



**stonebranch**

Universal Controller 7.2.x

Instances

© 2022 by Stonebranch, Inc. All Rights Reserved.

- 1. Instances ..... 3
  - 1.1 Manually Running and Controlling Tasks ..... 4
  - 1.2 Monitoring Activity from the Task Instances List ..... 38
  - 1.3 Monitoring Activity History ..... 43
  - 1.4 Viewing Task Instances for a Specific Task ..... 49
  - 1.5 Displaying Task Instance Status ..... 58
  - 1.6 Retrieving Output ..... 61

# Instances



## Task Instances



## History

[Monitoring Activity from the Task Instances List](#)

[Viewing Task Instances for a Specific Task](#)

[Displaying Task Instance Status](#)

[Retrieving Output](#)

[Monitoring Activity History](#)



The information on these pages also is located in the [Universal Controller 7.2.x Tasks.pdf](#) and the [Universal Controller 7.2.x Workflows.pdf](#).

# Manually Running and Controlling Tasks

- Overview
- Issuing Commands Against Task Instances
- Issuing Commands from the Activity Monitor
  - Issue a Command Against a Single Task Instance
  - Issue a Command Against Multiple Task Instances
- Issuing Commands from the Task Instances List
  - Issue a Command Against a Single Task Instance
  - Issue a Command Against Multiple Task Instances
- Issuing Commands from the Workflow Monitor
  - Issue a Command Against a Task Instance within the Workflow
  - Issue a Command Against the Workflow Task Instance
- Commands Supported for Task Instance Statuses
  - Agent-Based Task Types
- Manually Launching a Task
  - Launch One or More Tasks from a Tasks List
  - Launch a Task from Task Details
  - Launch a Task Manually with Temporary Variable Values
  - Launch a Task Using Unsaved Modifications
- Changing the Priority of a Task Instance
  - Set Priority on a Task Instance from the Activity Monitor or Task Instances List
  - Set Priority on a Task Instance from the Workflow Monitor
- Re-running a Task Instance
  - Re-run a Task Instance from the Activity Monitor or Task Instances List
  - Re-run a Task Instance from the Workflow Monitor
  - Re-run a z/OS Task Instance in the In Doubt Status
- Cancelling a Task Instance
  - Cancel a Task Instance from the Activity Monitor or Task Instances List
  - Cancel a Task Instance from the Workflow Monitor
- Force Finishing a Task Instance
  - Force Finish a Task Instance from the Activity Monitor or Task Instances List
  - Force Finish a Task Instance from the Workflow Monitor
- Force Finishing (Halt) a Task Instance
  - Force Finish (Halt) a Task Instance from the Activity Monitor or Task Instances List
  - Force Finish (Halt) a Task Instance from the Workflow Monitor
- Force Finish/Cancelling a Task Instance
  - Force Finish/Cancel a Task Instance from the Activity Monitor
  - Force Finish/Cancel a Task Instance from the Workflow Monitor
- Force Finish/Cancelling (Halt) a Task Instance
  - Force Finish/Cancel (Halt) a Task Instance from the Activity Monitor
  - Force Finish/Cancel (Halt) a Task Instance from the Workflow Monitor
- Putting a Task Instance on Hold
  - Hold a Task Instance from the Activity Monitor or Task Instances List
  - Hold a Task Instance from the Workflow Monitor
- Releasing a Task Instance from Hold
  - Release a Held Task Instance from the Activity Monitor or Task Instances List
  - Release a Held Task Instance from the Workflow Monitor
- Skipping a Task Instance
  - Skip a Task Instance from the Activity Monitor or Task Instances List
  - Skip a Task Instance from the Workflow Monitor
- Showing or Hiding Skipped Task Instances
  - Setting Show / Hide Skipped Tasks from the Workflow Task Details
  - Setting Show / Hide Skipped Tasks from the Workflow Task Instance Details
  - Setting Show / Hide Skipped Tasks from the Workflow Monitor

- [Unskipping a Task Instance](#)
  - [Unskip a Task Instance from the Activity Monitor or Task Instances List](#)
  - [Unskip a Task Instance from the Workflow Monitor](#)
- [Marking a Dependency as Satisfied](#)
  - [Marking a Dependency as Satisfied from the Workflow Monitor](#)
- [Clearing Predecessor Dependencies of a Task Instance](#)
  - [Clearing Predecessor Dependencies of a Task Instance from the Workflow Monitor](#)
- [Clearing Resource Dependencies of a Task Instance](#)
  - [Clear Resource Dependencies of a Task Instance from the Activity Monitor or Task Instances List](#)
  - [Clear Resource Dependencies of a Task Instance from the Workflow Monitor](#)
- [Clearing Mutually Exclusive Dependencies of a Task Instance](#)
  - [Clear Mutually Exclusive Dependencies of a Task Instance from the Activity Monitor or Task Instances List](#)
  - [Clear Mutually Exclusive Dependencies of a Task Instance from the Workflow Monitor](#)
- [Clearing All Dependencies of a Task Instance](#)
  - [Clear All Dependencies of a Task Instance from the Activity Monitor or Task Instances List](#)
  - [Clear All Dependencies of a Task Instance within a Workflow from the Workflow Monitor](#)
  - [Clear All Dependencies of a Workflow Task Instance from the Workflow Monitor](#)
- [Clearing Time Wait/Delay Specifications of a Task Instance](#)
  - [Clear All Time Wait/Delay Specifications of a Task Instance from the Activity Monitor or Task Instances List](#)
  - [Clear All Time Wait/Delay Specifications of a Task Instance from the Workflow Monitor](#)
  - [Clear All Time Wait/Delay Specifications of a Task Instance from the Task Instance Details](#)

## Overview

A number of commands are available on the [Activity Monitor](#) and the [Task Instances list](#) 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](#).

## Issuing Commands Against Task Instances

You can issue commands against task instances from:

- [Activity Monitor](#)
- [Task Instances list](#) (and the [Task Instances list for a specific task](#))
- [Workflow Monitor](#).

See [Commands Supported for Task Instance Statuses](#) for a list of task instances (and their statuses) for which these commands can be issued.

See [Task Instance Status Types](#) for a description of each type of task instance status.

Command	Description
Cancel	Cancels a running task instance (see <a href="#">Cancelling a Task Instance</a> ), including a task instance in a completed workflow (status = Success, Finished, Skipped).
Clear All Dependencies	Workflow tasks only: Clears all dependencies (predecessors, resources, and exclusive) of a task instance (see <a href="#">Clearing All Dependencies of a Task Instance</a> ).
Clear Exclusive	Clears mutually exclusive dependencies of a task instance (see <a href="#">Clearing Mutually Exclusive Dependencies of a Task Instance</a> ).
Clear Predecessors	Workflow tasks only: Clears predecessor dependencies of a task instance (see <a href="#">Clearing Predecessor Dependencies of a Task Instance</a> ).
Clear Resources	Clears resource dependencies of a task instance (see <a href="#">Clearing Resource Dependencies of a Task Instance</a> ).

Clear Time/Wait Delay	Clears all Wait To Start and Delay On Start specifications for this task instance (see <a href="#">Clearing Time Wait/Delay Specifications of a Task Instance</a> ).
Force Finish	Places a task instance into the Finished status (see <a href="#">Force Finishing a Task Instance</a> ), including a task instance in a completed workflow (status = Success, Finished, Skipped).
Force Finish (Halt)	Places a task instance into the Finished status (see <a href="#">Force Finishing (Halt) a Task Instance</a> ), including a task instance in a completed workflow (status = Success, Finished, Skipped).
Force Finish/Cancel	Cancels a task and places it into the Finished status (see <a href="#">Force Finish/Cancelling a Task Instance</a> ); including a task instance in a completed workflow (status = Success, Finished, Skipped).
Force Finish/Cancel (Halt)	Cancels a task and places it into the Finished status (see <a href="#">Force Finish/Cancelling (Halt) a Task Instance</a> ).
Hold	Temporarily prevents a task instance from running (see <a href="#">Putting a Task Instance on Hold</a> ).
Release	Removes a task instance from being on Hold (see <a href="#">Releasing a Task Instance from Hold</a> ).
Release Recursive	Workflow tasks only: Removes a workflow and its task instances from being on Hold (see <a href="#">Releasing a Task Instance from Hold</a> ).
Re-run	Not applicable for Workflow tasks: Re-runs a task instance (see <a href="#">Re-running a Task Instance</a> ), including a task instance in a completed workflow (status = Success, Finished, Skipped).
Re-run (Suppress Intermediate Failures)	Not applicable for Workflow tasks: Re-runs a task instance (see <a href="#">Re-running a Task Instance</a> ) specifying that intermediate failures be suppressed, including a task instance in a completed workflow (status = Success, Finished, Skipped).
Retrieve Output	<p>Retrieves output (Standard Output and/or Standard Error, or in the case of z/OS, the Job Log) for any running or completed task instance, limited to the following supported task types:</p> <ul style="list-style-type: none"> <li>• Application Control</li> <li>• Remote File Monitor</li> <li>• Linux/Unix</li> <li>• PeopleSoft</li> <li>• SAP</li> <li>• Universal</li> <li>• Universal Command</li> <li>• Windows</li> <li>• z/OS</li> </ul>
Set Completed	Sets a <a href="#">Manual Task</a> instance to the Success status.
Set Priority	Sets the priority of a task instance in Started, Running, or Queued status to High, Medium, or Low (see <a href="#">Changing the Priority of a Task Instance</a> ).
Set Started	Resets the Started Time of a <a href="#">Manual Task</a> instance.
Skip	Disregards a task instance (see <a href="#">Skipping a Task Instance</a> ).
Skip Path	Disregards a task instance and all of its dependent task instances (see <a href="#">Skipping a Task Instance</a> ).
Unskip	Removes the Skip status of a task instance (see <a href="#">Unskipping a Task Instance</a> ).

