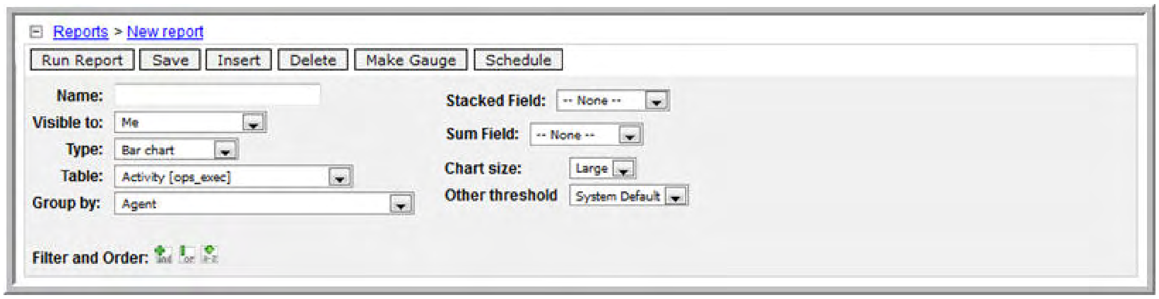
- Group Membership

User Role

- Role Allocation

Creating a New Report

1. From the Navigation Pane, select **Automation Center > Reports**. The Report List screen displays.
2. Click **New**. The New Report screen displays.



Name:	
Visible to:	Me
Type:	Bar chart
Table:	Activity [ops_exec]
Group by:	Agent
Stacked Field:	-- None --
Sum Field:	-- None --
Chart size:	Large
Other threshold	System Default

3. Using the field descriptions provided below as a guide, complete the fields as needed.
4. Click the **Save** button to save the report.

Report Field Descriptions

Field Name	Description
Name	Name of this report.
Visible to:	Defines who has access to this report. Options: • Me - Only the user who created the report can access the report. • Everyone - All users assigned the report_global role can access the report. • Other group names - All users assigned the report_group role can access the report.

Type	Format of the report. The option you select in this field determines what additional fields display in the second column of the report screen. Options: - Pie chart - A circular graph, useful when comparing sections. - Bar chart - A chart of color-coded bars. Can be displayed horizontally or vertically. - List - A list of items that match the report parameters. - Box chart - A graphical representation of groups of numerical data via five-number summaries. - Calendar - Used for date-based events, displays a calendar showing events that match the report parameters. - Control chart - A chart used to determine whether a given process is in a state of statistical control. - Histogram - Used in statistics, a histogram uses bars to show the distribution of data. - Horizontal bar - See Bar Chart above. - Line chart - A graph showing information as a series of data points. - Pareto chart - A combination chart showing bars (individual values) and a line graph (cumulative totals) - Trend chart - Used to show trends in data over time. - Trendbox chart - A box chart trended by a secondary value over a period. - Vertical bar - See Bar Chart above.
Table	Opswise table being used to create the report.
Group by	A field within the table that will be used to sort the report.
Stacked Field	For bar charts, allows you to specify a second field for displaying additional color-coded information. For example, you might display a bar chart showing today's system activity sorted by agent, additionally sorted by color-coded status.
Sum Field	For bar charts, control charts, histograms, line charts, trend charts, and trendbox charts, allows you to select a field into which task instance data will be summarized.
Chart size	For bar charts, Pareto charts, and trend charts, allows you to specify the chart size.
Other threshold	For bar charts, Pareto charts, and trend charts, allows you to limit the number of bars in your chart.
Columns (Available and Selected)	For lists of data, specifies fields from the table that should appear as columns in the list report. if you leave the Selected section blank, the system uses a hard-coded default selection.
Measured Field	For box charts, allows you to choose a field from the record that is used as one measurement of the data.
Calendar Field	For Calendars, allows you to choose a field from the table to be displayed on one or more calendar dates.
Trend Field	For control charts, histograms, line charts, trend charts, and trendbox charts, allows you to specify the field being monitored and the period of the trend.
Filter and Order	Allows you to create selection parameters for the report.
Run Report button	Generates the report and displays the results on the current screen.
Save button	Saves the report and adds it to the Reports list.
Insert button	Copies the current report and adds it to your reports list. Make sure you enter a new name in the Name field before clicking this button or it will overwrite the previous report.
Delete button	Deletes the current report.
Make Gauge button	Creates a gauge using the report parameters. You can then import the gauge into your home page or dashboard.
Schedule button	Opens a page that allows you to schedule the generation and distribution of this report.

Viewing, Running, Printing an Existing Report

1. Select **Reports** from the Navigation Pane. The Reports page displays.
2. Scroll down and click on the report you want to view and run. The report specifications display. The lower pane displays the results, if any, of the last time the report was run.
3. If necessary, you can make changes to the report specifications. To run the report, click **Run Report**.
4. To print the report, click the printer icon on the right end of the title bar. A preview screen generates. Select **Click to Print** to print the report.

Modifying a Report

1. Display the report you want to modify.
2. Make your changes and click **Update**.

Copying a Report

1. Display the report you want to copy.
2. Give the report a new name, specify any changes, and click **Insert**.

Deleting a Report

1. Display the report you want to delete.
2. Click **Delete**.

Creating a Gauge from a Report

You can create a gauge from your report and, optionally, add it to your home page or dashboard. A gauge is a "live" report that is updated according to the refresh setting on the home page or dashboard.

1. Display the report you want to make into a gauge.
2. Click **Make Gauge**.
3. Refer to Customizing Your Home Page for information about adding a gauge to a home page.

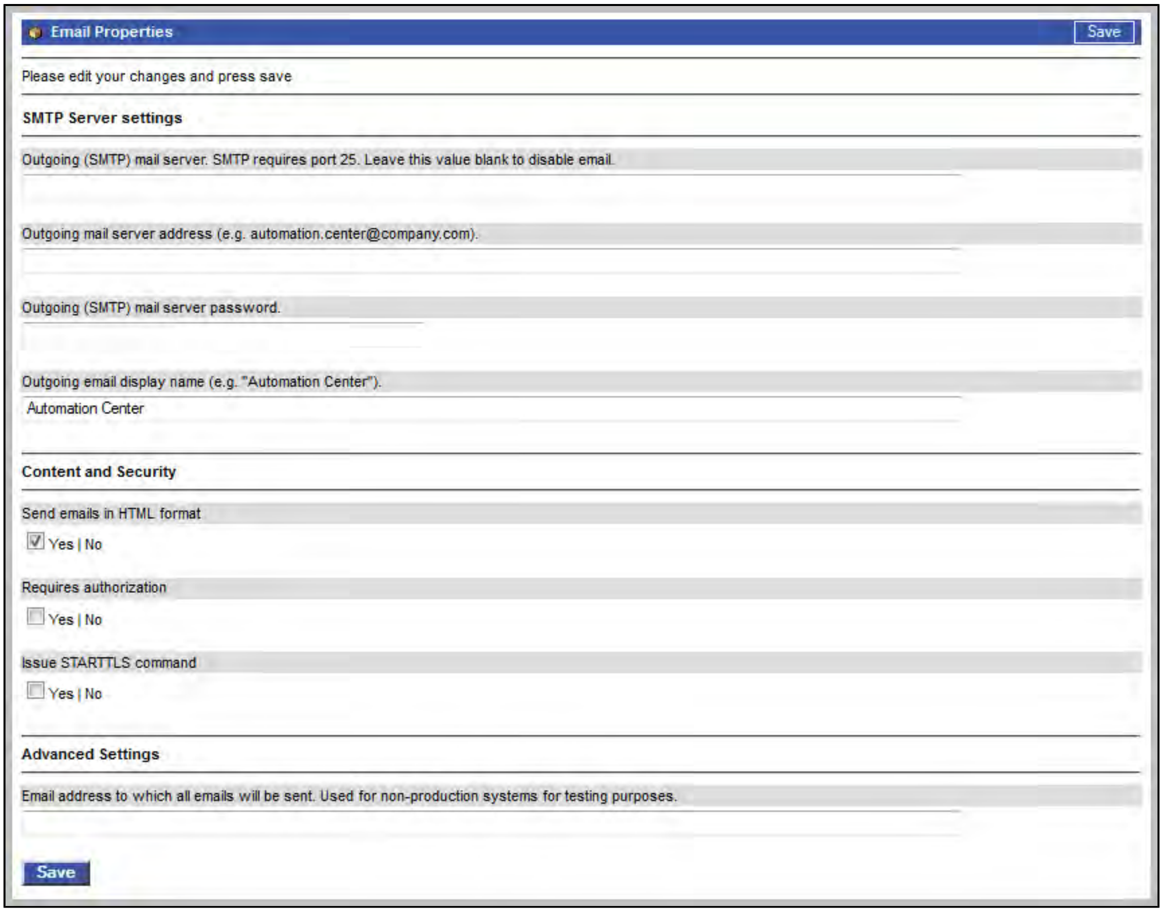
Scheduling Automatic Report Distribution

These instructions show you how to set up a report to be run and distributed to an emailing list on specific dates and times.

Setting Up the Email Properties

Before you can schedule automatic report distribution, you must first set up the email server you want to use.

1. From the navigation pane, select **Automation Center Administration > Configuration > Report Email Properties**.
2. Using the field descriptions provided on the screen, complete the fields as needed and click **Submit**.



Email Properties

Please edit your changes and press save

SMTP Server settings

Outgoing (SMTP) mail server. SMTP requires port 25. Leave this value blank to disable email.

Outgoing mail server address (e.g. automation.center@company.com).

Outgoing (SMTP) mail server password.

Outgoing email display name (e.g. "Automation Center").

Automation Center

Content and Security

Send emails in HTML format

☑ Yes | No

Requires authorization

□ Yes | No

Issue STARTTLS command

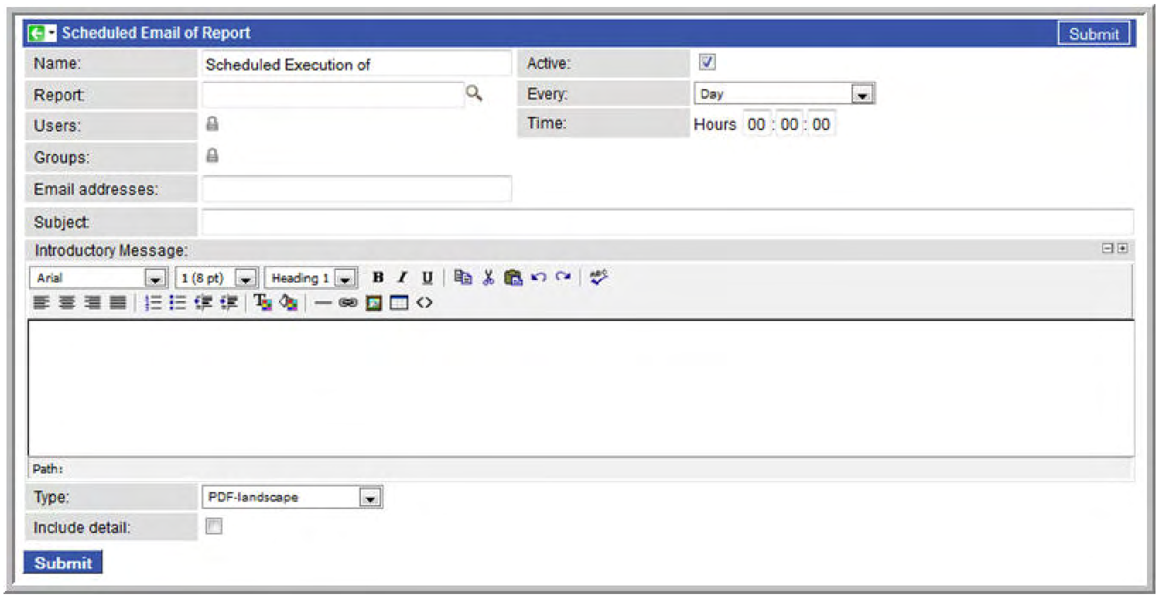
□ Yes | No

Advanced Settings

Email address to which all emails will be sent. Used for non-production systems for testing purposes.

Scheduling the Report

1. Display the report you want to schedule and click the **Schedule** button. Or, from the navigation pane, select **Automation Center > Scheduled Report Emails**, click **New**, and browse for the report you want to schedule.



Scheduled Email of Report

Name:	Scheduled Execution of	Active:	☑
Report:		Every:	Day
Users:		Time:	Hours 00 : 00 : 00
Groups:			
Email addresses:			
Subject:			

Introductory Message:

Type:	PDF-landscape
Include detail:	□

2. Use the field descriptions below for guidance to complete the fields.
3. Click **Submit**.

Scheduled Email of Report Field Descriptions

Field Name	Description
Name	Name of the report schedule.
Active	Indicates whether the schedule is active or not. Only active schedules are executed.
Report	Name of the report being scheduled. Click the magnifying glass to browse for a report.
Every	Specification for when the report should be run and distributed. Options: • Day • Week • Month
Users	See Adding Users.
Day (of week)	If Every = Week, specify the day of the week when the report should be run and distributed. Options: Monday through Sunday.
Day (of month)	If Every = Month, specify the numeric day of the month when the report should be run and distributed. Options: 1 through 31.
Time	In conjunction with the Every field, specifies a time when the report should be run and distributed.
Groups	See Adding Groups.
Email Addresses	Comma-separated list of email addresses for those receiving this report.
Subject	Subject line of the email.
Introductory Message	Message that will go in the body of the email. You can format the message using the standard formatting options provided. You can also specify font color T, background color, horizontal rule —, web link, images, tables. You can also toggle between wysiwyg and html source <>.
Type	Output type for the report. Options: PDF landscape, PDF, Excel, CSV (comma-separated values).
Include detail	For PDF list reports, includes any additional details available for the record type(s).