## Issuing Commands from the Activity Monitor

Instance Name	Type	Status	Invoked By	Start Time	End Time	Updated
lecu-wkfl-sleep	Workflow	Success	Manually Launched	2014-09-02 12:57:55 -0400	2014-09-02 12:58:36 -0400	2014-09-02 12:58:36 -0400
sleep 10	Timer	Success	Workflow: lecu-wkfl-sleep	2014-09-02 12:58:26 -0400	2014-09-02 12:58:36 -0400	2014-09-02 12:58:36 -0400
Sleep 30	Timer	Success	Workflow: lecu-wkfl-sleep	2014-09-02 12:57:56 -0400	2014-09-02 12:58:26 -0400	2014-09-02 12:58:26 -0400
Sleep 0	Timer	Success	Workflow: lecu-wkfl-sleep	2014-09-02 12:57:56 -0400	2014-09-02 12:57:56 -0400	2014-09-02 12:57:56 -0400
Sleep 60	Timer	Skipped	Workflow: lecu-wkfl-sleep		2014-09-02 12:57:55 -0400	2014-09-02 12:57:55 -0400
Sleep 60	Timer	Skipped	Workflow: lecu-wkfl-sleep		2014-09-02 12:57:55 -0400	2014-09-02 12:57:55 -0400
Sleep 60	Timer	Skipped	Workflow: lecu-wkfl-sleep		2014-09-02 12:57:55 -0400	2014-09-02 12:57:55 -0400
Sleep 30	Timer	Skipped	Workflow: lecu-wkfl-sleep		2014-09-02 12:57:55 -0400	2014-09-02 12:57:55 -0400
Sleep 30	Timer	Skipped	Workflow: lecu-wkfl-sleep		2014-09-02 12:57:55 -0400	2014-09-02 12:57:55 -0400
win-exit-code	Windows	Failed	Manually Launched	2014-09-02 12:44:03 -0400	2014-09-02 12:44:03 -0400	2014-09-02 12:44:03 -0400
zos-workflow-regression-test	Workflow	Success	Manually Launched	2014-09-02 11:52:47 -0400	2014-09-02 12:10:30 -0400	2014-09-02 12:10:30 -0400
zos-workflow-simple-load-test-01	Workflow	Success	Workflow: zos-workflow-regression-test	2014-09-02 11:56:39 -0400	2014-09-02 12:10:30 -0400	2014-09-02 12:10:30 -0400
zos-task-load-simple-01	z/OS	Success	Workflow: zos-workflow-simple-load-test-01	2014-09-02 12:10:29 -0400	2014-09-02 12:10:29 -0400	2014-09-02 12:10:30 -0400
zos-task-load-simple-01	z/OS	Success	Workflow: zos-workflow-simple-load-test-01	2014-09-02 12:10:25 -0400	2014-09-02 12:10:26 -0400	2014-09-02 12:10:27 -0400
zos-task-load-simple-01	z/OS	Success	Workflow: zos-workflow-simple-load-test-01	2014-09-02 12:10:23 -0400	2014-09-02 12:10:23 -0400	2014-09-02 12:10:24 -0400
zos-task-load-simple-01	z/OS	Success	Workflow: zos-workflow-simple-load-test-01	2014-09-02 12:10:20 -0400	2014-09-02 12:10:20 -0400	2014-09-02 12:10:21 -0400
zos-workflow-simple-load-test-02	Workflow	Success	Workflow: zos-workflow-regression-test	2014-09-02 11:56:39 -0400	2014-09-02 12:10:19 -0400	2014-09-02 12:10:19 -0400
zos-task-load-simple-02	z/OS	Success	Workflow: zos-workflow-simple-load-test-02	2014-09-02 12:10:17 -0400	2014-09-02 12:10:17 -0400	2014-09-02 12:10:19 -0400
zos-task-load-simple-01	z/OS	Success	Workflow: zos-workflow-simple-load-test-01	2014-09-02 12:10:15 -0400	2014-09-02 12:10:15 -0400	2014-09-02 12:10:19 -0400
zos-task-load-simple-02	z/OS	Success	Workflow: zos-workflow-simple-load-test-02	2014-09-02 12:10:12 -0400	2014-09-02 12:10:12 -0400	2014-09-02 12:10:15 -0400
zos-task-load-simple-01	z/OS	Success	Workflow: zos-workflow-simple-load-test-01	2014-09-02 12:10:11 -0400	2014-09-02 12:10:11 -0400	2014-09-02 12:10:14 -0400
zos-workflow-simple-load-test-03	Workflow	Success	Workflow: zos-workflow-regression-test	2014-09-02 11:56:39 -0400	2014-09-02 12:10:13 -0400	2014-09-02 12:10:13 -0400
zos-task-load-simple-03	z/OS	Success	Workflow: zos-workflow-simple-load-test-03	2014-09-02 12:10:09 -0400	2014-09-02 12:10:09 -0400	2014-09-02 12:10:13 -0400
zos-task-load-simple-02	z/OS	Success	Workflow: zos-workflow-simple-load-test-02	2014-09-02 12:10:07 -0400	2014-09-02 12:10:08 -0400	2014-09-02 12:10:11 -0400
zos-task-load-simple-01	z/OS	Success	Workflow: zos-workflow-simple-load-test-01	2014-09-02 12:10:06 -0400	2014-09-02 12:10:06 -0400	2014-09-02 12:10:10 -0400

## Issue a Command Against a Single Task Instance

Either:

- Right-click a task instance on the list to display an [Action menu](#) of available commands for that task instance.
- Click the Details icon of a task instance to display the Task Instance Details, and then right-click in the Details to display an [Action menu](#) of available commands for that task instance.

## Issue a Command Against Multiple Task Instances

Press **Ctrl** and right-click each task instance that you want to issue a command against to display an [Action menu](#) of available commands that can be issued against all of the selected task instances.

(You also can press **Ctrl** and right-click a single task instance and then press **Shift** and right-click another task instance to select the group of task instances between the first and second task instance, inclusive.)

## Issuing Commands from the Task Instances List

Instance Name	Type	Status	Invoked By	Start Time	End Time	Updated
ecu-uag-d03237rc	z/OS	Success	Manually Launched	2014-08-21 21:28:09 -0400	2014-08-21 21:28:10 -0400	2014-08-21 21:28:11 -0400
zos-task-run-simple	z/OS	Success	Workflow: zos-workflow-regression-test	2014-08-21 21:29:39 -0400	2014-08-21 21:29:39 -0400	2014-08-21 21:29:41 -0400
zos-task-security-auth-01	z/OS	Success	Workflow: zos-workflow-user-authentication	2014-08-21 21:29:42 -0400	2014-08-21 21:29:42 -0400	2014-08-21 21:29:44 -0400
zos-task-security-auth-02	z/OS	Success	Workflow: zos-workflow-user-authentication	2014-08-21 21:29:42 -0400	2014-08-21 21:29:43 -0400	2014-08-21 21:29:45 -0400
zos-task-security-auth-03	z/OS	Success	Workflow: zos-workflow-user-authentication	2014-08-21 21:29:43 -0400	2014-08-21 21:29:44 -0400	2014-08-21 21:29:47 -0400
Sleep 0	Timer	Success	Workflow: zos-workflow-user-authentication	2014-08-21 21:29:47 -0400	2014-08-21 21:29:47 -0400	2014-08-21 21:29:47 -0400
zos-task-security-auth-13	z/OS	Finished	Workflow: zos-workflow-user-authentication	2014-08-21 21:29:48 -0400	2014-08-21 21:29:48 -0400	2014-08-21 21:29:48 -0400
zos-task-security-auth-10	z/OS	Finished	Workflow: zos-workflow-user-authentication	2014-08-21 21:29:48 -0400	2014-08-21 21:29:48 -0400	2014-08-21 21:29:48 -0400
zos-task-security-auth-12	z/OS	Finished	Workflow: zos-workflow-user-authentication	2014-08-21 21:29:48 -0400	2014-08-21 21:29:48 -0400	2014-08-21 21:29:48 -0400
zos-task-security-auth-11	z/OS	Finished	Workflow: zos-workflow-user-authentication	2014-08-21 21:29:48 -0400	2014-08-21 21:29:48 -0400	2014-08-21 21:29:49 -0400
zos-task-failure	z/OS	Skipped	Workflow: zos-workflow-user-authentication	2014-08-21 21:29:48 -0400	2014-08-21 21:29:48 -0400	2014-08-21 21:29:49 -0400
Sleep 0	Timer	Success	Workflow: zos-workflow-user-authentication	2014-08-21 21:29:49 -0400	2014-08-21 21:29:49 -0400	2014-08-21 21:29:49 -0400
zos-workflow-user-authenti...	Work...	Success	Workflow: zos-workflow-regression-test	2014-08-21 21:29:41 -0400	2014-08-21 21:29:49 -0400	2014-08-21 21:29:49 -0400
zos-step-action-setup	Timer	Success	Workflow: zos-workflow-step-actions	2014-08-21 21:29:49 -0400	2014-08-21 21:29:50 -0400	2014-08-21 21:29:50 -0400

### Issue a Command Against a Single Task Instance

Either:

- Right-click a task instance to display an [Action menu](#) of available commands for that task instance.
- Click a task instance to display the Task Instance Details below the list, or click the Details icon to display a Details pop-up for the task instance, and then right-click in the Details to display an [Action menu](#) of available commands for that task instance.

### Issue a Command Against Multiple Task Instances

Press **Ctrl** and right-click each task instance that you want to issue a command against to display an [Action menu](#) of available commands that can be issued against all of the selected task instances.

(You also can press **Ctrl** and right-click a single task instance and then press **Shift** and right-click another task instance to select the group of task instances between the first and second task instance, inclusive.)

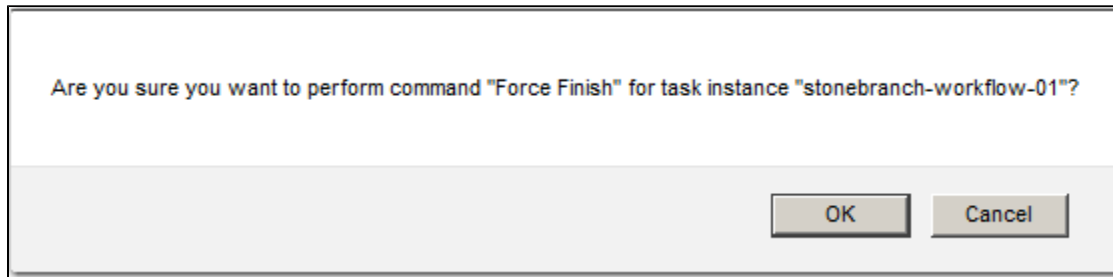
## Issuing Commands from the Workflow Monitor

From the Workflow Monitor, you can issue a command against a single task instance within the workflow or against the workflow task instance itself.

### Command Confirmation

If you want to receive a confirmation message after issuing a command but before the command is performed, set the [System Default Confirm Task Instance Commands](#) Universal Controller system property to **Yes** (the default is **No**).

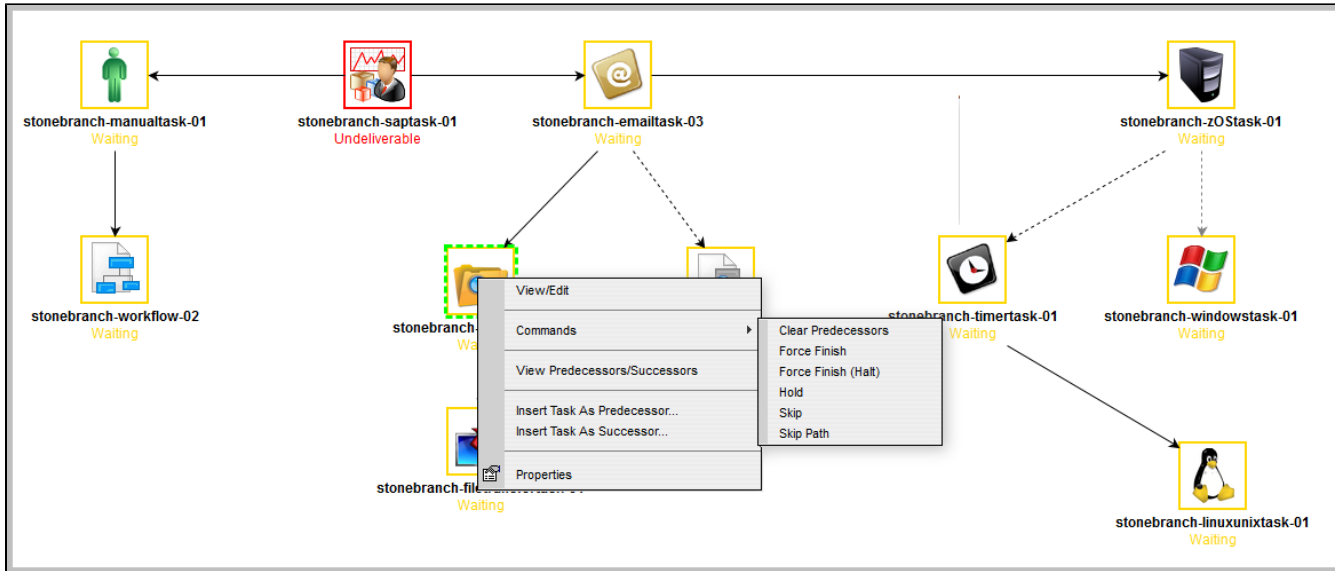
For example:



### Issue a Command Against a Task Instance within the Workflow

<b>Step 1</b>	Right-click the task instance to display a pop-up menu of commands appropriate for the selected task instance.
<b>Step 2</b>	Click <b>Commands</b> and then click the command that you want to issue against the selected task instance.

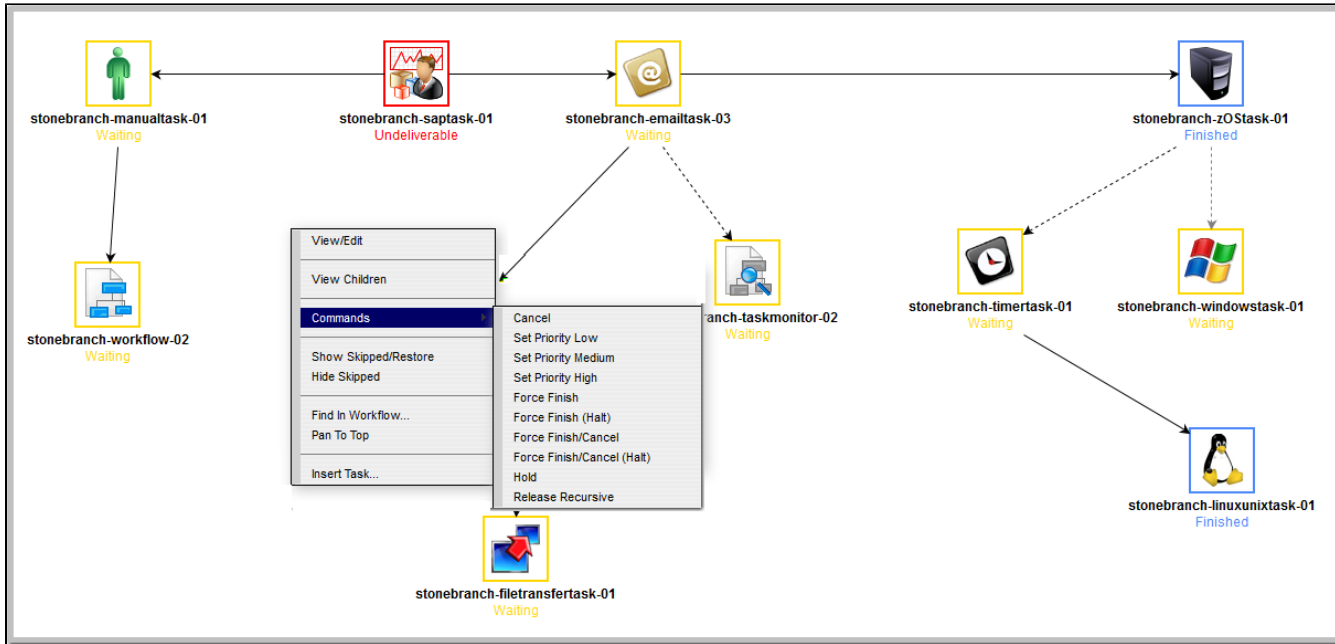
For example:



## Issue a Command Against the Workflow Task Instance

<b>Step 1</b>	Right-click the Workflow Monitor canvas to display a pop-up menu of commands appropriate for the workflow task instance.
<b>Step 2</b>	Click <b>Commands</b> and then click the command that you want to issue against the workflow task instance.

For example:



## Commands Supported for Task Instance Statuses

The following table identifies all possible task instance statuses, the task types they are valid for, and the commands that you can issue against a task instance in each status.

For a description of each status, see [Task Instance Status Types](#).

For a description of each command, see [Issuing Commands Against Task Instances](#).

For details and instructions on issuing these commands, see the specific section (below) on this page.

Status	Task Type	Supported Commands
Action Required (60)	Manual	<ul style="list-style-type: none"> <li>• Cancel</li> <li>• Force Finish</li> <li>• Force Finish (Halt)</li> <li>• Force Finish/Cancel</li> <li>• Force Finish/Cancel (Halt)</li> <li>• Set Started</li> <li>• Set Completed</li> </ul>

Cancel Pending (99)	Agent-based	<ul style="list-style-type: none"> <li>• Force Finish</li> <li>• Force Finish (Halt)</li> <li>• Retrieve Output</li> </ul>
Cancelled (130)	All	<ul style="list-style-type: none"> <li>• Force Finish</li> <li>• Force Finish (Halt)</li> <li>• Re-run - Not applicable for Workflow tasks.</li> <li>• Re-run (Suppress Intermediate Failures) - Not applicable for Workflow tasks.</li> <li>• Retrieve Output</li> </ul>
Confirmation Required (125)	z/OS	<ul style="list-style-type: none"> <li>• Force Finish</li> <li>• Force Finish (Halt)</li> <li>• Re-run - Not applicable for Workflow tasks.</li> <li>• Re-run (Suppress Intermediate Failures) - Not applicable for Workflow tasks.</li> <li>• Retrieve Output</li> </ul>
Defined (0)	All	<ul style="list-style-type: none"> <li>• Clear All Dependencies</li> <li>• Clear Predecessors</li> <li>• Clear Time Wait/Delay</li> <li>• Force Finish</li> <li>• Force Finish (Halt)</li> <li>• Hold</li> <li>• Skip</li> <li>• Skip Path</li> <li>• Release Recursive - Workflow tasks only.</li> </ul>
Exclusive Requested (22)	All	<ul style="list-style-type: none"> <li>• Clear All Dependencies</li> <li>• Force Finish</li> <li>• Force Finish (Halt)</li> <li>• Hold</li> <li>• Skip</li> <li>• Skip Path</li> </ul>
Exclusive Wait (23)	All	<ul style="list-style-type: none"> <li>• Clear All Dependencies</li> <li>• Clear Exclusive</li> <li>• Force Finish</li> <li>• Force Finish (Halt)</li> <li>• Hold</li> <li>• Skip</li> <li>• Skip Path</li> <li>• Release Recursive - Workflow tasks only.</li> </ul>

Execution Wait (33)	Agent-based	<ul style="list-style-type: none"> <li>• Force Finish</li> <li>• Force Finish (Halt)</li> <li>• Hold</li> <li>• Skip</li> <li>• Skip Path</li> </ul>
Failed (140)	All (except Workflows)	<ul style="list-style-type: none"> <li>• Force Finish</li> <li>• Force Finish (Halt)</li> <li>• Re-run - Not applicable for Workflow tasks.</li> <li>• Re-run (Suppress Intermediate Failures) - Not applicable for Workflow tasks.</li> <li>• Retrieve Output</li> </ul>
Finished (190)	All	<ul style="list-style-type: none"> <li>• Re-run - Not applicable for Workflow tasks.</li> <li>• Re-run (Suppress Intermediate Failures) - Not applicable for Workflow tasks.</li> <li>• Retrieve Output</li> </ul>
Held (20)	All	<ul style="list-style-type: none"> <li>• Clear All Dependencies</li> <li>• Clear Time Wait/Delay</li> <li>• Clear Predecessors</li> <li>• Force Finish</li> <li>• Force Finish (Halt)</li> <li>• Release</li> <li>• Release Recursive - Workflow tasks only.</li> <li>• Skip</li> <li>• Skip Path</li> </ul>
In Doubt (110)	Agent-based	<ul style="list-style-type: none"> <li>• Force Finish</li> <li>• Force Finish (Halt)</li> <li>• Re-run - Not applicable for Workflow tasks.</li> <li>• Re-run (Suppress Intermediate Failures) - Not applicable for Workflow tasks.</li> <li>• Retrieve Output</li> </ul>
Queued (40)	Agent-based	<ul style="list-style-type: none"> <li>• Cancel</li> <li>• Force Finish</li> <li>• Force Finish (Halt)</li> <li>• Hold</li> <li>• Set Priority</li> </ul>
Resource Requested (25)	All tasks using Virtual Resources	<ul style="list-style-type: none"> <li>• Clear All Dependencies</li> <li>• Force Finish</li> <li>• Force Finish (Halt)</li> <li>• Hold</li> <li>• Skip</li> <li>• Skip Path</li> </ul>