Opswise Activity Table (ops_exec)

When you create an Activity report, you only select data from the Activity (ops_exec) table. This table contains all data about executed task instances.

Field Name	Description
Agent	For agent-based tasks, the name of the agent.
Attempt	A counter that keeps track of the number of times this task instance was attempted.
Average Estimated End Time	Task instance only; system-supplied.
CPU Time	Task instance only; system-supplied. The amount of CPU time the task took to run.

Calendar	If the user selects **Special Restriction**, the calendar defines what the Holidays or Non Business days are. Type in a calendar name or click the magnifying glass to browse for an existing calendar or to add a new calendar. To display details about the calendar specified in this field, hover over the paper icon.
Class	The type of task instance, such as Sleep task instance or Workflow task instance.
Created	The date and time when the task instance was created.
Created by	The user ID of the user who created the task.
Credentials(credentials)	Login credentials Opswise Automation Center will use to access the remote machine. They must be provided either in the resource definition or task definition, or both. Credentials provided in the task definition override any credentials provided in the agent resource definition. See Credentials.
Credentials(credentials_var)	The variable specified in the login credentials field, if enabled.
Credentials Variable	Optional. If enabled, the Credentials field (see above) converts from a reference field (where you browse and select a record) into a text field that allows you to enter a variable. Use the format: ${variable name}. The variable must be a supported type as described in Using Variables and Functions .
Current Retry Count	Task instance only; system-supplied. Displays only for a running task instance. Displays the current number of times the system has retried the task after it first went to failure status.
Duration	Task instance only; system-supplied. The amount of time the task took to run.
Duration in Seconds	The amount of time, in seconds, the task instance took to run.
Early Finish	If enabled, and if the task instance finishes before the time or period specified, the task instance is flagged as early. You can specify a time or duration to determine an early finish (see Early Finish Type below). To determine whether a task instance finished early, open the task instance and locate the Finished Early field; the field is checked if the instance finished before the specified time or did not last as long as expected. This field only appears on the task instance if the user added Early Finish specifications to the task definition.
Early Finish Duration	If Early Finish Type is Duration, use this to specify the shortest amount of time this task instance should take to run. You can specify any combination of hours, minutes, and seconds.
Early Finish Time	If Early Finish Type is Time, use this to specify the time before which the task finish time is considered early. That is, enter a time at which the task should still be running. Use hh:mm, 24-hour time.
Early Finish Type	If Early Finish is enabled, use this field to specify whether you want to use time (see Early Finish Time below) or duration (see Early Finish Duration below) to determine whether the task instance finished early.
End Time	Task instance only; system-supplied. The date and time the task instance completed.
Execution User	Task instance only; system-supplied. If the task was launched manually, the ID of the user who launched it.

Exit Code	The exit code, if any, returned by the process.
Finished Early	Task instance only; system-supplied. This field is flagged if the task finished earlier than the time specified in the Early Finish fields.
Finished Late	Task instance only; system-supplied. This field is flagged if the task finished later than the time or duration specified in the Late Finish fields.
Forced Finished	True or False. Indicates whether the task instance was force-finished.
Hold Reason	Information about why the task will be put on hold when it starts.
Hold on Start	If enabled, when the task is launched it appears in the Activity display with a status of Held. The task runs when the user releases it.
IO Other	Total input/output operations for this task.
IO Reads	Total input/output reads for this task.
IO Writes	Total input/output writes for this task.
Instance Name	Required. Name used within Opswise to identify this task. Up to 40 alphanumerics; variables supported. It is the user's responsibility to develop a workable naming scheme for tasks.
Instance Reference Id	Task instance only; system-supplied. The system increments this number each time the task is run.
Invoked by	Task instance only; system-supplied. How the task instance was launched. One of the following: - Trigger: (Trigger Name) - The instance was launched by the named trigger. - Workflow: (Workflow Name) - The instance was launched by the named workflow. - Manually Launched - The instance was launched by a user. To determine the name of the user: 1. From the Activity or All Task Instances screen, click the task instance name to open the record. 2. The Execution User field identifies the user who launched the task instance.
Late Finish	If enabled, and if the task instance finishes after the time or period specified, the task instance is flagged as late. You can specify a time or duration to determine a late finish (see Late Finish Type below). To determine whether a task instance finished late, open the task instance and locate the Finished Late field; the field is checked if the instance finished after the specified time or lasted longer than expected. This field only appears on the task instance if the user specified a Late Finish in the task definition.
Late Finish Duration	If Late Finish Type is Duration, use this to specify the longest amount of time this task instance should take to run. You can specify any combination of hours, minutes, and seconds.
Late Finish Time	If Late Finish Type is Time, use this to specify the time after which the task finish time is considered late. Use hh:mm, 24-hour time.

Late Finish Type	If Late Finish is enabled, use this field to specify whether you want to use time (see Late Finish Time below) or duration (see Late Finish Duration below) to determine whether the task instance finished late.
Late Start	If enabled, and if the task instance starts after the time specified in the Late Start Time field (see below), the task instance is flagged as late. To determine whether a task instance started late, open the task instance and locate the Started Late field; the field is checked if the instance started late.
Late Start Duration	If Late Start Type is Duration, use this to specify the longest amount of time this task instance can wait before starting. You can specify any combination of hours, minutes, and seconds.
Late Start Time	Time after which the task start time is considered late. Use hh:mm, 24-hour time
Late Start Type	Required if Late Start is enabled. Options are: • **Time** - Flag the task if it starts after the specified time. • **Duration** - Flag the task if it starts a certain amount of time after the programmed start time. The task must have a specific start time.
Longest Estimated End Time	Task instance only; system-supplied.
Maximum Retries	User-defined. The maximum number of times the system should retry this task after it has started and gone to a failed state.
Member of Groups	User-defined. Allows you to select one or more Opswise groups that this record definition belongs to. Click on the lock icon to unlock the field and select groups.
Memory Peak	The peak amount of memory used during the execution of this task instance.
Memory Used	The amount of memory used during the execution of this task instance.
Queued Time	Task instance only; system-supplied. The time the task was queued for processing.
Resources State	The current status of the resource being used by a task. The system uses the same process each time it launches a job on a resource and the resource goes through the same series of states: 1. INITIAL - The initial state. This is the default value at launch time. 2. REQUESTED - When the task requests the resources it needs, the resource state becomes Requested. 3. ACQUIRED - When all of the requested resources are available and acquired by the server, the resource state becomes Acquired. 4. RETURNED - When the task completes, the server returns the resources it was using, and the resource state becomes Returned.
Retry Indefinitely	User-defined. Enabled or disabled. Indicates whether the system should continue trying indefinitely to run this task. If you enable this field, it overrides any value placed in the Maximum Retries field (above).
Retry Interval (Seconds)	User-defined. The number of seconds between each retry.
Run Called	(Internal property.)
Run Criteria Run Time	Indicates that run-time run criteria was specified for that task.

Run Criteria Trigger Time	Indicates that run trigger run criteria was specified for that task.
Security Name	Task name.
Shortest Estimated End Time	Task instance only; system-supplied.
Start Time	Task instance only; system-supplied. The date and time the task started.
Started Late	Task instance only; system-supplied. This field is flagged if the task started later than the time specified in the Late Start fields.
Status	Task instance only; system-supplied. Provides additional information, if any, about the status of the task.
Status Description	Task instance only; system-supplied. Provides additional information, if any, about the status of the task.
Summary	User-supplied description of this record.
Sys id	Unique system identifier associated with a task instance.
Task	Required. Name used within Opswise to identify this task. Up to 40 alphanumerics; variables supported. It is the user's responsibility to develop a workable naming scheme for tasks.
Task Priority	Task instance only; the priority of this task instance, as set by the user via the Set Priority command. Options are: HIGH, MEDIUM, LOW.
Trigger	User-defined. Name used within Opswise to identify this trigger. It can contain a maximum of 40 alphanumeric characters. It is the user's responsibility to develop a workable naming scheme for triggers.
Type	Type of task instance.
Updated	Date and time this record was last updated.
Updated by	The user who last updated this record.
Updates	The number of updates that have been made to the task record.
User Estimated End Time	Task instance only; system-supplied. If the user entered information into the User Estimated Duration field in the task definition, the system uses this information to calculate an end time for the task instance, based on the date/time the task instance started.
Vertex Id	Each task within a workflow has a unique vertex ID, which distinguishes it from other tasks of the same name, if any.
Waited for Resources	Indicates that the task instance could not get resources immediately and went into a resource wait state.

Workflow Id	The name of the workflow, if appropriate.
Show Related Files	Not supported for reports.

Opswise Report Tables

Described below are the tables that you can use when creating reports.

Field Name	Description
Abort Action (ops_abort_action)	Contains details about Abort actions.
Abort Action (ops_abort_action_v)	Contains details about previous versions of Abort actions. New versions of Abort Action records are created when a task record is updated.
Action (ops_notification)	Contains details about all actions on the Actions menu (Abort actions, Email notifications, SNMP notifications, and Set Variable actions).
Action (ops_notification_v)	Contains details about previous versions of all actions on the Actions menu (Abort actions, Email notifications, SNMP notifications, and Set Variable actions). New versions of Action records are created when a task record is updated.
Activity (ops_exec)	Task instance activity (running tasks).
Agent (ops_agent)	Displays a list of Agents.
Agent Cluster (ops_agent_cluster)	Contains details about Agent Clusters.
Agent Cluster (ops_agent_cluster_v)	Contains details about previous versions of Agent Clusters.
Agent Clusters (ops_bundle_agent_cluster_join)	Shows relationship information between Agent Clusters and Bundles, that is, which agent clusters belong to which bundles.
Agent Mapping (ops_agent_mapping)	Shows all the agents connected to one or more Promotion Targets (as retrieved using the **Refresh Target Agents** button).
Agents In Cluster (ops_unix_agent_cluster_join)	Shows relationship information between Unix agents and Unix agent clusters, that is, which agents belong to which clusters.
Agents In Cluster (ops_unix_agent_cluster_join_v)	Shows previous versions of relationship information between Unix agents and Unix agent clusters.
Agents In Cluster (ops_windows_agent_cluster_join)	Shows relationship information between Windows agents and Windows agent clusters, that is, which agents belong to which clusters.
Agents In Cluster (ops_windows_agent_cluster_join_v)	Shows previous versions of relationship information between Windows agents and Windows agent clusters.
Application (ops_application)	Shows a list of Application resources.
Application Control task (ops_task_application_control)	Contains details about Application Control tasks.
Application Control task instance (ops_exec_application_control)	Contains details about Application Control task instances.
Application Control task (ops_task_application_control_v)	Shows previous versions of Application Control tasks.
Application Monitor trigger (ops_trigger_appl_monitor)	Contains details about Application Monitor triggers.
Application Monitor trigger version (ops_trigger_appl_monitor_v)	Contains details about previous versions of Application Monitor triggers.
Application Version (ops_application_v)	Contains details about previous versions of Application resources.
Applications (ops_bundle_application_join)	Shows relationship information between Application resources and Bundles, that is, which Application resources belong to which bundles.

Audit Record (ops_audit)	Contains details of events being written to the Audit history.
Backup (ops_backup)	Contains Backup and Purge records.
Bundle (ops_bundle)	Contains all Bundles records.
Calendar (ops_calendar)	Contains details about Calendar records.
Calendar Custom Days (ops_cal_cust_join)	Contains details about which Custom Days are associated with which Calendar records.
Calendar Custom Days (ops_cal_cust_join_v)	Contains previous versions of the association between Custom Days and Calendar records.
Calendar Version (ops_calendar_v)	Contains previous versions of Calendar records.
Cluster Nodes (ops_cluster_node)	Provides details about Opswise core processors (cluster nodes).
Cluster Notification (ops_cluster_notification)	Contains Email and SNMP notification records associated with the Core Processor (cluster node).
Connector (ops_connector)	Contains Connector records.
Connector Notification (ops_connector_notifications)	Contains notification records associated with Connectors .
Credentials (ops_bundle_credentials_join)	Contains relationship information between Credentials and Bundles, that is, which Credential records belong to which bundles.
Credentials (ops_credentials)	Login credentials used by Opswise to access remote machines.
Credentials Version (ops_credentials_v)	Contains previous versions of Credentials records.
Cron Trigger (ops_trigger_cron)	Contains details about Cron trigger records.
Cron Trigger Version (ops_trigger_cron_v)	Contains previous versions of Cron trigger records.
Currently In Use By (ops_resource_usage)	Contains details about Virtual resource usage, as displayed in the Currently In Use By tab.
Custom Day Version (ops_custom_day)	Contains previous versions of Custom Days records.
Custom Days (ops_custom_day)	Contains details about defined Custom Days.
Database Connection (ops_database_connection)	Contains details about Database Connections defined in the Opswise database.
Database Connection Version (ops_database_connection_v)	Contains previous versions of Database Connections records.
Database Connections (ops_bundle_db_cntn_join)	Contains information about the relationship between Database Connections and Bundles, that is, which Database Connections belong to which Bundles.
Email Connection (ops_connect_email)	Contains details about Email Connection resources.
Email Connection Version (ops_email_connection_v)	Contains previous versions of Email Connection records.
Email Connections (ops_bundle_email_cntn_join)	Contains information about the relationship between Email Connection and Bundles, that is, which Email Connections belong to which Bundles.
Email Notification (ops_email_cluster_notification)	Contains Email Notification records associated with the Core Processor (cluster node).
Email Notification (ops_email_conn_notification)	Contains Email Notification records associated with a Connector.
Email Notification (ops_email_notification)	Contains details about Email Notifications associated with tasks.
Email Notification (ops_email_notification_v)	Contains previous versions of Email Notifications. Note that a new version is created only when the task is updated.