Resource Wait (30)	All tasks using Virtual Resources	<ul style="list-style-type: none"> <li>• Clear All Dependencies</li> <li>• Clear Resources</li> <li>• Force Finish</li> <li>• Force Finish (Halt)</li> <li>• Hold</li> <li>• Skip</li> <li>• Skip Path</li> <li>• Release Recursive - Workflow tasks only.</li> </ul>
Running (80)	All	<ul style="list-style-type: none"> <li>• Cancel</li> <li>• Force Finish</li> <li>• Force Finish (Halt)</li> <li>• Force Finish/Cancel</li> <li>• Force Finish/Cancel (Halt)</li> <li>• Release Recursive - Workflow tasks only.</li> <li>• Retrieve Output</li> <li>• Set Priority</li> </ul>
Running Problems (81)	Workflow	<ul style="list-style-type: none"> <li>• Cancel</li> <li>• Force Finish</li> <li>• Force Finish (Halt)</li> <li>• Force Finish/Cancel</li> <li>• Force Finish/Cancel (Halt)</li> <li>• Hold</li> <li>• Release Recursive - Workflow tasks only.</li> </ul>
Skipped (180)	All	<ul style="list-style-type: none"> <li>• Unskip</li> </ul>
Start Failure (120)	All	<ul style="list-style-type: none"> <li>• Force Finish</li> <li>• Force Finish (Halt)</li> <li>• Re-run - Not applicable for Workflow tasks.</li> <li>• Re-run (Suppress Intermediate Failures) - Not applicable for Workflow tasks.</li> <li>• Retrieve Output</li> </ul>
Started (70)	<a href="#">Agent-based</a> and Manual	<ul style="list-style-type: none"> <li>• Cancel</li> <li>• Force Finish</li> <li>• Force Finish (Halt)</li> <li>• Force Finish/Cancel</li> <li>• Force Finish/Cancel (Halt)</li> <li>• Retrieve Output</li> <li>• Set Completed - Manual tasks only.</li> <li>• Set Priority</li> </ul>
Submitted (43)	z/OS	<ul style="list-style-type: none"> <li>• Force Finish</li> <li>• Force Finish (Halt)</li> </ul>

Success (200)	All	<ul style="list-style-type: none"> <li>• Re-run - Not applicable for Workflow tasks.</li> <li>• Re-run (Suppress Intermediate Failures) - Not applicable for Workflow tasks.</li> <li>• Retrieve Output</li> </ul>
Time Wait (15)	All (except Timer)	<ul style="list-style-type: none"> <li>• Clear All Dependencies</li> <li>• Clear Time Wait/Delay</li> <li>• Force Finish</li> <li>• Force Finish (Halt)</li> <li>• Hold</li> <li>• Skip</li> <li>• Skip Path</li> <li>• Release Recursive - Workflow tasks only</li> </ul>
Undeliverable (35)	Agent-based	<ul style="list-style-type: none"> <li>• Force Finish</li> <li>• Force Finish (Halt)</li> <li>• Hold</li> <li>• Skip</li> <li>• Skip Path</li> </ul>
Waiting (10)	All	<ul style="list-style-type: none"> <li>• Clear All Dependencies</li> <li>• Clear Predecessors</li> <li>• Clear Time Wait/Delay</li> <li>• Force Finish</li> <li>• Force Finish (Halt)</li> <li>• Hold</li> <li>• Skip</li> <li>• Skip Path</li> <li>• Release Recursive - Workflow tasks only</li> </ul>

## Agent-Based Task Types

The following task types are Agent-based task types:

- Linux/Unix
- Windows
- z/OS
- Universal Command
- SAP
- PeopleSoft
- File Transfer
- Agent File Monitor
- Remote File Monitor
- System Monitor
- Universal

## Manually Launching a Task

Two methods are available for manually launching a task:

- From a tasks list
- From task Details

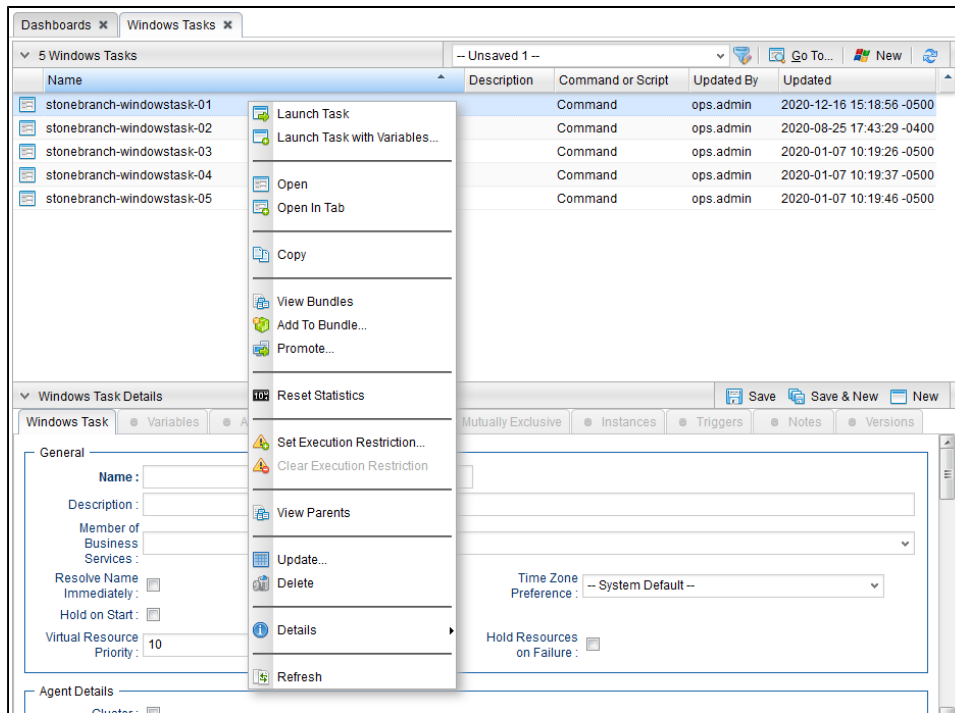
## Launch One or More Tasks from a Tasks List


**Step 1** From the **Automation Center** navigation pane, select **Tasks ><type of task>**. The Tasks list for that task type displays.

**Step 2** Either:

- Right-click a single task.
- Ctrl-click multiple tasks, release the Ctrl key, and right-click any of the selected tasks.

An **Action menu** displays.



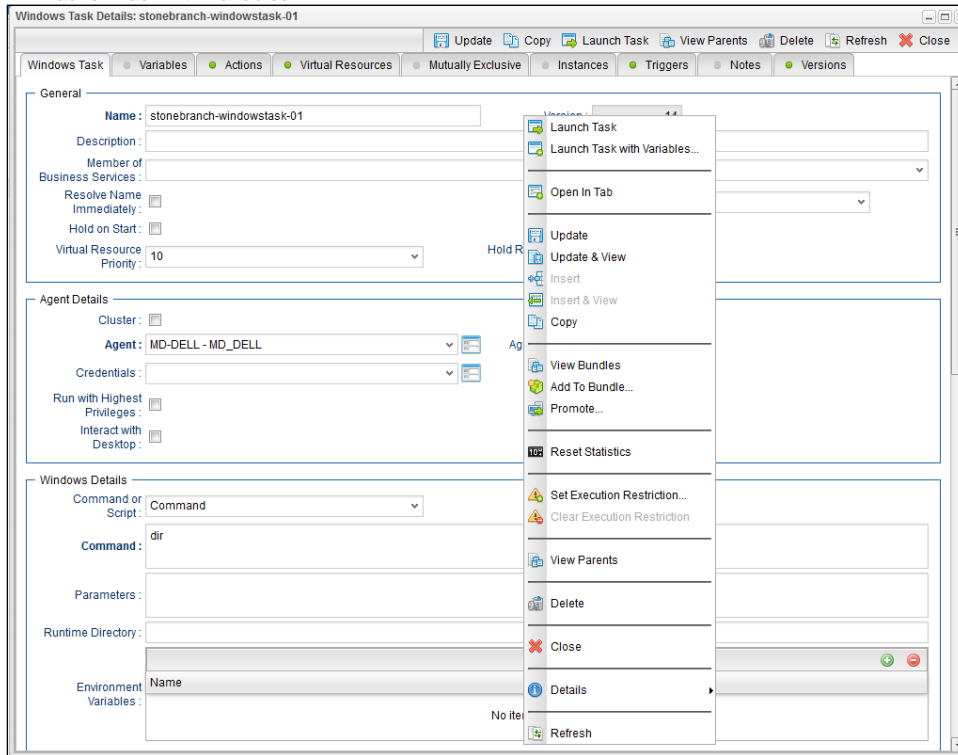
<b>Step 3</b>	Select either: <ul style="list-style-type: none"><li>• <b>Launch Task</b></li><li>• <a href="#">Launch Task with Variables</a></li></ul> Note  If you selected multiple tasks, <b>Launch with Variables</b> is disabled.  Universal Controller creates an instance of each selected task and runs it.
<b>Step 4</b>	To view Details about running task instances, select <b>Task Instances &gt; Activity</b> from the <b>Automation Center</b> navigation pane and click on the task instance.

## Launch a Task from Task Details

<b>Step 1</b>	Select the task you want to launch.
---------------	-------------------------------------

**Step 2** Either:

- Click the **Launch Task** button.
- Right-click anywhere in the Details to display an **Action menu** and click either:
  - **Launch Task**
  - **Launch Task with Variables**



## Launch a Task Manually with Temporary Variable Values

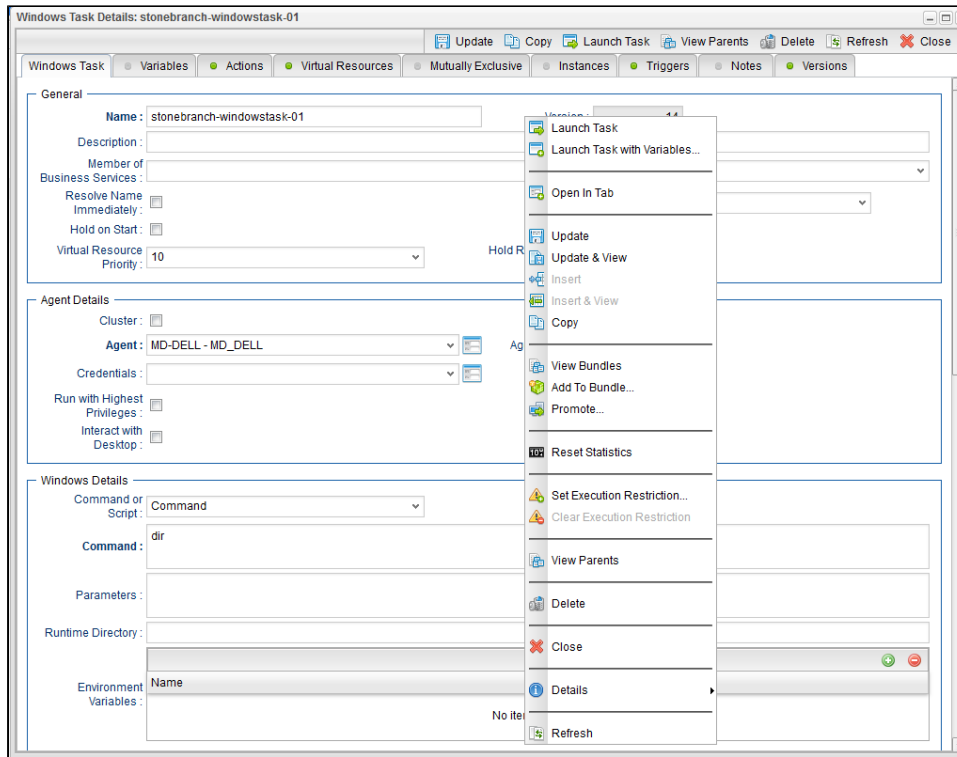
The Controller supports a Launch with Variables feature that allows you to quickly provide values for the variables specified in the task and launch it. All task types support the Launch with Variables feature.

(In the following procedure, the task is a Workflow already set up with variables where required.)

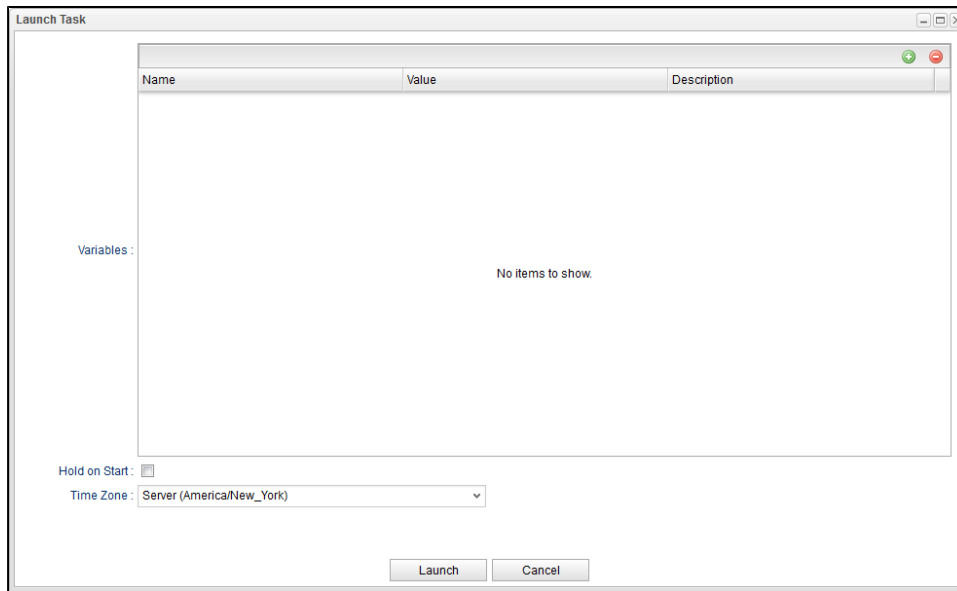
To launch a task using Launch with Variables:

- Step 1** Display the task you want to launch.

**Step 2** Access the **Action** menu.



**Step 3** Select **Launch Task with Variables...** The Launch Task (with Variables) dialog displays. Any variables attached to this task automatically are displayed in alphabetic order (a-z).



**Step 4** As needed, set the variable values or add new variables.

**Step 5** If you want to put the task instance in **held** status when the task is started, select **Hold on Start**. A **Hold Reason** field then displays which allows you to enter information about why the task will be p

**Step 6** If the task is a Workflow (as shown here), and:

- The Time Zone Preference field in the Workflow Details = Inherited, or the Time Zone Preference field in the Workflow Details = --System Default-- and the [Task Timezone Preference Universal](#) = Inherited, the dialog shows the **Time Zone** field.
- The Time Zone Preference field in the Workflow Details = Server, or the Time Zone Preference field in the Workflow Details = --System Default-- and the [Task Timezone Preference Universal C](#) Server, the dialog does not show the **Time Zone** field.

**Time Zone** lets you select a time zone for this specific launch of the Workflow so that it runs, and evaluates both Run Criteria and Execution Restrictions, according to that time zone.

By default, the value of **Time Zone** is Server.

**Step 7** When you are finished, click **Submit**. The Controller populates the variables with the values you supplied and launches the task.

## Variable Resolution

If the [Task Field Resolution Required](#) Universal Controller system property is set to true, and there is a variable in a resolvable field - such as Command, Script, Parameters (including z/OS Parameters), and Environment Variables - that is unresolved at runtime, the task will transition to [Start Failure](#).

A resolution check is not performed for variables used in the following places:

1. Workflow Task
2. Notification

3. PeopleSoft Task: Run Time Parameters
4. Stored Procedure Task: Stored Procedure Parameters
5. Email Template
6. Email Monitor: Advanced Criteria
7. Abort Action
8. Set Variable Action
9. Wait/Delay Options
10. Recurring Task: Override Variables

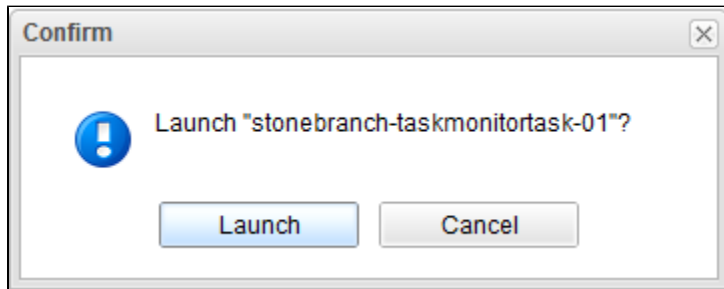
Note



For a Recurring task, the task will transition to Start Failure only if there are unresolved variables in the Recurring task definition itself. Unresolved variables in the target task are irrelevant when determining if the Recurring task should transition to Start Failure.

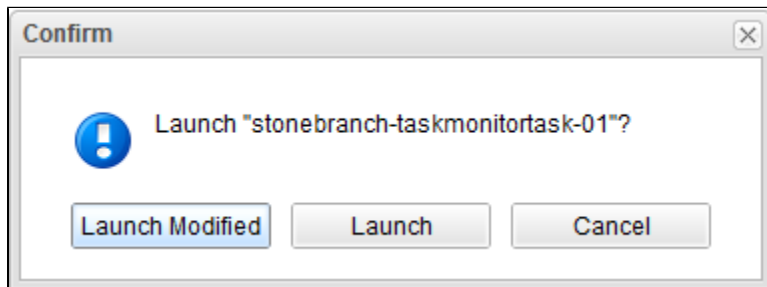
## Launch a Task Using Unsaved Modifications

When you select to launch a task, a dialog displays that ask you to confirm that task that you want to launch:



You also can launch a task that you have modified without saving the modifications.

When you select to launch a modified but unsaved task (single task only), the confirmation dialog specifies that you are about to launch a task with unsaved modifications:



## Changing the Priority of a Task Instance

You can change the priority of a task instance so that it will run sooner or later, as described below.

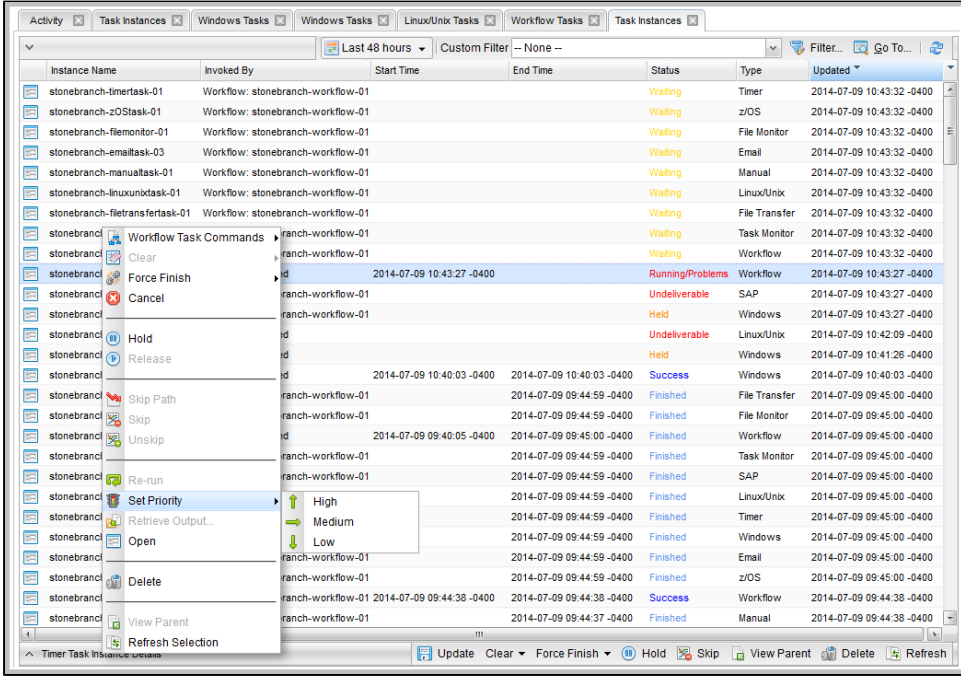
The priority specified here is meaningful only in relation to the priority setting of other tasks sent to that Agent from the same Controller instance.

You can change the priority of a Linux/Unix, Universal, Windows, or z/OS task instance while it is in any of the following statuses: Started, Running, Queued.