Email Task (ops_task_email)	Contains details about Email tasks.
Email Task Instance (ops_exec_email)	Contains details about Email task instances.
Email Task Version (ops_task_email_v)	Contains previous versions of Email task records.
Email Template (ops_email_template)	Contains details about Email templates.
Email Template Version (ops_email_template_v)	Contains previous versions of Email templates records.
Email Templates (ops_bundle_email_tmplt_join)	Contains relationship information between Email templates and Bundles, that is, which Email Templates belong to which Bundles.
File Monitor (ops_task_file_monitor)	Contains details about File Monitor tasks.
File Monitor Instance (ops_exec_file_monitor)	Contains details about File Monitor task instances.
File Monitor Trigger (ops_trigger_fm)	Contains details about File Monitor triggers.
File Monitor Trigger Version (ops_trigger_fm_v)	Contains previous versions of File Monitor trigger records.
File Monitor Version (ops_task_file_monitor_v)	Contains previous versions of File Monitor task records.
File Transfer Task (ops_task_ftp)	Contains details about File Transfer tasks.
File Transfer Task Instance (ops_exec_ftp)	Contains details about File Transfer task instances.
File Transfer Task Version (ops_task_ftp_v)	Contains previous versions of File transfer task records.
Forecasts (ops_trigger_forecast)	Contains details about trigger forecasts.
FTP File Monitor (ops_task_ftp_file_monitor)	Contains details about FTP File Monitor tasks.
FTP File Monitor Instance (ops_exec_ftp_file_monitor)	Contains details about FTP File Monitor task instances.
FTP File Monitor Version (ops_task_ftp_file_monitor_v)	Contains previous versions of FTP File Monitor task records.
Group (sys_user_group)	Contains details about Opswise Security Groups.
Group Member (sys_user_grmember)	Contains relationship information between Opswise Security Groups and Opswise Users; that is, which Users belong to which Groups.
Group Role (sys_group_has_role)	Contains relationship information between Opswise Security Groups and Roles; that is, which Groups have been assigned which Roles.
History (ops_history)	Contains a history of task activity.
Indesca Agent (ops_indesca_agent)	Contains details about Indesca agents.
Indesca Agent Mapping (ops_agent_mapping_indesca)	Shows the Agent mapping specifications between local Indesca agents and Indesca agents on a Promotion Target (as retrieved using the **Refresh Target Agents** button).
Indesca Task (ops_task_indesca)	Contains details about Indesca tasks.
Indesca Task Instance (ops_exec_indesca)	Contains details about Indesca task instances.
Indesca Task Version (ops_task_indesca_v)	Contains previous versions of Indesca task records.
JobStep Files Data (ops_exec_zos_files)	Contains details about jobsteps in a z/OS task.
Label (label)	Contains details about currently displayed navigation pane labels.

Label entry (label_entry)	Contains details about all navigation pane labels, including those that have been deleted.
Label history (label_history)	Contains the event history for navigation pane label records, including creation, updates, and deletions.
Linux/Unix Agent (ops_unix_agent)	Contains details about Linux/Unix agent resources.
Linux/Unix Agent Cluster (ops_unix_agent_cluster)	Contains details about Linux/Unix agent clusters.
Linux/Unix Agent Cluster (ops_unix_agent_cluster_v)	Contains previous versions of Linux/Unix cluster records.
Linux/Unix Agent Mapping (ops_agent_mapping_unix)	Shows the mapping specifications between local Linux/Unix agents and Linux/Unix agents on a Promotion Target (as retrieved using the **Refresh Target Agents** button).
Linux/Unix Task (ops_task_unix)	Contains details about Linux/Unix tasks.
Linux/Unix Task Instance (ops_exec_unix)	Contains details about Linux/Unix task instances.
Linux/Unix Task Version (ops_task_unix_v)	Contains previous versions of Linux/Unix task records.
Manual Task (ops_task_manual)	Contains details about Manual tasks.
Manual Task Instance (ops_exec_manual)	Contains details about Manual task instances.
Manual Task Version (ops_task_manual_v)	Contains previous versions of Manual task records.
Manual Trigger (ops_trigger_manual)	Contains Manual trigger records.
Manual Trigger Version (ops_trigger_manual_v)	Contains previous versions of Manual trigger records.
Note (ops_note)	Contains details about Notes attached to Opswise records.
Note (ops_note_v)	Contains previous versions of Notes records.
Opswise Groups (ops_generic_group)	Contains details about Opswise Groups.
Opswise Group Version (ops_generic_group_v)	Contains previous versions of Group records.
Opswise Groups (ops_bundle_generic_group_join)	Contains relationship information between Opswise Groups and Bundles; that is, which Groups belong to which Bundles.
Opswise Permissions (ops_permission)	Contains details about Opswise Permissions assigned to Opswise Users and Opswise Security Groups.
Opswise Properties (ops_config)	Contains Opswise Configuration Properties.
Output (ops_exec_output)	Contains any output (such as STDOUT) attached to task instances.
Outstanding Requests (ops_resource_order)	Contains any outstanding requests for a resource by a task instance.
Promotion History (ops_promotion_history)	Contains a list of Bundles that have been promoted into the current database.
Promotion History Item (ops_promotion_history_item)	Contains a list of records that have been promoted into the current database. If a record has been promoted more than once, each version is listed separately.
Promotion Target (ops_bundle_target)	Contains Promotion Target records.
Restart Confirmation (ops_exec_ zos_confirm)	Contains details about any restart confirmations performed on z/OS tasks.
SAP Connection (ops_sap_connection)	Contains SAP Connection records.
SAP Connection Version (ops_sap_connection_v)	Contains previous versions of SAP Connection records.

SAP Connections (ops_bundle_sap_cntn_join)	Contains relationship information between SAP Connection records and Bundles; that is, which SAP Connection records are in which Bundles.
SAP Task (ops_task_sap)	Contains SAP task records.
SAP Task Instance (ops_exec_sap)	Contains SAP task instance records.
SAP Task Version (ops_task_sap_v)	Contains previous versions of SAP task records.
Script (ops_script)	Contains Script records.
Script (ops_script_v)	Contains previous versions of Script records.
Script (ops_bundle_script_join)	Contains relationship information between Script records and Bundles; that is, which Scripts belong to which Bundles.
Set Variable (ops_variable_action)	Contains details about Set Variable actions.
Set Variable (ops_variable_action_v)	Contains previous versions of Set Variable actions.
Sleep Task (ops_task_sleep)	Contains details about Sleep tasks.
Sleep Task Instance (ops_exec_sleep)	Contains details about Sleep task instances.
Sleep Task Version (ops_task_sleep_v)	Contains previous versions of Sleep tasks records.
SNMP Manager (ops_snmp_connection)	Contains SNMP Connection records.
SNMP Manager Version (ops_snmp_connection_v)	Contains previous versions of SNMP Connection records.
SNMP Managers (ops_bundle_snmp_cntn_join)	Contains relationship information between SNMP Connection records and Bundles; that is, which SNMP Connections belong to which Bundles.
SNMP Notification (ops_snmp_cluster_notification)	Contains SNMP notifications defined for Opswise Core Processors (Cluster Nodes) .
SNMP Notification (ops_snmp_conn_notification)	Contains SNMP notifications defined for Connectors .
SNMP Notification (ops_snmp_notification)	Contains SNMP notifications defined for Tasks.
SNMP Notification (ops_snmp_notification_v)	Contains previous versions of task-related SNMP notifications. Note that a new version is created only when the task is updated.
SQL Result Set (ops_sql_results)	Contains output from SQL tasks.
SQL Task (ops_task_sql)	Contains details about SQL tasks.
SQL Task Instance (ops_exec_sql)	Contains details about SQL task instances.
SQL Task Version (ops_task_sql_v)	Contains previous versions of SQL tasks records.
SQL Warning Set (ops_sql_warnings)	Contains warnings returned by executed SQL statements.
Stored Procedure Parameters (ops_stored_proc_param)	Contains Parameter records associated with Stored Procedure Task tasks.
Stored Procedure Parameters (ops_stored_proc_param_v)	Contains previous versions of Parameter records associated with Stored Procedure Task tasks. Note that versions are only created when the task is updated.
Stored Procedure Task (ops_task_stored_proc)	Contains details about Stored Procedure tasks.
Stored Procedure Task Instance (ops_exec_stored_proc)	Contains details about Stored Procedure task instances.
Stored Procedure Task Version (ops_task_stored_proc_v)	Contains previous versions of Stored Procedure tasks records.

System Monitor (ops_task_system_monitor)	Contains System Monitor task records.
System Monitor Task Instance (ops_exec_system_monitor)	Contains System Monitor task task instance records.
System Monitor Version (ops_task_system_monitor_v)	Contains previous versions of System Monitor task task records.
Task (ops_task)	Contains details about tasks of every type, along with associated Task Instance information.
Task Instance Run Criteria (ops_exec_run_criteria)	Contains run criteria information for task instances within a Workflow.
Task Monitor (ops_task_monitor)	Contains details about Task Monitor tasks.
Task Monitor Instance (ops_exec_monitor)	Contains details about Task Monitor task instances.
Task Monitor Trigger (ops_trigger_tm)	Contains details about Task Monitor triggers.
Task Monitor Trigger Version (ops_trigger_tm_v)	Contains previous versions of Task Monitor trigger records.
Task Monitor Version (ops_task_monitor_v)	Contains previous versions of Task Monitor task records.
Task Run Criteria (ops_task_run_criteria)	Contains run criteria information for tasks within Workflows.
Task Run Criteria (ops_task_run_criteria_v)	Contains previous versions of run criteria information for tasks within Workflow. Note that new versions are created only when the Workflow task is updated.
Task Version (ops_task_v)	Contains previous versions of all task records.
Task Virtual Resources (ops_task_to_resource)	Contains relationship information between Virtual resources and tasks; that is, which tasks are assigned to which Virtual Resources.
Task Virtual Resources (ops_task_to_resource_v)	Contains previous versions of relationship information between Virtual resources and tasks.
Tasks (ops_bundle_task_join)	Contains relationship information between Task records and Bundles; that is, which Tasks are in which Bundles.
Temporary Trigger (ops_trigger_temp)	Contains details about Temporary triggers.
Temporary Trigger (ops_trigger_temp_v)	Contains previous versions of Temporary trigger records.
Time Trigger (ops_trigger_time)	Contains details about Time triggers.
Time Trigger Version (ops_trigger_time_v)	Contains previous versions of Time trigger records.
Trigger (ops_trigger)	Contains details about triggers of every type.
Trigger Version (ops_trigger_v)	Contains previous versions of trigger records.
Triggers (ops_bundle_trigger_join)	Contains relationship information between Trigger records and Bundles; that is, which Triggers are in which Bundles.
User (sys_user)	Contains details about User records.
User Role (sys_user_has_role)	Contains details about User and Role records, including which Users have which Roles.
Variable Version (ops_variable_v)	Contains previous versions of Global variables.
Variables (ops_bundle_variable_join)	Contains relationship information between Global variables and Bundles; that is, which Global variables belong to which Bundles.
Variables (ops_local_variable)	Contains details about task and trigger variables (also called local variables), entered into the Variables tab on a task or trigger record.
Variables (ops_local_variable_v)	Contains previous versions of Local variables associated with tasks or triggers. Note that new version records are only created when the task or trigger is updated.

Variables (ops_variable)	Contains details about Global variables, entered by selecting Variables from the Navigation pane.
Virtual Resource (ops_virtual_resource)	Contains details about Virtual resource records.
Virtual Resources (ops_bundle_resource_join)	Contains relationship information between Virtual resources and Bundles; that is, which Virtual resources belong to which Bundles.
Windows Agent (ops_windows_agent)	Contains details about Windows agents.
Windows Agent Cluster (ops_windows_agent_cluster)	Contains details about Windows agent clusters.
Windows Agent Cluster Version (ops_windows_agent_cluster_v)	Contains previous versions of Windows Agent Cluster records.
Windows Agent Mapping (ops_agent_mapping_windows)	Shows the mapping specifications between local Windows agents and Windows agents on a Promotion Target (as retrieved using the **Refresh Target Agents** button).
Windows Task (ops_task_windows)	Contains details about Windows tasks.
Windows Task Instance (ops_exec_windows)	Contains details about Windows task instances.
Windows Task Version (ops_task_windows_v)	Contains previous versions of Windows task records.
Workflow Task (ops_task_workflow)	Contains details about Workflow tasks.
Workflow Task Edges (ops_task_workflow_edge)	Contains information about the conditions specified between tasks in workflows.
Workflow Task Edges (ops_task_workflow_edge_v)	Contains previous versions of information about the conditions specified among tasks in workflows. Note that new version records are only created when the Workflow task is updated.
Workflow Task Instance (ops_exec_workflow)	Contains details about Workflow task instances.
Workflow Task Instance Edges (ops_exec_workflow_edge)	Contains information about the conditions specified between task instances within workflows.
Workflow Task Instance Vertices (ops_exec_workflow_vertex)	Contains relationship information between workflows instances and task instances; that is, which tasks are running in which workflows.
Workflow Task Version (ops_task_workflow_v)	Contains previous versions of Workflow task records.
Workflow Task Vertices (ops_task_workflow_vertex)	Contains relationship information between tasks and workflows; that is, which tasks are in which workflows.
Workflow Task Vertices (ops_task_workflow_vertex_v)	Contains previous versions of the relationship between tasks and workflows. Note that new version records are created only when the Workflow task is updated.
z/OS Agent (ops_zos_agent)	Contains details about z/OS agents.
z/OS Agent Mapping (ops_agent_mapping_zos)	Shows the mapping specifications between local z/OS agents and z/OS agents on a Promotion Target (as retrieved using the **Refresh Target Agents** button).
z/OS Restartable JobSteps (ops_exec_zos_jobstepsui)	Contains details about restartable job steps in a z/OS task.
z/OS Restartable JobSteps ND (ops_exec_zos_jobsteps)	Contains details about restartable job steps in a z/OS task.
z/OS Task (ops_task_zos)	Contains details about z/OS tasks.
z/OS Task Instance (ops_exec_zos)	Contains details about z/OS task instances.
z/OS Task Version (ops_task_zos_v)	Contains previous versions of z/OS task records.

Audits

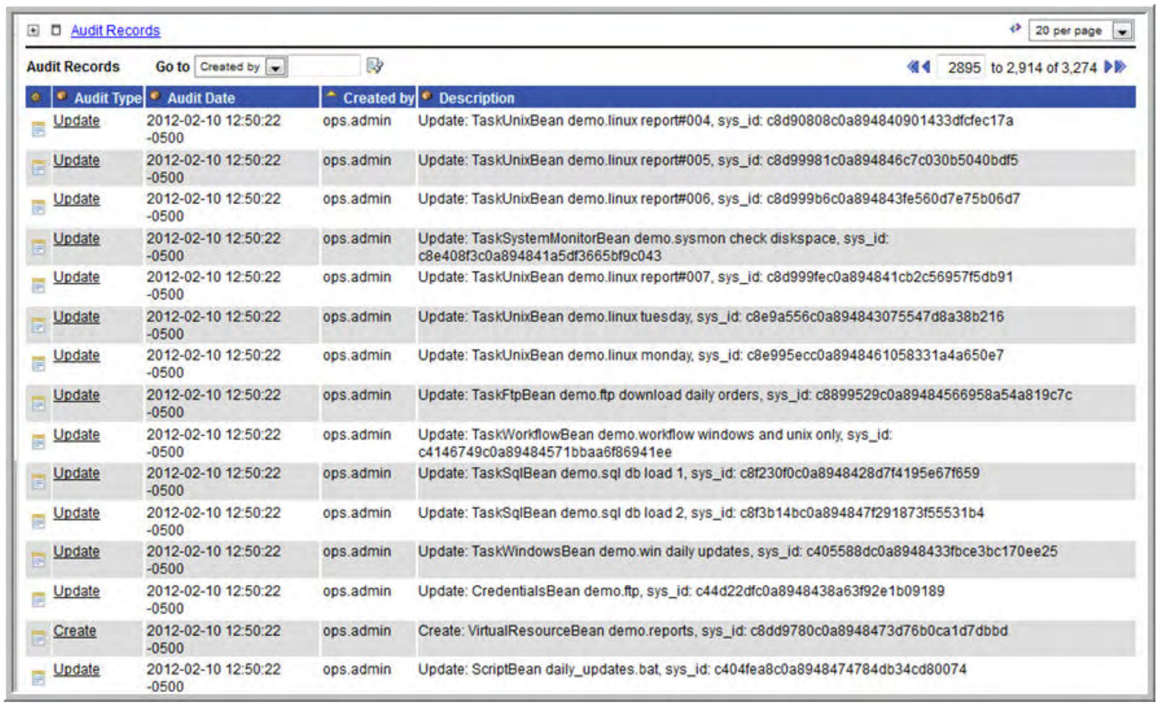
Overview

The Opswise audit function maintains a detailed record of all user interactions with the system, including before and after images related to any change and a description of the differences. Audit records are written when the user performs any of the following actions:

- Creates a new record
- Updates a record
- Deletes a record
- Issues a command against a task or task instance (Launch, Skip, Re-run, Hold, Release, Cancel, and Force Finish)

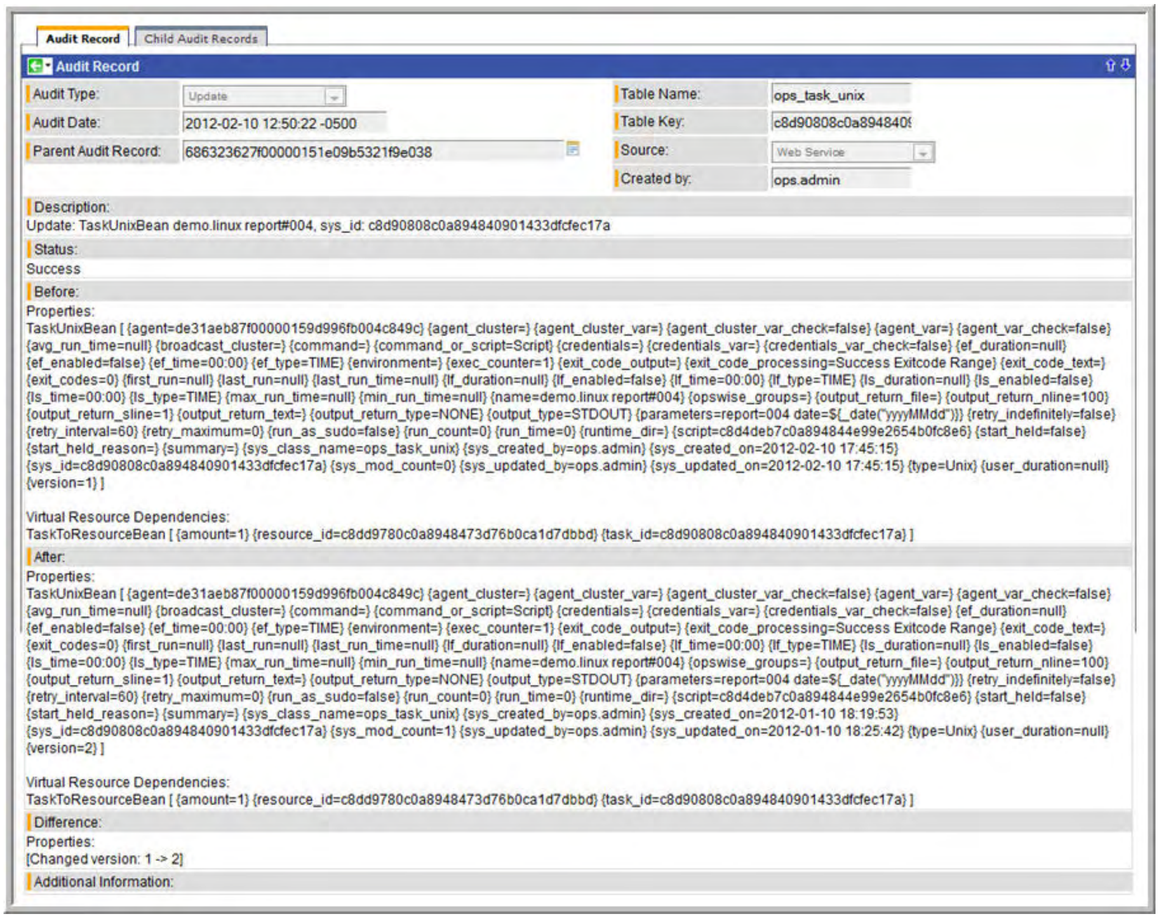
Displaying the Audit Table

1. From the navigation pane, select **Automation Center Administration > Security >**. The Audit Records screen displays the first page of audit activity.



	Audit Type	Audit Date	Created by	Description
	Update	2012-02-10 12:50:22 -0500	ops.admin	Update: TaskUnixBean demo.linux report#004, sys_id: c8d90808c0a894840901433dfcfec17a
	Update	2012-02-10 12:50:22 -0500	ops.admin	Update: TaskUnixBean demo.linux report#005, sys_id: c8d99981c0a894846c7c030b5040bdf5
	Update	2012-02-10 12:50:22 -0500	ops.admin	Update: TaskUnixBean demo.linux report#006, sys_id: c8d999b6c0a894843fe560d7e75b06d7
	Update	2012-02-10 12:50:22 -0500	ops.admin	Update: TaskSystemMonitorBean demo.sysmon check diskspace, sys_id: c8e408f3c0a894841a5df3665bf9c043
	Update	2012-02-10 12:50:22 -0500	ops.admin	Update: TaskUnixBean demo.linux report#007, sys_id: c8d999fec0a894841cb2c56957f5db91
	Update	2012-02-10 12:50:22 -0500	ops.admin	Update: TaskUnixBean demo.linux tuesday, sys_id: c8e9a556c0a894843075547d8a38b216
	Update	2012-02-10 12:50:22 -0500	ops.admin	Update: TaskUnixBean demo.linux monday, sys_id: c8e995ecc0a8948461058331a4a650e7
	Update	2012-02-10 12:50:22 -0500	ops.admin	Update: TaskFtpBean demo.ftp download daily orders, sys_id: c8899529c0a89484566958a54a819c7c
	Update	2012-02-10 12:50:22 -0500	ops.admin	Update: TaskWorkflowBean demo.workflow windows and unix only, sys_id: c4146749c0a89484571bbaa6f86941ee
	Update	2012-02-10 12:50:22 -0500	ops.admin	Update: TaskSqlBean demo.sql db load 1, sys_id: c8f230f0c0a8948428d7f4195e67f659
	Update	2012-02-10 12:50:22 -0500	ops.admin	Update: TaskSqlBean demo.sql db load 2, sys_id: c8f3b14bc0a894847f291873f55531b4
	Update	2012-02-10 12:50:22 -0500	ops.admin	Update: TaskWindowsBean demo.win daily updates, sys_id: c405588dc0a8948433fbce3bc170ee25
	Update	2012-02-10 12:50:22 -0500	ops.admin	Update: CredentialsBean demo.ftp, sys_id: c44d22dfc0a8948438a63f92e1b09189
	Create	2012-02-10 12:50:22 -0500	ops.admin	Create: VirtualResourceBean demo.reports, sys_id: c8dd9780c0a8948473d76b0ca1d7dbbd
	Update	2012-02-10 12:50:22 -0500	ops.admin	Update: ScriptBean daily_updates.bat, sys_id: c404fea8c0a8948474784db34cd80074

2. To change the record selection, modify the display filter. For example, you may want to display all Audit activity for the past month. For instructions, see Sorting and Filtering a List.
3. To display details about a particular audit record, click on the underlined field of the record (in the leftmost column).



Audit Type:	Update	Table Name:	ops_task_unix
Audit Date:	2012-02-10 12:50:22 -0500	Table Key:	c8d90808c0a894840(
Parent Audit Record:	686323627f00000151e09b5321f9e038	Source:	Web Service
		Created by:	ops.admin

Description:
Update: TaskUnixBean demo.linux report#004, sys_id: c8d90808c0a894840901433dfcfec17a

Status:
Success

Before:
Properties:
TaskUnixBean [{agent=de31aeb87f00000159d996fb004c849c} {agent_cluster=} {agent_cluster_var=} {agent_cluster_var_check=false} {agent_var=} {agent_var_check=false} {avg_run_time=null} {broadcast_cluster=} {command=} {command_or_script=Script} {credentials=} {credentials_var=} {credentials_var_check=false} {ef_duration=null} {ef_enabled=false} {ef_time=00:00} {ef_type=TIME} {environment=} {exec_counter=1} {exit_code_output=} {exit_code_processing=Success Exitcode Range} {exit_code_text=} {exit_codes=0} {first_run=null} {last_run=null} {last_run_time=null} {lf_duration=null} {lf_enabled=false} {lf_time=00:00} {lf_type=TIME} {ls_duration=null} {ls_enabled=false} {ls_time=00:00} {ls_type=TIME} {max_run_time=null} {min_run_time=null} {name=demo.linux report#004} {opswise_groups=} {output_return_file=} {output_return_nline=100} {output_return_sline=1} {output_return_text=} {output_return_type=NONE} {output_type=STDOUT} {parameters=report=004 date=${_date("yyyyMMdd")}} {retry_indefinitely=false} {retry_interval=60} {retry_maximum=0} {run_as_sudo=false} {run_count=0} {run_time=0} {runtime_dir=} {script=c8d4deb7c0a894844e99e2654b0fc8e6} {start_held=false} {start_held_reason=} {summary=} {sys_class_name=ops_task_unix} {sys_created_by=ops.admin} {sys_created_on=2012-02-10 17:45:15} {sys_id=c8d90808c0a894840901433dfcfec17a} {sys_mod_count=0} {sys_updated_by=ops.admin} {sys_updated_on=2012-02-10 17:45:15} {type=Unix} {user_duration=null} {version=1}]

Virtual Resource Dependencies:
TaskToResourceBean [{amount=1} {resource_id=c8dd9780c0a8948473d76b0ca1d7dbbd} {task_id=c8d90808c0a894840901433dfcfec17a}]

After:
Properties:
TaskUnixBean [{agent=de31aeb87f00000159d996fb004c849c} {agent_cluster=} {agent_cluster_var=} {agent_cluster_var_check=false} {agent_var=} {agent_var_check=false} {avg_run_time=null} {broadcast_cluster=} {command=} {command_or_script=Script} {credentials=} {credentials_var=} {credentials_var_check=false} {ef_duration=null} {ef_enabled=false} {ef_time=00:00} {ef_type=TIME} {environment=} {exec_counter=1} {exit_code_output=} {exit_code_processing=Success Exitcode Range} {exit_code_text=} {exit_codes=0} {first_run=null} {last_run=null} {last_run_time=null} {lf_duration=null} {lf_enabled=false} {lf_time=00:00} {lf_type=TIME} {ls_duration=null} {ls_enabled=false} {ls_time=00:00} {ls_type=TIME} {max_run_time=null} {min_run_time=null} {name=demo.linux report#004} {opswise_groups=} {output_return_file=} {output_return_nline=100} {output_return_sline=1} {output_return_text=} {output_return_type=NONE} {output_type=STDOUT} {parameters=report=004 date=${_date("yyyyMMdd")}} {retry_indefinitely=false} {retry_interval=60} {retry_maximum=0} {run_as_sudo=false} {run_count=0} {run_time=0} {runtime_dir=} {script=c8d4deb7c0a894844e99e2654b0fc8e6} {start_held=false} {start_held_reason=} {summary=} {sys_class_name=ops_task_unix} {sys_created_by=ops.admin} {sys_created_on=2012-01-10 18:19:53} {sys_id=c8d90808c0a894840901433dfcfec17a} {sys_mod_count=1} {sys_updated_by=ops.admin} {sys_updated_on=2012-01-10 18:25:42} {type=Unix} {user_duration=null} {version=2}]

Virtual Resource Dependencies:
TaskToResourceBean [{amount=1} {resource_id=c8dd9780c0a8948473d76b0ca1d7dbbd} {task_id=c8d90808c0a894840901433dfcfec17a}]

Difference:
Properties:
[Changed version: 1 -> 2]

Additional Information:

Managing Opswise Records

Opswise supports several features that allow you to manage records and control the amount of data in your database. These include the following:

- Record Versioning
- Bundling and Promoting Records
- Backing Up and Purging Data
- Exporting and Importing Records

Bundling and Promoting Records

Overview

The Opswise Bundling and Promoting features allow you to select and bundle a group of Opswise records and "promote" them from one Opswise server to another. For example, you can use these features when you create your workflows on a development installation then move them to a QA installation for testing. Once you are satisfied with the stability of the workflows, you can promote them to your production system.