Two methods are available for changing the priority of a task instance:

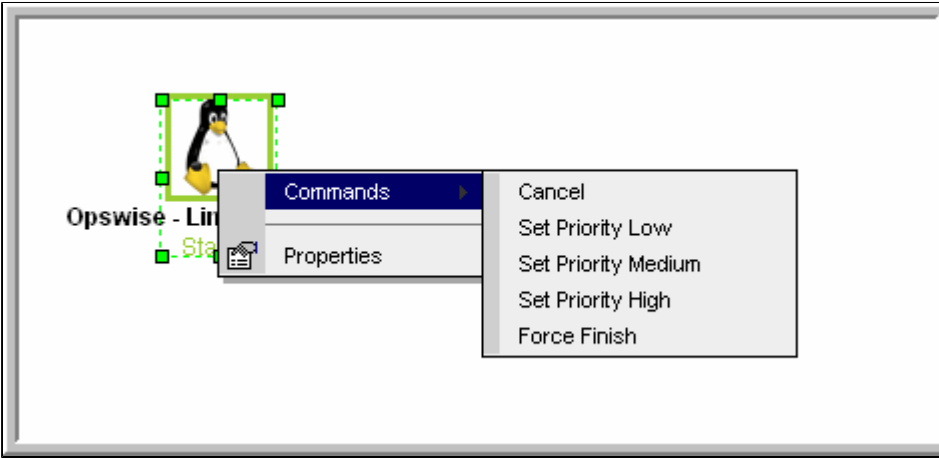
- From the Activity Monitor or Task Instances list
- From the Workflow Monitor

## Set Priority on a Task Instance from the Activity Monitor or Task Instances List

<b>Step 1</b>	Select the <a href="#">task instance</a> for which you want to set the priority.
<b>Step 2</b>	Right-click the task instance and, on the <a href="#">Action menu</a> , select the priority level.
 <p>The screenshot shows a table of task instances with columns: Instance Name, Invoked By, Start Time, End Time, Status, Type, and Updated. A context menu is open over a task instance, and the 'Set Priority' option is selected, showing a sub-menu with 'High', 'Medium', and 'Low' options.</p>	
<b>Step 3</b>	<p>When the Set Priority command has been executed, the Controller displays the following message at the bottom of the list:</p> <p><b>Command Set Priority &lt;High/Medium/Low&gt; successfully against task instance &lt;name&gt;.</b></p>

## Set Priority on a Task Instance from the Workflow Monitor

<b>Step 1</b>	View the Workflow that contains the task instance for which you want to set the priority.
<b>Step 2</b>	Select the <a href="#">task instance</a> for which you want to set priority.

<b>Step 3</b>	<b>Select Commands.</b> 
<b>Step 4</b>	<b>Select a priority for the task instance.</b>

## Re-running a Task Instance

If a task instance is part of a Workflow, you can re-run the task instance as long as the task instance and the Workflow have not been deleted.

If a task instance is not part of a Workflow, you can re-run the task instance as long as the task instance has not been deleted.

To qualify for re-run, a task instance must be in one of the following statuses: Success, Start Failure, Failed, Cancelled, Finished.


Additionally, you can re-run a task instance in the [In Doubt](#) status if the [Allow In Doubt Re-run](#) Universal Controller system property is set to true.

To [suppress intermediate failures](#) during a manual re-run, use Re-run (Suppress Intermediate Failures) instead of Re-run.

Typically, you can re-run multiple task instances; however, task instances in the In Doubt status must be re-run one instance at a time.

You can re-run a task even if it already is scheduled for [automatic retry](#). The retry attempt counts as one of the scheduled retries.

### Note

 You cannot re-run a Workflow task instance. However, you can [re-insert](#) a sub-Workflow into an active Workflow task instance in order to re-run it.

When you re-run a task instance, the Controller 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 on the [History list](#) (one for each time it ran).

Two methods are available for re-running a task instance:

- From the Activity Monitor or Task Instances list
- From the Workflow Monitor (if the task instance is running as part of a Workflow)

## Suppressing Intermediate Failures

If a task instance is in the [Failed status](#) following the usage of Re-run (Suppress Intermediate Failures), the following will be suppressed.

- All Actions ([Abort](#), [Email Notification](#), [Set Variable](#), [SNMP Notification](#), and [System Operation](#)) defined for the task instance on a Failed status.
- Workflow conditional path processing; any Successors waiting on a failure path will not be released.
- Task Monitors will not be notified of the Failed status. Also, any [Task Monitor](#) task that has a [Time Scope](#) in the past will disqualify any matching task instance in the past with a Failed status resulting from Re-run (Suppress Intermediate Failures).
- Any Workflow containing the Failed task instance will not transition to the [Running/Problems](#) status.

## Re-run a Task Instance from the Activity Monitor or Task Instances List

<b>Step 1</b>	Select the <a href="#">task instance</a> you want to re-run.
<b>Step 2</b>	Click <b>Re-run</b> . The task status changes to the next appropriate status as though it had just been launched.


## Re-run a Task Instance from the Workflow Monitor

<b>Step 1</b>	View the Workflow that contains the task instance you want to re-run.
<b>Step 2</b>	Select the <a href="#">task instance</a> you want to re-run.
<b>Step 3</b>	Select <b>Commands</b> .
<b>Step 4</b>	Select <b>Re-run</b> . The task status changes to the next appropriate status as though it had just been launched, and the Workflow Console opens to display information about the re-run.

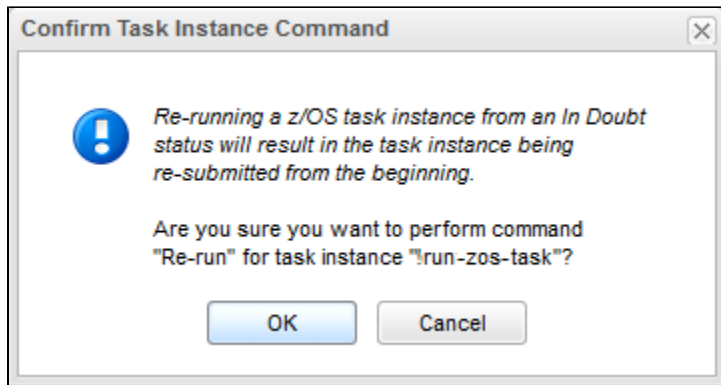
## Re-run a z/OS Task Instance in the In Doubt Status

If a z/OS task instance is in the In Doubt status, re-running the task instance will re-submit it from the beginning, which is equivalent to performing an Insert Task of the same task into the Workflow except, in this case, the UUID will not change.

Note

 The standard clean-up of sequential datasets still applies to avoid a NOTCAT2 condition. For information on how to prevent automatic data set deletion from occurring when the task instance is re-submitted, see [Disabling Automatic Data Set Deletion](#).

If you choose to re-run a z/OS task instance from an In Doubt status, a confirmation pop-up displays to make sure that you are aware that the task instance will be re-submitted from the beginning.



## Note



The confirmation for z/OS Re-run from an In Doubt status will happen regardless of the [System Default Confirm Task Instance Commands](#) Universal Controller system property value.

## Cancelling a Task Instance

The Cancel command cancels a running task instance.

You can cancel a task instance while it is in any of the following statuses: Queued, Action Required, Started, Running.

If the task instance is part of a Workflow, you also can Cancel the task instance if the task instance is re-run after the Workflow completes.

For tasks that run on Agents, including Windows, Linux, Unix, z/OS, FTP, Agent File Monitor, and Universal Command tasks, the Cancel command is sent to the Agent.

- If the task instance has not yet been launched, it does not launch.
- If the task instance already has been launched, the Agent cancels it, if possible.
- If the task instance is a Workflow, any of its task instances in Running status go to Cancelled status; the Workflow itself goes to Running/Problems status.
- If the task instance is in a Workflow, the Workflow goes to Running/Problems status. If the task is re-run, the Workflow returns to Running status.

## Note



For Extension-based Universal Task Instances, the Controller allows you to forcibly execute the Cancel command while already in Cancel Pending, which immediately terminates the extension process and any custom cancel logic being executed.

Two methods are available for cancelling a task instance:

- From the Activity Monitor or Task Instances list
- From the Workflow Monitor (if the task instance is running as part of a Workflow)

## Note



Cancelling a Web Service task instance with Protocol = SOAP is not supported.

Cancelling a PeopleSoft task instance cancels the PeopleSoft process itself, not the PeopleSoft task process. Once the Peoplesoft process has been cancelled, its status will filter through to the PeopleSoft task.

## Cancel a Task Instance from the Activity Monitor or Task Instances List

<b>Step 1</b>	Select the <a href="#">task instance</a> you want to cancel.
<b>Step 2</b>	Click <b>Cancel</b> . The task status changes to Cancelled.

## Cancel a Task Instance from the Workflow Monitor

<b>Step 1</b>	Open the Workflow Monitor for the workflow that contains the task instance you want to cancel.
<b>Step 2</b>	Select the <a href="#">task instance</a> .
<b>Step 3</b>	Select <b>Commands</b> .
<b>Step 4</b>	Select <b>Cancel</b> . The task status changes to Cancelled and the Workflow Console opens to display information about the cancellation.

## Force Finishing a Task Instance

The Force Finish command puts a task instance into the Finished status, regardless of what the task instance is doing.

You can Force Finish a task instance 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.

If the task instance is part of a Workflow, you also can Force Finish the task instance if the task instance is re-run after the Workflow completes.

### Note

The Force Finish command also cancels a Universal Task instance if **Always Cancel On Force Finish** is enabled for its Universal Template.

Since force finishing a Workflow task instance also force finishes its children, Force Finish also cancels a child Universal Task instance if **Always Cancel On Force Finish** is enabled for its Universal Template.

Although Force Finish sets the status of a task instance to Finished, the associated process (if any) will continue to run. For example, when force finishing a Windows task, the process will not be canceled by the agent and will continue to run until completion.

One purpose of Force Finish is to allow successor task instances in a workflow to launch without waiting for the current task instance to complete. You also may want to Force Finish a stand-alone task instance; for example, you may want to mark a failed job as Finished, rather than rerunning the job.

If a task instance is running when the user issues a Force Finish, the Controller marks the task instance as Finished even though the actual process continues running. Two exceptions are the Agent File Monitor and Remote File Monitor; for these task types, the monitoring processes are aborted by a Force Finish command. Assuming they have no other dependencies, all successor task instances waiting for successful completion of this task instance will start.

When you issue a Force Finish against a Workflow, the Workflow and any of its tasks that are not already in Success, Finished, or Skipped status will go to Finished status.

Two methods are available for Force Finishing a task instance:

- From the Activity Monitor or Task Instances list
- From the Workflow Monitor (if the task instance is running as part of a Workflow)

## Force Finish a Task Instance from the Activity Monitor or Task Instances List

<b>Step 1</b>	Select the <a href="#">task instance</a> you want to Force Finish.
<b>Step 2</b>	Click <b>Force Finish</b> . The task status changes to Finished.

## Force Finish a Task Instance from the Workflow Monitor

<b>Step 1</b>	View the workflow that contains the task instance you want to Force Finish.
<b>Step 2</b>	Select the <a href="#">task instance</a> .
<b>Step 3</b>	Select <b>Commands</b> .
<b>Step 4</b>	Select <b>Force Finish</b> . The task status changes to Finished and the Workflow Console opens to display information about the Force Finish.