The general process for bundling and promoting your data from a source server is:

1. On the source server, define one or more Bundle records.
2. On the source server, create a Promotion Target record for each target server.
3. Specify agent mappings between the source and target server.
4. Promote Bundles and/or individual records to the target.

These features use web services calls to communicate when you are promoting Bundles of records from one server to another.

Click hereto watch a video about Bundling and Promoting Records.

Creating Bundles

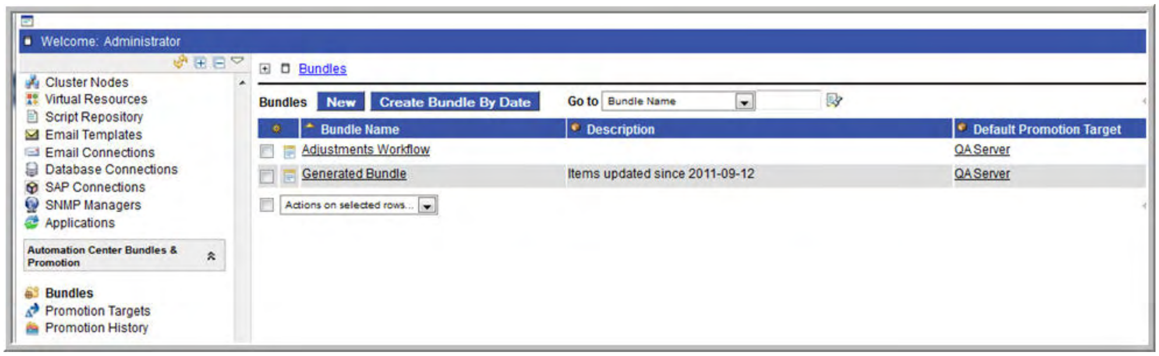
You can create Bundles manually by selecting records yourself or you can specify a date parameter that automatically selects all records added or changed on or after the date. Each procedure is described below.

Manually Defining a New Bundle

Your userID must have the ops_Bundle_admin role to use this feature.

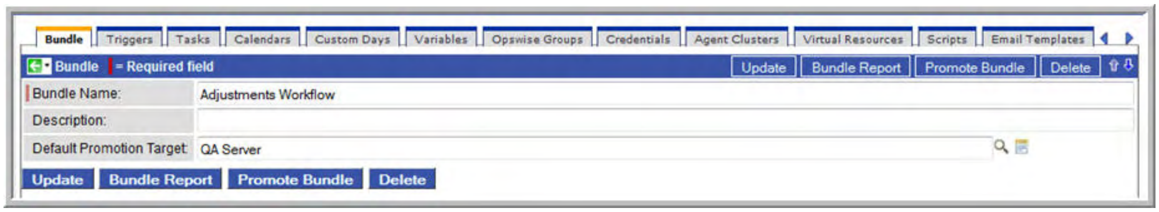
The procedure for defining a new Bundle involves creating the Bundle record, selecting which records you want to include, and selecting a target for promotion. The steps are described in detail below.

1. From the navigation pane, select **Automation Center Bundles & Promotion > Bundles.**



	Bundle Name	Description	Default Promotion Target
□	Adjustments Workflow		QA Server
□	Generated Bundle	Items updated since 2011-09-12	QA Server

2. Click the **New** button. The system displays the Bundle screen, as shown in the following example.



Bundle Name: Adjustments Workflow

Description:

Default Promotion Target: QA Server

3. Each Bundle record includes a separate tab for each type of record you may choose to include in the Bundle. Use the field descriptions below for guidance in filling out the fields. You must minimally specify a Bundle name and the target to which you will be promoting this Bundle.
4. Before selecting records to include in the Bundle, you must first create the Bundle record in the database. Click the down arrow and select **Save** or click **Submit**.
5. To select records for inclusion in this Bundle, select the tab for that record type. You can create a new record to add to the Bundle by selecting **New** and filling out the form. Or, you can select from existing records as follows:
    a. Click the **Edit** button. The system displays all records for which you have read permission.
    b. The records listed under Collection are existing records that do not already belong to this Bundle. The records listed under Has (Record Type) List are records that belong to this Bundle. You can filter the records listed under Collection as follows:
        - Click **Add Filter**. See To create a filter for information about how to construct a filter.
        - When you have defined the filter you want, click **Run Filter**. The Collection list now displays only those records that match the filter.
        - To remove filtering, click the **X** to the right of the filter, then click **Run Filter** again.
    c. To add to or remove records from the Has (Record Type) List:
        - To add a record to the list, double-click on the record in the Collection list. Or, use **CTRL-click** to select multiple records and click the right arrow ▶.
        - To remove a record from the list, double-click on the name in the Has (Record Type) List. Or, use **CTRL-click** to select multiple records and click the left arrow ◀.
          As you click on a record, the system displays details about it at the bottom of the form.
    d. When you are finished, click **Save**.
6. Repeat the above steps in the appropriate tabs for all records you want to add.

Bundle Field Descriptions

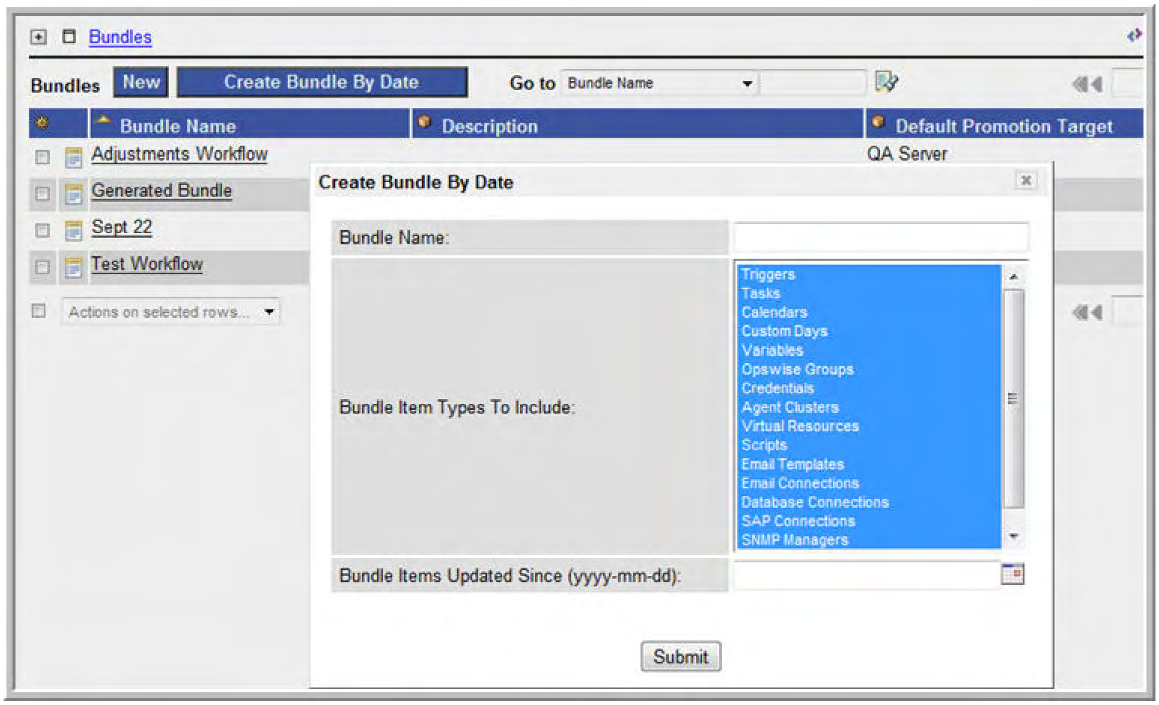
Field Name	Description
Bundle Name	Name of this record.
Description	User-supplied description of this record.
Default Promotion Target	Allows you to browse for and select a Promotion Target, which you defined using the Promotion Targets feature.
Submit button	Submits the new record to the database.
Update button	Saves updates to the record.
Bundle Report button	Allows you to generate a report about the current bundle. See Generating a Bundle Report.
Promote Bundle button	Allows you to promote this bundle to a target Opswise server. See Promoting a Bundle to a Target.
Delete button	Deletes the current record.
Record type tabs	Each tab shows the records of that type that belong to this Bundle. The **New** and **Edit** keys allow you to add records to the Bundle, as described in the above procedure.

Creating a Bundle Based on Date

Your userID must have the ops_Bundle_admin role to use this feature.

This feature allows you to Bundle all records created or updated on the current server on or since a specific date. The procedure involves selecting which records you want to include and specifying the date parameter. The steps are described in more detail below.

1. From the navigation pane, select **Automation Center Bundles & Promotion > Bundles.**
2. Click the **Create Bundle by Date** button. The system displays the Create Bundle by Date pop-up screen. By default, all record types are selected, as shown below:



3. You can leave the default all record types selected. Or, to deselect the group of records, click anywhere in the list. Use **CTRL-click** to select multiple record types.
4. To select the date parameter, click the calendar in the lower right corner. A calendar pops up.
5. Click the date you want applied to this Bundle. All records created or updated on or after that date will be included.
6. Click the **Submit** button. If any records qualified for inclusion in the Bundle, the Bundle is created and saved to the database. However, if no records qualified according to the specified date, the Bundle is not saved.

Adding Record(s) to a Bundle From Record Lists and Forms

At any time, you can add record(s) to a Bundle from lists of records or from the record itself, as described below.

To add a single record from a list of records:

1. Right click on the record you want to add.
2. Select **Add to Bundle**. The system displays a window, prompting you to select the Bundle from a drop-down list of Bundles.
3. Select the Bundle from the dropdown list and click **Submit**.

To add multiple records from a list of records:

1. Select the records you want to add by clicking the box to the left of the record name.
2. From the **Actions on selected rows** menu, select **Add to Bundle**. The system displays a window, prompting you to select the Bundle from a drop-down list of Bundles.
3. Select the Bundle from the dropdown list and click **Submit**.

To add the current record to a Bundle:

1. Open the record you want to add.
2. Click the down-arrow next to the record name or right-click on the record name. A menu appears.
3. Select **Add To Bundle**. The system displays a window, prompting you to select the Bundle from a drop-down list of Bundles.
4. Select the Bundle from the dropdown list and click **Submit**.

Displaying a Record's Bundles

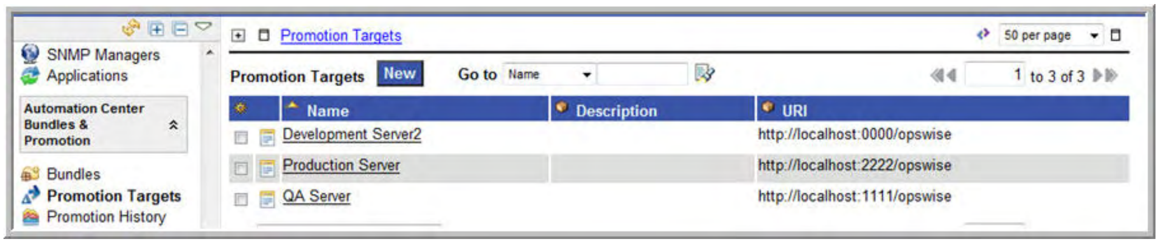
To find out what Bundles a record belongs to:

1. Open the record.
2. Move your cursor over the down-arrow next to the record name or right-click on the record title bar. A menu appears.
3. Select **View Bundles**. The system displays a list of Bundles to which the current record belongs. You can add a new Bundle on the fly, click on a Bundle name to view it, or click the browser's back button to return to the record.

Defining Promotion Targets

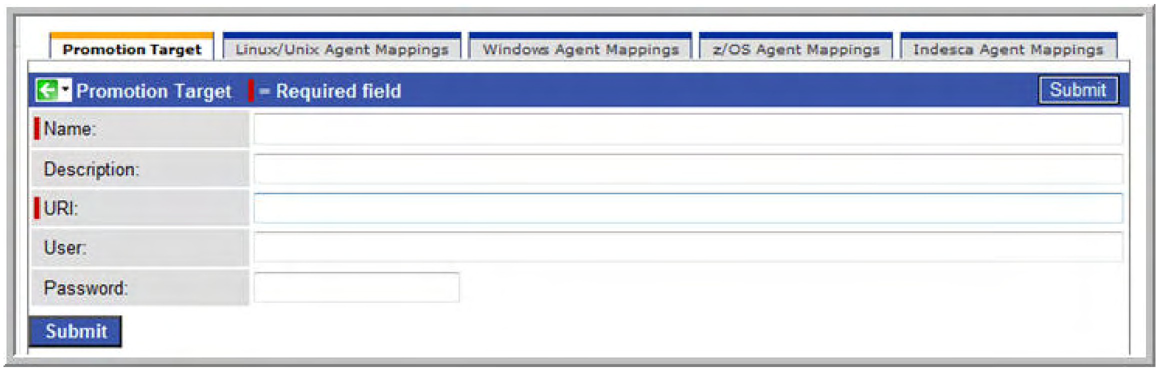
Before you can promote Bundles or individual records, you must identify and create a Promotion Target record(s) for the target machine(s). The Promotion Target record provides the Uniform Resource Identifier (URI) of the target server, along with the user name and password required to log on to Opswise on the target server.

1. From the navigation pane, select **Automation Center Bundles & Promotion > Promotion Targets**. The system displays a list of existing Promotion Targets.



Name	Description	URI
Development Server2		http://localhost:0000/opswise
Production Server		http://localhost:2222/opswise
QA Server		http://localhost:1111/opswise

2. To create a new record, click **New**. The system displays a blank Promotion Target form.



3. Using the field descriptions below for guidance, complete the form. Minimally, you must be provide a record name and URI.
4. Click **Submit** to save the record.

Promotion Target Field Descriptions

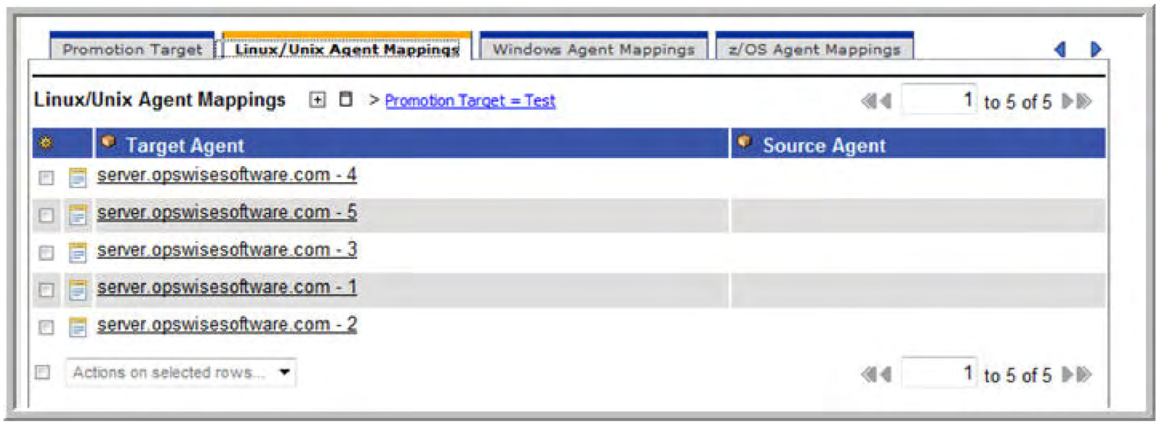
Field Name	Description
Name	Name of this record.
Description	User-supplied description of this record.
URI	Uniform Resource Identifier (URI) used to locate the target server. Similar to a URL.
User	Login ID on the target instance of Opswise.

Password	Login password on the target instance of Opswise.
Submit button	Submits the new record to the database.
Update button	Saves updates to the record.
Refresh Target Agents button	Accesses the specified Opswise server and fetches all agent records. For details, see Specify Agent Mapping .
Delete button	Deletes the current record.
Agent Mappings tabs	Each tab contains the agent mapping instructions between the source server and the target server. See Specify Agent Mapping

Specifying Agent Mapping

Because your source and target Opswise instances may not have the same agents, you must provide instructions to Opswise on how to map agents on the source machine to agents on the target machine. The process consists of instructing Opswise to fetch the list of agents on the target server and manually identifying how each agent should be mapped.

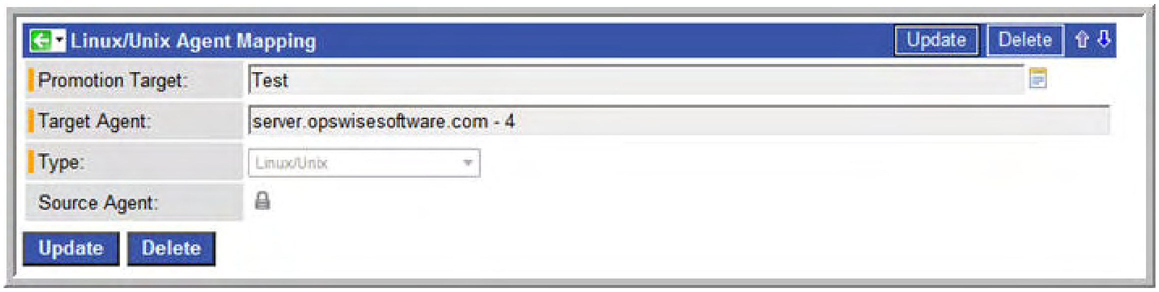
1. Open the Promotion Target record for which you want to specify the mapping.
2. Click **Refresh Target Agents**. If the Promotion Target record does not provide the user ID and password, the system prompts for them. If you want to override the ID and password from the Promotion Target record, click **Override User/Password** and type in the new information.
3. Click **Submit**. The system logs in to the target machine specified in the URI field, accesses the appropriate tables, sorts the agents into agent types and lists them in the appropriate agent tab in the Promotion Target record.
4. To specify agent mapping, complete the steps below for each tab that contains agents:
    a. Click the tab title to display the target agents. The example below shows a list of Unix agents on a target server:



Linux/Unix Agent Mappings > Promotion Target = Test   1 to 5 of 5

Target Agent	Source Agent
server.opswisesoftware.com - 4	
server.opswisesoftware.com - 5	
server.opswisesoftware.com - 3	
server.opswisesoftware.com - 1	
server.opswisesoftware.com - 2	

    b. To specify mapping, click the name of a Target Agent. A record opens showing more detail, as shown in the example below.



Linux/Unix Agent Mapping

Promotion Target:	Test
Target Agent:	server.opswisesoftware.com - 4
Type:	Linux/Unix
Source Agent:	🔒

    c. In the Source Agent field, click the Lock icon 🔒. Click the magnifying glass to browse for and select a record from the list of agents defined on the source machine. Select the source agent that you want to map to the target agent.

d. Repeat the above procedure for each agent listed. For best results, you should make sure all the agents on your source system are mapped to an agent on the target system. You can map as many source agents to a single target agent as needed. Once you have specified the mapping for all your source agents, you can easily promote Bundles or individual records to this Promotion Target. When you promote records (via Bundle or individually) to the target machine, the target agent will replace the source agent.

Refresh Agent Error Messages

If your setup is incorrect, you may see the error message described below.

If you tried to refresh target agents using a non-existent user or invalid password on the Promotion Target:

Error Message	Location

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

`2012-03-29-16:27:17:134  ERROR [http-8080-6]`
`com.sun.jersey.api.client.UniformInterfaceException:`
`GET returned a response status of 401 Unauthorized`
	Opswise log on source machine

`2012-03-29-16:27:16:138  ERROR [http-8080-1] *** ERROR ***`
`Login using Basic Authentication failed for: [userID]`
	Opswise log on target machine

Promoting Bundles and Records to a Target

⚠ To use this feature, the user logged in to the source machine must have the ops_promotion_admin role. Also, the user ID and password specified for the Promotion Target must be a valid user on the target instance with the ops_promotion_admin role.

Introduction

Promoting a Bundle means copying all the records in the Bundle from a source to a target Opswise instance. You can also promote one or more individual records without first bundling them.

For every Bundle of records or individual record promoted, all of its associated records are also promoted. For example:

- If you promote a workflow, all of the tasks in the workflow are also promoted.
- If you promote a Windows task, all of its variables, Actions, Notifications, and so on are also promoted.
- If you promote an Application Resource, its associated start, stop, and query tasks are also promoted.
- If you promote a calendar, its associated days are also promoted.

To Promote a Bundle

1. Display the Bundle you want to promote.
2. Click **Promote Bundle**. The system displays a window, prompting you to select the Promotion Target from a drop-down list. Select the target.
3. The default login ID and password are provided from the Promotion Target record, if specified. If you want to override the default, click **Override User/Password** and type in the new information.
4. Click **Submit**. The system logs in to the target machine specified in the URI field of the Promotion Target Record and copies the bundled records to the target server. Based on the specified agent mapping, the target agent replaces the source agent where required.

This process creates audit records on the source and target machines. On the target machine, the system also creates a Promotion History record. For details see Promotion History and the Restore Option.

To Promote One or More Individual Records

Opswise also allows you to promote records to a target server without going through the process of creating a Bundle.

1. Select the record(s) you want to promote:
   - To promote a single record from a list of records:
     a. Right click on the record you want to promote.
     b. Select **Promote**.
   - To promote multiple records from a list of records:
     a. Select the records you want to promote by clicking the box to the left of the record name.
     b. From the **Actions on selected rows** menu, select **Promote**.
   - To promote the current record:
     a. Open the record you want to promote.
     b. Click the down-arrow next to the record name or right-click on the record name. A menu appears.
     c. Select **Promote**.
2. The system displays a window, prompting you to select the Promotion Target from a drop-down lists. Select the target.
3. The default login ID and password will be provided from the Promotion Target record. If you want to override the default, click **Override User/Password** and type in the new information.
4. Click **Submit**. The system logs in to the target machine specified in the URI field of the Promotion Target Record and copies the selected records to the target server.

This process creates audit records on the source and target machines. On the target machine, the system also creates a Promotion History record.

Promotion Error Messages

If your setup is incorrect, you may see the error messages described below.

If you tried to promote a bundle or record using a non-existent user or invalid password on the Promotion Target:

Error Message	Location

`Command Promote Bundle failed to execute:`
`POST`
`http://NN.NNN.NN.N:8080/opswise/resources/bundle/promote`
`returned a response status of 401 Unauthorized.`
	User interface and Opswise log on source machine

`2012-03-29-16:41:36:185  ERROR [http-8080-4] *** ERROR ***`
`Login using Basic Authentication failed for: [userID]`
	Opswise log on target machine

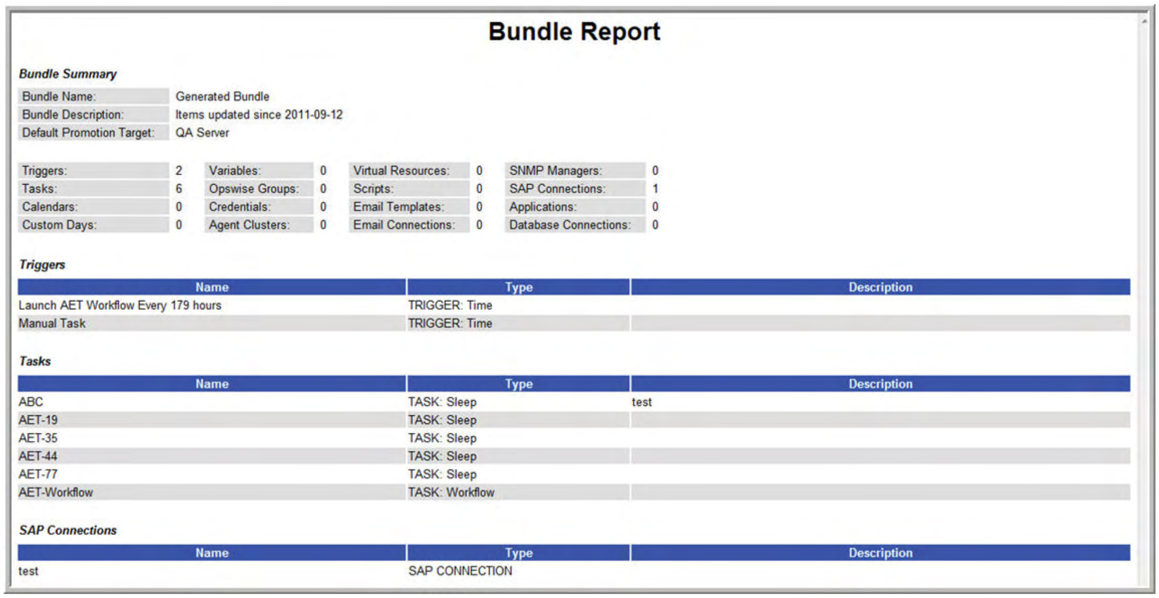
If you tried to promote a Bundle or record using a valid user/password on the Promotion Record that does not have the ops_promotion_admin role :

Error Message	Location

`Command Promote Bundle failed to execute:`
`[Command Accept Bundle prohibited due to security constraints.].`
	User interface on source machine

Generating a Bundle Report

The Bundle Report feature allows you to display on a single page all the records included in the current Bundle, with a summary of records at the top. To generate the report, display the Bundle and click the **Bundle Report** button. An example is shown below:



Bundle Report

Bundle Summary

Bundle Name:	Generated Bundle
Bundle Description:	Items updated since 2011-09-12
Default Promotion Target:	QA Server

Triggers:	2	Variables:	0	Virtual Resources:	0	SNMP Managers:	0
Tasks:	6	Opswise Groups:	0	Scripts:	0	SAP Connections:	1
Calendars:	0	Credentials:	0	Email Templates:	0	Applications:	0
Custom Days:	0	Agent Clusters:	0	Email Connections:	0	Database Connections:	0

Triggers

Name	Type	Description
Launch AET Workflow Every 179 hours	TRIGGER: Time	
Manual Task	TRIGGER: Time	

Tasks

Name	Type	Description
ABC	TASK: Sleep	test
AET-19	TASK: Sleep	
AET-35	TASK: Sleep	
AET-44	TASK: Sleep	
AET-77	TASK: Sleep	
AET-Workflow	TASK: Workflow	