## Force Finishing (Halt) a Task Instance

Just as with the [Force Finish](#) command, the Force Finish (Halt) command puts a task instance into the Finished status, regardless of what the task instance is doing.

You can Force Finish (Halt) a task instance 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.

If the task instance is part of a Workflow, you also can Force Finish (Halt) the task instance if the task instance is re-run after the Workflow completes.

However, Force Finish (Halt) prevents successor task instances in a Workflow from being run. Those tasks will not run until you re-run the task against which you had executed Force Finish (Halt).

Although Force Finish sets the status of a task instance to Finished, the associated process (if any) will continue to run. For example, when force finishing a Windows task, the process will not be canceled by the agent and will continue to run until completion.

If a task instance is running when the user issues a Force Finish (Halt), the Controller marks the task instance as Finished even though the actual process continues running. Two exceptions are the Agent File Monitor and Remote File Monitor; for these task types, the monitoring processes are aborted by a Force Finish (Halt) command. All successor task instances waiting for successful completion of this task instance will remain in Waiting status.

Similarly, task monitors are not released if a Force Finish (Halt) is executed against a task being monitored.

### Note



There are two areas in the user interface that you can check to determine if a task instance was forced finished with halt:

- [Additional Information](#) field in the [Audit Details](#) for that force finished task instance.
- [Status Description](#) field in the [Task Instance Details](#) for that force finished task instance.

Two methods are available for Force Finishing (Halt) a task instance:

- From the Activity Monitor
- From the Workflow Monitor (if the task instance is running as part of a Workflow)

## Force Finish (Halt) a Task Instance from the Activity Monitor or Task Instances List

<b>Step 1</b>	Select the <a href="#">task instance</a> you want to Force Finish (Halt).
<b>Step 2</b>	Click <b>Force Finish (Halt)</b> . The task status changes to Finished.

## Force Finish (Halt) a Task Instance from the Workflow Monitor

<b>Step 1</b>	View the workflow that contains the task instance you want to Force Finish (Halt).
<b>Step 2</b>	Select the <a href="#">task instance</a> .
<b>Step 3</b>	Select <b>Commands</b> .
<b>Step 4</b>	Select <b>Force Finish (Halt)</b> . The task status changes to Finished and the Workflow Console opens to display information about the Force Finish (Halt).

## Force Finish/Canceling a Task Instance

The Force Finish/Cancel command cancels a task instance and puts it into Finished status, regardless of what the task instance is doing.

You can Force Finish/Cancel a task instance while it is in any of the following statuses: Queued, Action Required, Started, Running.

If the task instance is part of a Workflow, you also can Force Finish/Cancel the task instance if the task instance is re-run after the Workflow completes.

Force Finish/Cancel sets the status of a task instance to Finished and cancels the associated process (if any).

One purpose of Force Finish/Cancel is to cancel a task instance and allow successor task instances in a Workflow to launch without waiting for that task instance to complete. You also may want to Force Finish/Cancel a stand-alone task instance; for example, you may want to mark a failed job as Finished, rather than rerunning the job.

### Note



The Force Finish/Cancel command is not implemented for Timer tasks, since for this type of task, the Cancel and Force Finish commands essentially perform the same function.

For tasks that run on Agents, including Windows, Linux, Unix, z/OS, FTP, Agent File Monitor, and Universal Command tasks, the Force Finish/Cancel command is sent to the Agent.

- If the task instance has not yet been launched, it does not launch.
- If a task instance is running when the user issues a Force Finish/Cancel command, the Agent cancels the task instance, if possible, and then the Controller marks the task instance as Finished; processing does not continue. Assuming they have no other dependencies, all successor task instances waiting for successful completion of this task instance will start.
- If the task instance is a workflow, any eligible task instances in the workflow are cancelled and set to the Finished status, and then the workflow itself is set to the Finished status.

Two methods are available for Force Finish/Canceling a task instance:

- From the Activity Monitor or Task Instances list
- From the Workflow Monitor (if the task instance is running as part of a Workflow)

## Force Finish/Cancel a Task Instance from the Activity Monitor

<b>Step 1</b>	Select the <a href="#">task instance</a> you want to Force Finish/Cancel.
<b>Step 2</b>	Click <b>Force Finish/Cancel</b> . The task status changes to Finished.

## Force Finish/Cancel a Task Instance from the Workflow Monitor

<b>Step 1</b>	View the workflow that contains the task instance you want to Force Finish/Cancel.
<b>Step 2</b>	<a href="#">Select the task instance.</a>
<b>Step 3</b>	Select <b>Commands</b> .
<b>Step 4</b>	Select <b>Force Finish/Cancel</b> . The task status changes to Finished and the Console opens to display information about the Force Finish/Cancel.

## Force Finish/Cancelling (Halt) a Task Instance

Just as with the [Force Finish/Cancel](#) command, the Force Finish/Cancel (Halt) command cancels a task instance and puts it into Finished status, regardless of what the task instance is doing.

If the task instance is part of a Workflow, you also can Force Finish/Cancel (Halt) the task instance if the task instance is re-run after the Workflow completes.

Force Finish/Cancel sets the status of a task instance to Finished and cancels the associated process (if any).

However, Force Finish/Cancel (Halt) prevents successor task instances in a Workflow from being run. Those tasks will not run until you re-run the task against which you had executed Force Finish/Cancel (Halt).

Task monitors are not released if a Force Finish/Cancel (Halt) is executed against a task being monitored.

You can Force Finish/Cancel (Halt) a task instance while it is in any of the following statuses: Queued, Action Required, Started, Running.

### Note



The Force Finish/Cancel (Halt) command is not implemented for Timer tasks, since for this type of task, the Cancel and Force Finish commands essentially perform the same function.

For tasks that run on Agents, including Windows, Linux, Unix, z/OS, FTP, Agent File Monitor, and Universal Command tasks, the Force Finish/Cancel (Halt) command is sent to the Agent.

- If the task instance has not yet been launched, it does not launch.
- If a task instance is running when the user issues a Force Finish/Cancel (Halt) command, the Agent cancels the task instance, if possible, and then the Controller marks the task instance as Finished; processing does not continue. All successor task instances waiting for successful completion of this task instance remain in Waiting status.
- If the task instance is a workflow, any eligible task instances in the workflow are cancelled and set to the Finished status, and then the workflow itself is set to the Finished status.

### Note



There are two areas in the user interface that you can check to determine if a task instance was forced finish / cancelled with halt:

- [Additional Information](#) field in the [Audit Details](#) for that force finish / cancelled task instance.
- [Status Description](#) field in the [Task Instance Details](#) for that force finish / cancelled task instance.

Two methods are available for Force Finish/Cancelling (Halt) a task instance:

- From the Activity Monitor
- From the Workflow Monitor (if the task instance is running as part of a Workflow)

## Force Finish/Cancel (Halt) a Task Instance from the Activity Monitor

<b>Step 1</b>	Select the <a href="#">task instance</a> you want to Force Finish/Cancel (Halt).
<b>Step 2</b>	Click <b>Force Finish/Cancel (Halt)</b> . The task status changes to Finished.

## Force Finish/Cancel (Halt) a Task Instance from the Workflow Monitor

<b>Step 1</b>	View the workflow that contains the task instance you want to Force Finish/Cancel (Halt).
<b>Step 2</b>	Select the <a href="#">task instance</a> .
<b>Step 3</b>	Select <b>Commands</b> .
<b>Step 4</b>	Select <b>Force Finish/Cancel (Halt)</b> . The task status changes to Finished and the Console opens to display information about the Force Finish/Cancel (Halt).

## Putting a Task Instance on Hold

If you put a Workflow on hold that has not yet started, the Workflow and all the task instances in it are put on hold.

If you put a Workflow on hold when it is in Running status, all the task instances within the Workflow that have not yet started are put on hold; however, the Workflow itself does not go to Hold status because it already has started.

To release the Workflow and all of its task instances that are on hold, issue the Release Recursive command against the Workflow.

To release the Workflow but keep the task instances on hold until you release them one by one, use Release on the Workflow first, then use Release on each task instance.

You can put a task instance on hold while it is in any of the following statuses: Defined, Waiting, Resource Wait, Queued.

Two methods are available for putting a task instance on hold:

- From the Activity Monitor or Task Instances list
- From the Workflow Monitor (if the task instance is running as part of a Workflow)

## Hold a Task Instance from the Activity Monitor or Task Instances List

<b>Step 1</b>	Select the <a href="#">task instance</a> you want to put on hold.
<b>Step 2</b>	Click <b>Hold</b> . The task status changes to Held.

## Hold a Task Instance from the Workflow Monitor

<b>Step 1</b>	View the workflow that contains the task instance you want to put on hold.
<b>Step 2</b>	Select the <a href="#">task instance</a> .
<b>Step 3</b>	Select <b>Commands</b> .

<b>Step 4</b>	Select <b>Hold</b> . The task status changes to Held and the Workflow Console opens to display information about the hold.
---------------	--

## Releasing a Task Instance from Hold

For Workflows, if the user held a Workflow that already was running, only the task instances 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.

You can release a non-Workflow task instance from hold from the Activity Monitor or Task Instances list while it is in the following status: Held.

To release the Workflow, use one of the following commands:

- To release the entire held Workflow and its task instances, use **Release Recursive**.
- To release a Workflow that is not in Held status but has task instances that are in Held status, use **Release Recursive**. In this case, you can issue **Release Recursive** on a Workflow in any of the following statuses: Defined, Waiting, Held, Resource Wait, Running.
- To release the Workflow but keep the task instances inside on hold so that you can release them one by one, use **Release**. In this case, release the Workflow first, then release each task instance manually.

Two methods are available for releasing a task instance from hold:

- From the Activity Monitor or Task Instances list
- From the Workflow Monitor (if the task instance is running as part of a Workflow)

## Release a Held Task Instance from the Activity Monitor or Task Instances List

<b>Step 1</b>	Select the <a href="#">task instance</a> you want to release from hold.
---------------	---

<b>Step 2</b>	Click <b>Release</b> or <b>Release Recursive</b> . The task status changes to the next appropriate status according to where it was in processing at the time it was put on hold.
---------------	---

## Release a Held Task Instance from the Workflow Monitor

<b>Step 1</b>	View the Workflow that contains the task instance you want to release.
---------------	--

<b>Step 2</b>	Select the <a href="#">task instance</a> .
---------------	--

<b>Step 3</b>	Select <b>Commands</b> .
---------------	--------------------------

<b>Step 4</b>	Select <b>Release</b> . The task status changes to the next appropriate status according to where it was in processing, and the Workflow Console opens to display information about the release.
---------------	--

## Skipping a Task Instance

You can skip a task instance or a task instance path so that the task instance and all of its dependent task instances automatically are skipped as well.