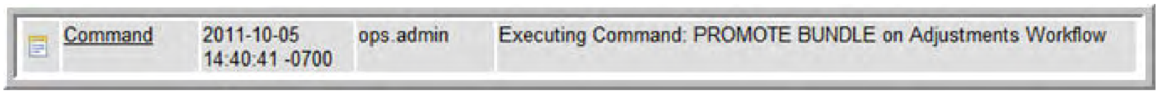
SAP Connections

Name	Type	Description
test	SAP CONNECTION	

Audit Records

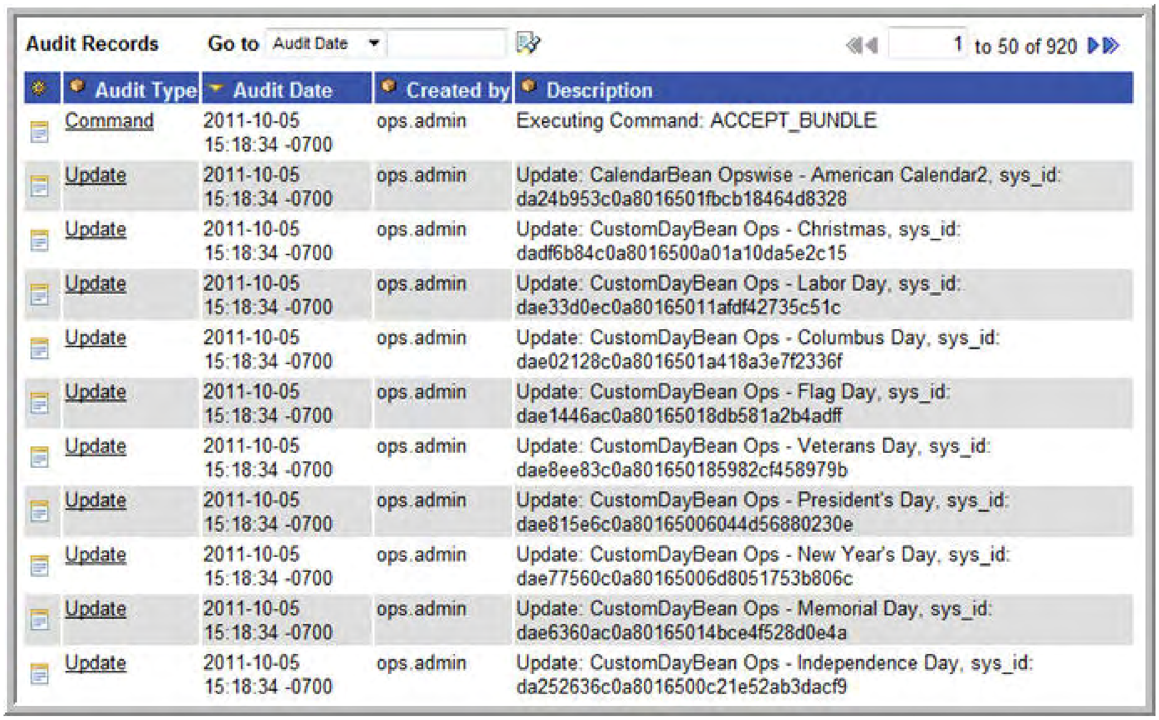
Whenever a Bundle or an individual record is promoted to a target server, the system creates audit records on both the source and target servers.

On the source system side, each time you promote a record or a Bundle, the system creates a single audit record for that event. If you promoted a Bundle, the audit message is PROMOTE_Bundle; if you promoted a single record or multiple records, the audit message is PROMOTE. An example audit record is shown below for a Bundle called Adjustments Workflow:



Command	2011-10-05 14:40:41 -0700	ops.admin	Executing Command: PROMOTE BUNDLE on Adjustments Workflow

On the target server side, the system creates an ACCEPT_BUNDLE audit record, along with "child" audit records associated with that promotion (either record[s] or a Bundle). These may include UPDATE commands for records that existed on the target already and CREATE commands for records that did not previously exist. For example, if you promote an updated Calendar record, the system creates an ACCEPT_BUNDLE audit for the promotion, and Update audit records for the calendar and each of the custom days used in the calendar, as shown in the following example:



Audit Type	Audit Date	Created by	Description
Command	2011-10-05 15:18:34 -0700	ops.admin	Executing Command: ACCEPT_BUNDLE
Update	2011-10-05 15:18:34 -0700	ops.admin	Update: CalendarBean Opswise - American Calendar2, sys_id: da24b953c0a8016501fbcb18464d8328
Update	2011-10-05 15:18:34 -0700	ops.admin	Update: CustomDayBean Ops - Christmas, sys_id: dadf6b84c0a8016500a01a10da5e2c15
Update	2011-10-05 15:18:34 -0700	ops.admin	Update: CustomDayBean Ops - Labor Day, sys_id: dae33d0ec0a80165011afdf42735c51c
Update	2011-10-05 15:18:34 -0700	ops.admin	Update: CustomDayBean Ops - Columbus Day, sys_id: dae02128c0a8016501a418a3e7f2336f
Update	2011-10-05 15:18:34 -0700	ops.admin	Update: CustomDayBean Ops - Flag Day, sys_id: dae1446ac0a80165018db581a2b4adff
Update	2011-10-05 15:18:34 -0700	ops.admin	Update: CustomDayBean Ops - Veterans Day, sys_id: dae8ee83c0a801650185982cf458979b
Update	2011-10-05 15:18:34 -0700	ops.admin	Update: CustomDayBean Ops - President's Day, sys_id: dae815e6c0a80165006044d56880230e
Update	2011-10-05 15:18:34 -0700	ops.admin	Update: CustomDayBean Ops - New Year's Day, sys_id: dae77560c0a80165006d8051753b806c
Update	2011-10-05 15:18:34 -0700	ops.admin	Update: CustomDayBean Ops - Memorial Day, sys_id: dae6360ac0a80165014bce4f528d0e4a
Update	2011-10-05 15:18:34 -0700	ops.admin	Update: CustomDayBean Ops - Independence Day, sys_id: da252636c0a8016500c21e52ab3dacf9

Promotion History and the Restore Option

The target system creates a Promotion History record each time a record or a Bundle is promoted into the system. You can access the Promotion History records by selecting **Automation Center Bundles & Promotion > Promotion History** from the navigation pane on the target server.

Each Promotion History record provides a complete list of all records promoted during this promotion event, along with their version numbers. This screen also provides the option of restoring records to the state they were in before the promotion. This applies only to records being updated by the promotion, not those being created by a promotion. See the field descriptions below for details.

The fields on the Promotion History screen are system-supplied and display-only.

Field Name	Description
Bundle Name	Name of this record.
Source Node	Machine name or URI of the machine where the source Opswise system is running.
Promotion User	UserID of the user who promoted the bundle or record(s).
Updated	Date and time this record was last updated.
Promotion History Items tab	This tab lists all the record (items) promoted as part of this bundle or set of records. Each item in the list provides the name and type of the record, the latest version number on the target, the previous version number on the target, and the source version number.
Run Report button	Generates a one page summary of the contents of the bundle. See Generating a Bundle Report.
Restore Unchanged button	For records that already existed on the target server, you can restore them to their state prior to the promotion. The **Restore Unchanged** button restores only those records that have not been changed since the promotion updated the record. For records that were created on the target machine by the promotion, no changes will occur since no previous version exists.
Restore All button	For records that already existed on the target server, you can restore them to their state prior to the promotion. The **Restore All** button restores all records that were updated by the promotion, including records that were modified since the promotion. For records that were created on the target machine by the promotion, no changes will occur since no previous version exists.
Delete button	Deletes the current record.

Backing Up and Purging Data

Overview

Opswise maintains a record of all system activity, including:

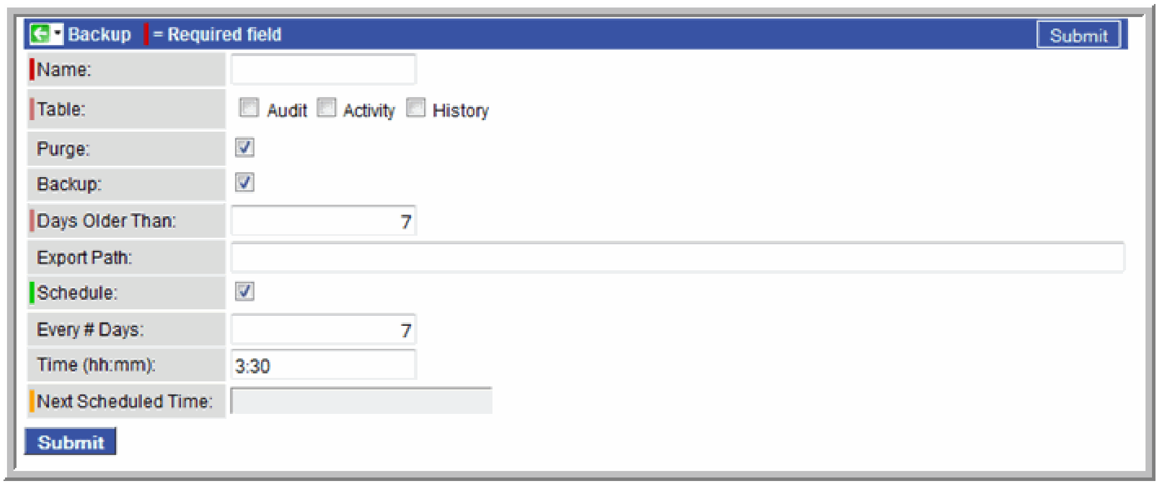
- Audit records
- Activity
- History

The Backup screen allows you to configure automatic backups and/or purges of some or all of the system activity data. Depending on your organization's needs, you should schedule regular data backups. Depending on the volume of your installation, the amount of data in your Opswise database could become unwieldy if you do not schedule regular purges of old data.

The data is written to XML files in the directory you specify.

Creating a New Backup / Purge Instruction

1. From the navigation pane, select **Automation Center Administration > Configuration > Data Backup / Purge**. The Backups List screen displays.
2. Click **New**. The Backup Definition screen displays.



Name:	
Table:	□ Audit □ Activity □ History
Purge:	☑
Backup:	☑
Days Older Than:	7
Export Path:	
Schedule:	☑
Every # Days:	7
Time (hh:mm):	3:30
Next Scheduled Time:	

3. Using the field descriptions provided below as a guide, complete the fields as needed. If you want the backup/purge to run automatically, enable the Schedule field and specify how often and what time it should run. Otherwise, you can run it manually.
4. Click the **Submit** button to save the record and return to the menu, or right-click the title bar and select **Save** to save the record and remain on the current display.
5. To enable the instructions, right-click on the record name to display the menu and select **Enable Backup**. The list of Backup records identifies which records are enabled and disabled.
6. If appropriate, repeat these steps for any additional Backup / Purge records you want to add.

Manually Running a Backup/Purge

If you want to manually run a backup or purge, right-click on the record you want to run and select **Run** or open the record and click the **Run** button.

Importing Backed/Purged Data into Opswise

If you need to import any of the XML files created by using the backup/purge function, you can copy the XML file(s) into the bulk export output path and run bulk import. See Running Opswise Import .

Backup Field Descriptions

Field Name	Description
Name	Name of this backup specification.
Table	Specifies which records you want to back up and/or purge: • Audit • Activity • History
Purge	If enabled, the process will purge the selected data from your Opswise database.
Backup	If enabled, the process will write all the selected data to XML files.
Days Older Than	Allows you to specify the minimum number of days you wish to retain data. The process will run according to the schedule you specify, only processing data that is older than the number of days you specify in this field.

Export Path	Specifies the path to the directory to which you want the backed up data written. The data must be backed up to a location on the server's file system. It is written to a separate XML file for each record type, as shown in the following examples:
	`Audit:`
	`ops_audit_Sat_Apr_30_08_30_00_PDT_2011.xml`
	`Activity:`
	`ops_exec_sleep_Sat_Apr_30_08_30_00_PDT_2011.xml`
	`ops_exec_unix_Sat_Apr_30_08_30_00_PDT_2011.xml`
	`ops_exec_workflow_Sat_Apr_30_08_30_00_PDT_2011.xml`
	`History:`
	`ops_history_Sat_Apr_30_08_30_00_PDT_2011.xml`

Schedule	If enabled, displays additional fields that allow you to specify an automated backup and/or purge schedule. If you do not select schedule, you must manually run the backup / purge process.
Every # Days	Specifies, in number of days, the frequency of the backup / purge process.
Time	Specifies the time of the backup / purge. Use 24:00 hour time.
Next Scheduled Time	Displays the next scheduled time the backup / purge process will run, based on the specifications in your schedule.
Submit button	Submits the new record to the database.

Update button	Saves updates to the record.
Enable Backup button	Enables these Backup / Purge instructions so that they will be processed by the Core.
Disable Backup button	Disables these backup / purge instructions so they will not be processed by the Core.
Delete button	Deletes the current record.
Run button	Manually runs the backup / purge instructions.

Record Versioning

Overview

Opswise maintains historical copies of most user-created records in the database. These include tasks and their associated records (virtual resources, variables, actions, notes), calendars and their custom day associations, custom days, variables, credentials, virtual resources, scripts (and associated notes), email templates and connections, database connections, SNMP managers, SAP connections, agent clusters, applications, Opswise groups, and triggers (and associated variables).

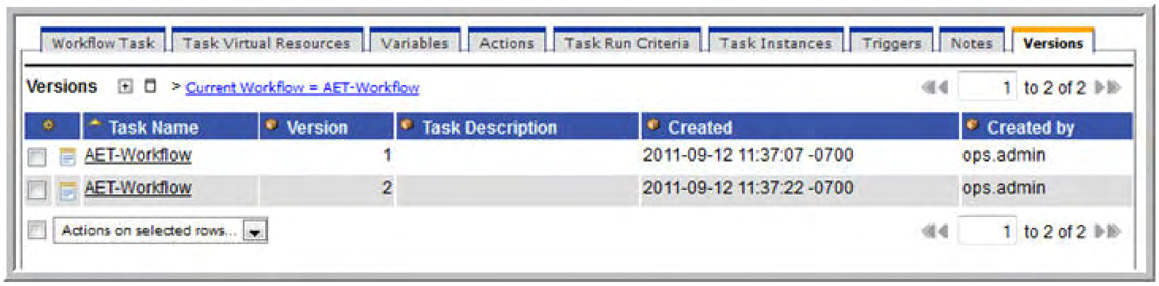
When the user updates one of the above-listed records, the system creates an image of the old version and stores it in the record's **Versions** tab. It also updates the Version field in the current version of the record. For example, if you have updated TaskABC three times, you will have three versions of the task stored in the Versions tab, along with the current version, which displays as Version #4.

Click hereto watch a video about Record Versioning.

Viewing Old Versions of Records

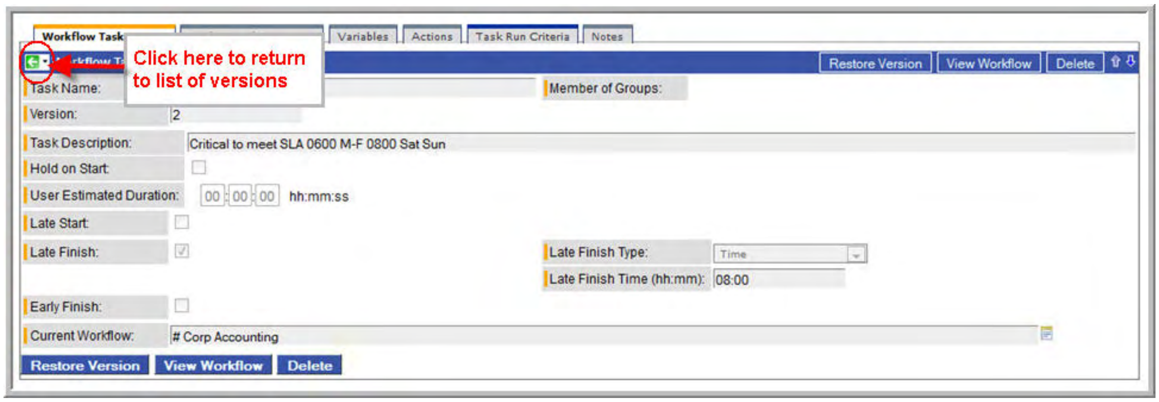
Previous record versions are read-only. To view previous versions:

1. Open the record you want to view.
2. Click the **Versions** tab. The system displays a list of versions that exist for this record. You can sort this list in ascending or descending order by clicking the **Version** column label.



Task Name	Version	Task Description	Created	Created by
AET-Workflow	1		2011-09-12 11:37:07 -0700	ops.admin
AET-Workflow	2		2011-09-12 11:37:22 -0700	ops.admin

3. Click the record you want to view. All of the associated records (tabs) contain the data as it existed before the user updated the record. At the bottom of the record, the Current (record type) field displays the current name of the record. That is, if a user had changed the name of the record, the current name displays in the Current (record type) field.
4. To return to the list of Record Versions, click the green arrow.



Task Name:		Member of Groups:	
Version:	2		
Task Description:	Critical to meet SLA 0600 M-F 0800 Sat Sun		
Hold on Start:	□		
User Estimated Duration:	00 : 00 : 00 hh:mm:ss		
Late Start:	□		
Late Finish:	☑	Late Finish Type:	Time
		Late Finish Time (hh:mm):	08:00
Early Finish:	□		
Current Workflow:	# Corp Accounting		

Restoring Old Versions of Records

You can restore old versions to the current version. When you restore an older version, the current version will become a new historical version.

1. Display the version of the record you want to restore.
2. Click the **Restore Record** button.

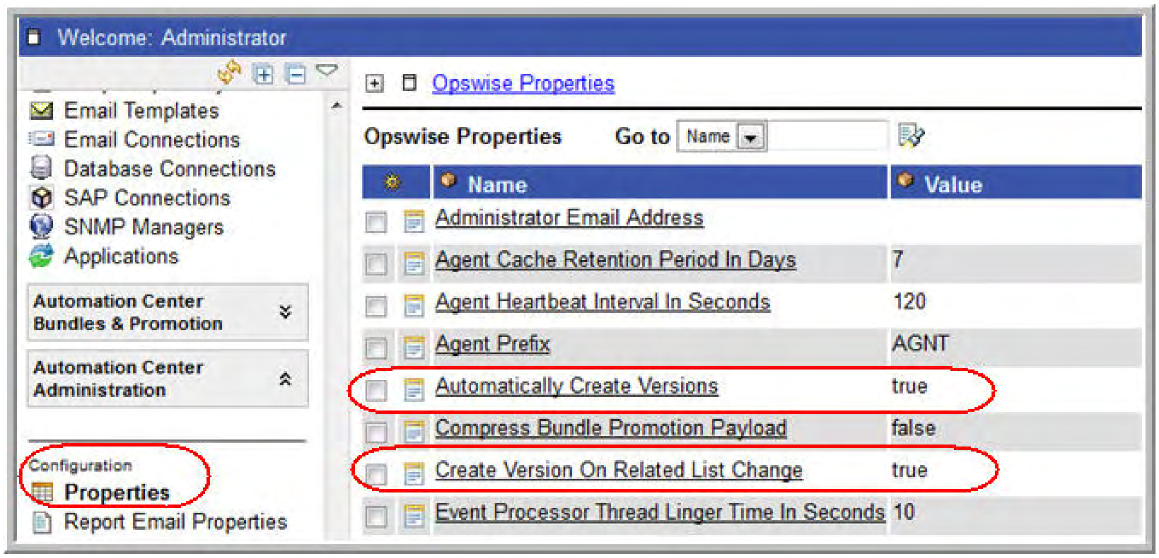
Enabling/Disabling Versioning

Two properties are available that allow you to control if and when Opswise automatically creates a new version of a record (and all its associated records):

- The **Automatically Create Versions** property (true or false) determines whether modifications to the record itself will cause the system to create a new version of the record. The default value is true. If this property is set to false, the system does not create versions.
- The **Create Version On Related List Change** property (true or false) determines whether changes, deletions, or additions to a related list will cause the system to create a new version of the record. For example, if this property is enabled, the system will create a new version of the task and all its associated records when the user adds a variable to the task, deletes a Note, or changes an Email Notification. The default value is true. If this property is set to false, and the **Automatically Create Versions** property is set to true, the system creates a new version only if the base record is updated.

To change one or both of the above-described properties:

1. From the navigation pane, select **Configuration > Properties.**
2. Locate the property you want to change, as shown in the following illustration.



Name	Value
Administrator Email Address	
Agent Cache Retention Period In Days	7
Agent Heartbeat Interval In Seconds	120
Agent Prefix	AGNT
Automatically Create Versions	true
Compress Bundle Promotion Payload	false
Create Version On Related List Change	true
Event Processor Thread Linger Time In Seconds	10

3. To change the value, click the name to open the record and type "true" or "false" in the Value field.
4. Click the **Update** button.

Exporting and Importing Records

Opswise provides several utilities that allow you to export and import records. Normally you will use these to migrate data during a system upgrade.

See Run an Opswise Export and Run Opswise Import in the migration documentation.

Troubleshooting

Frequently Asked Questions (FAQs)

Tomcat

- Can Tomcat run from the root directory?
- Can Tomcat run as user root?

Database

- MySQL - Can't Access root@localhost
- How to speed up MySQL performance (Windows)

Installation of Core Processor

- Error when starting core or msghub

Installation of Outboard

- Processes won't start automatically (Debian Linux)
- Windows install fails with 'Service marked for deletion'
- Error when starting core or msghub

Operations

- Cannot launch a task
- VBScript stuck in "Running" state
- My Connections have disappeared from the Connections list

Error Messages

- The conversion of a varchar data type to a datetime data type resulted in an out-of-range value. (SQLServer)
- ops_suexec___Not enough privileges. Check SUID bit and binary owner
- Packet for query is too large (1084852 > 1048576)
- SQLSTATE: HY000, SQLERR. 1040, ERRMSG. unixODBC MySQL ODBC 5.1 Driver Too many connections
- This Service is marked for deletion
- You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version
- Error 401 - Wrong Credentials

Can Tomcat run from the root directory?

Yes.

To run Tomcat from the root directory, add the following line into the <Host> section of your conf/server.xml configuration file in the Tomcat installation directory:

```
<Context path="" debug="0" docBase="opswise" />
```

The entire section would look similar to the following:

```
     <Host name="localhost"  appBase="webapps"
           unpackWARs="true" autoDeploy="true"
           xmlValidation="false" xmlNamespaceAware="false">
               <Context path="" debug="0" docBase="opswise" />
     </Host>
```

Can Tomcat run as user root?

Yes.

However, you need to tell Opswise where to write logs and export files. Normally these are written to $TOMCAT_HOME/opswise_logs and $TOMCAT_HOME/opswise_export.

To resolve this, you must allocate these directories and set ownership to Tomcat. Do these tasks before you start Opswise. (The user:group names may be different for your installation.)

```
cd $TOMCAT_DIR
mkdir opswise_logs
chown tomcat:tomcat opswise_logs
mkdir opswise_export
chown tomcat:tomcat opswise_export
```

MySQL - Can't Access rootatlocalhost

After you install a new MySQL server, if you are unable to access root@localhost, set the root password:

```
mysqladmin -u root password NEWPASSWORD
```

How to speed up MySQL performance

To speed up MySQL performance for Windows installations, you can add the following parameter to the appropriate MySQL.ini file:

```
innodb_flush_log_at_trx_commit=0
```

For more information about this parameter, see the MySQL documentation:

http://dev.mysql.com/doc/refman/5.1/en/innodb-parameters.html#sysvar_innodb_flush_log_at_trx_commit

Error when starting Core or Msghub

Upon starting the Core or the Msghub server, the opswise.log shows ERR:

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

You need to set additional connections in your database server:

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

Processes won't start automatically (Debian Linux)

For Debian Linux environment, if the outboard processes won't start automatically at boot time:

- Debian Linux does not provide the chkconfig command and therefore cannot work with the runlevels specified in the opsagent, opstransport, opsmsghub scripts provided in /etc/init.d. This is a known problem; we are working on a solution.

Windows install fails with 'Service marked for deletion'

In a Windows environment, for upgrade installs of the outboard components, if the install fails with the message:

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

Before running the install or upgrade process, make sure you first exit the Windows Services Console. If you don't, you will likely get this error. If this occurs, exit the Windows Services application and re-run the install.

Cannot launch a task

If you cannot launch a task, the problem may be with your credentials. Check the stderr for the following message:

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

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

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

VBScript stuck in "Running" state

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

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

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

Out of range value during z-boot

When using SQL Server during z-boot of the system, if you see the message:

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

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

Packet for query is too large

During operations, you may see this message in the Opswise or Msghub logs:

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

You can change this value on the server by setting the max_allowed_packet' variable.

You have an error in your SQL syntax

You may see this error when you execute an SQL task that includes multiple SQL commands; for example:

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

By default, multiple queries are disabled for MySQL. To enable multiple queries, append the following string to the Connection URL field in the Database Connection resource definition:

```
?allowMultiQueries=true
```

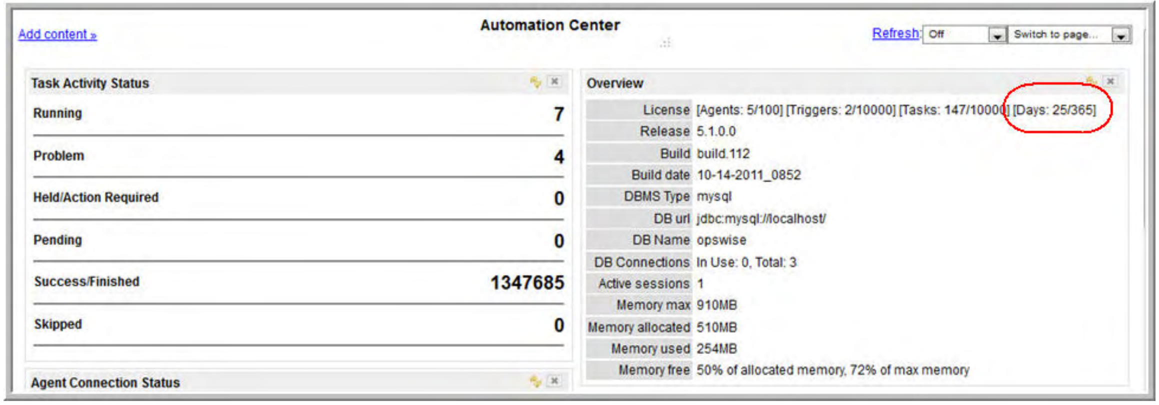
Here is an example URL connection string for a MySQL Database Connection resource definition:

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

My Connections have disappeared from the Connections list

This may occur if your license has expired; check your license as follows:

1. Click the Home 🏠 button.
2. In the Overview gauge, check the number of days left on your license, as illustrated below:



Task Activity Status

Running	7
Problem	4
Held/Action Required	0
Pending	0
Success/Finished	1347685
Skipped	0

Overview

License	[Agents: 5/100] [Triggers: 2/10000] [Tasks: 147/10000] [Days: 25/365]
Release	5.1.0.0
Build	build.112
Build date	10-14-2011_0852
DBMS Type	mysql
DB url	jdbc:mysql://localhost/
DB Name	opswise
DB Connections	In Use: 0, Total: 3
Active sessions	1
Memory max	910MB
Memory allocated	510MB
Memory used	254MB
Memory free	50% of allocated memory, 72% of max memory

Error 401 - Wrong Credentials

You will see this error when you click **Refresh Target Agents** on a Promotion Target record and you are using invalid login credentials for the target Opswise instance.

The Opswise log will show the following error.

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

The target machine will return the following error:

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

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