You can skip any task instance as long as it has not yet started running; that is, while it is in any of the following statuses: Defined, Waiting, Held, Resource Requested, Resource Wait.

Two methods are available for skipping a task instance:

- From the Activity Monitor or Task Instances list

- From the Workflow Monitor (if the task instance is running as part of a Workflow)

Note



You also can specify that a task instance will be skipped (before the task or its Workflow is launched) by:

1. Modifying a trigger Details (using the trigger's Skip Count field) so that the Controller skips the next N number of trigger occurrences for launching the task.
2. Modifying a Workflow Details by specifying [conditional paths](#) that may place one or more task instances in the Skipped status when the workflow is run.
3. Modifying a Workflow Details by specifying that one or more task instances should be skipped (or run) at specific times (see [Adding Skip/Run Criteria for Specific Tasks](#)).

If you skip a Workflow task instance, all the task instances within the Workflow also are skipped, along with any nested Workflows.

Once a task instance has been skipped, the only command you can run against it is [Unskip](#).

## Skip a Task Instance from the Activity Monitor or Task Instances List

<b>Step 1</b>	Select the <a href="#">task instance</a> you want to skip.
<b>Step 2</b>	Click <b>Skip</b> . The task status changes to Skipped.
<b>Step 3</b>	To skip the task instance and all of its dependent task instances, click <b>Skip Path</b> . The task status of the task instance and all of its dependent task instances changes to Skipped.

## Skip a Task Instance from the Workflow Monitor

<b>Step 1</b>	View the Workflow that contains the task instance you want to skip.
<b>Step 2</b>	Select the <a href="#">task instance</a> .
<b>Step 3</b>	Select <b>Commands</b> .
<b>Step 4</b>	Select <b>Skip</b> . The task status changes to Skipped, and the Console opens to display information about the skip.
<b>Step 5</b>	To skip the task instance and all of its dependent task instances, click <b>Skip Path</b> . The task status of the task instance and all of its dependent task instances changes to Skipped, and the Console opens to display information about the skip.

## Showing or Hiding Skipped Task Instances

You can select whether to show or hide skipped task instances on the Workflow Monitor either:

- Before the Workflow is running
- While the Workflow is running

Three methods are available for selecting whether or not to show or hide skipped task instances:

- From the Workflow Task Details
- From the Workflow Task Instance Details
- From the Workflow Monitor

## Setting Show / Hide Skipped Tasks from the Workflow Task Details

<b>Step 1</b>	Display the Workflow Task Details for the Workflow that you want to show/hide Skipped task instances.
<b>Step 2</b>	Use the Show / Hide Skipped Tasks field to select whether you want to show or hide skipped task instances (default is Show Skipped). When viewing a running Workflow in the Workflow Monitor, the skipped task instances will be shown or hidden based on your selection.

## Setting Show / Hide Skipped Tasks from the Workflow Task Instance Details

<b>Step 1</b>	Display the Workflow Task Instance Details for the Workflow task instance that you want to show/hide Skipped task instances.
<b>Step 2</b>	Use the Show / Hide Skipped Tasks field to select whether you want to show or hide skipped task instances (default is Show Skipped). When viewing the Workflow instance in the Workflow Monitor, the skipped task instances will be shown or hidden based on your selection.

## Setting Show / Hide Skipped Tasks from the Workflow Monitor

Open the Workflow task instance in the Workflow Monitor. By default, the Workflow Monitor will show or hide skipped task instances based on the Workflow task instance's Show / Hide Skipped Tasks option.

To temporarily change the behavior, right-click in the Workflow Monitor canvas and select either of the following entries from the pop-up menu:

- Show Skipped / Restore
- Hide Skipped

## Unskipping a Task Instance

If a task instance in a Workflow has been skipped (perhaps at trigger time due to run criteria or manually by running the skip command), you can unskip that task instance while the Workflow is running.

Note



If you unskip a task instance that was skipped by issuing a Skip Path command against it, which automatically skip all of its dependent tasks, those dependent tasks stay in Skipped status. You must manually unskip each task to remove them from Skipped status.

Two methods are available for unskipping a task instance:

1. From the Activity Monitor or Task Instances list
2. From the Workflow Monitor

### Unskip a Task Instance from the Activity Monitor or Task Instances List

<b>Step 1</b>	Select the <a href="#">task instance</a> you want to unskip.
<b>Step 2</b>	Click <b>Unskip</b> . The task instance will run when all of its dependencies have been satisfied.

### Unskip a Task Instance from the Workflow Monitor

<b>Step 1</b>	View the Workflow that contains the task instance(s) you want to unskip.
<b>Step 2</b>	Select a <a href="#">task instance</a> . (You can issue commands only against one task at a time within the Workflow Monitor.)

<b>Step 3</b>	Select <b>Commands</b> .
<b>Step 4</b>	Select <b>Unskip</b> . A confirmation message will appear in the Console, and the task instance will run when all of its dependencies have been satisfied.

## Marking a Dependency as Satisfied

For task instances running inside of a Workflow, you can mark a single predecessor dependency as satisfied to allow the task instance to run.

Marking a dependency as satisfied has the same result as [clearing a dependency](#).

You can mark a dependency as satisfied on task instances in the following status: Defined, Waiting, Held.

One method is available for marking a dependency as satisfied:

- From the Workflow Monitor

### Marking a Dependency as Satisfied from the Workflow Monitor

<b>Step 1</b>	View the Workflow that contains the task instance whose dependencies you want to satisfy.
<b>Step 2</b>	Locate and right-click on the task dependency (the connector line between two tasks).
<b>Step 3</b>	Select <b>Commands</b> .
<b>Step 4</b>	Select <b>Mark as Satisfied</b> . If all other dependencies are satisfied, the task instance is launched normally.

## Clearing Predecessor Dependencies of a Task Instance

For a task instance running inside of a Workflow, you can clear all predecessor dependencies to allow that task instance to run. Clearing a predecessor dependency has the same result as [satisfying a predecessor dependency](#).

You can clear predecessor dependencies of a task instance while it is in any of the following statuses: Defined, Waiting, Held.

Note



Clearing predecessor dependencies does not include the clearing of resource and mutually exclusive dependencies. To clear these dependencies, see [Clearing Resource Dependencies of a Task Instance](#) and [Clearing Mutually Exclusive Dependencies of a Task Instance](#), below. To clear all dependencies, see [Clearing All Dependencies of a Task Instance](#), below.

One method is available for clearing predecessor dependencies of a task instance:

- From the Workflow Monitor

### Clearing Predecessor Dependencies of a Task Instance from the Workflow Monitor

<b>Step 1</b>	View the Workflow that contains the task instance whose predecessor dependencies you want to satisfy.
---------------	---

<b>Step 2</b>	Select the <a href="#">task instance</a> for which you want to clear predecessor dependencies.
<b>Step 3</b>	Select <b>Commands</b> .
<b>Step 4</b>	Select <b>Clear Predecessors</b> . The task instance is launched normally.

## Clearing Resource Dependencies of a Task Instance

For task instances for which resources have been defined, you can clear those resource dependencies.

You can clear resource dependencies of task instances while it is in the following status: Resource Wait.

Two methods are available for clearing resource dependencies from task instances:

1. From the Activity Monitor or Task Instances list
2. From the Workflow Monitor

### Clear Resource Dependencies of a Task Instance from the Activity Monitor or Task Instances List

<b>Step 1</b>	Select the <a href="#">task instance</a> whose resources you want to clear.
<b>Step 2</b>	Click <b>Clear Resources</b> . Resource dependencies are cleared from the task instance.

### Clear Resource Dependencies of a Task Instance from the Workflow Monitor

<b>Step 1</b>	View the Workflow that contains the task instance(s) you want to clear of resource dependencies.
<b>Step 2</b>	Select a <a href="#">task instance</a> . (You can issue commands only against one task at a time within the Workflow Monitor.)
<b>Step 3</b>	Select <b>Commands</b> .
<b>Step 4</b>	Select <b>Clear Resources</b> . A confirmation message will appear in the Console, and the task instance will run without resources.

## Clearing Mutually Exclusive Dependencies of a Task Instance

For task instances that are mutually exclusive with other task instances, you can clear those mutually exclusive dependencies.

Any task instances that were mutually exclusive with this task instance will no longer be mutually exclusive.

You can clear mutually exclusive dependencies of a task instance while it is in the following status: Exclusive Wait.

Two methods are available for clearing mutually exclusive dependencies from task instances:

1. From the Activity Monitor or Task Instances list
2. From the Workflow Monitor

## Clear Mutually Exclusive Dependencies of a Task Instance from the Activity Monitor or Task Instances List

<b>Step 1</b>	Select the <a href="#">task instance</a> whose mutually exclusive dependencies you want to clear.
<b>Step 2</b>	Click <b>Clear Exclusive</b> . Mutually exclusive dependencies of the task instance are cleared, and the task instance is launched normally.

## Clear Mutually Exclusive Dependencies of a Task Instance from the Workflow Monitor

<b>Step 1</b>	View the Workflow that contains the task instance you want to clear of mutually exclusive dependencies.
<b>Step 2</b>	Select a <a href="#">task instance</a> . (You can issue commands only against one task at a time within the Workflow Monitor.)
<b>Step 3</b>	Select <b>Commands</b> .
<b>Step 4</b>	Select <b>Clear Exclusive</b> . A confirmation message will appear in the Console, and the task instance will run normally.

## Clearing All Dependencies of a Task Instance

You can clear all dependencies (time wait/delay, predecessors, resources, and exclusive) to allow a task instance to run.

You can clear all dependencies of a task instance while it is in any the following status: Defined, Waiting, Held, Exclusive Requested, Exclusive Wait, Resource Requested, Resource Wait.

Three methods are available for clearing all dependencies of a task instance:

1. From the Activity Monitor or Task Instances list
2. From the Workflow Monitor (for a task instance within the Workflow)
3. From the Workflow Monitor (for the Workflow itself)

If you issue a Clear All Dependencies command against a Workflow task instance, all dependencies of only the Workflow task instance are cleared, not the dependencies of any of its task instances.

## Clear All Dependencies of a Task Instance from the Activity Monitor or Task Instances List

<b>Step 1</b>	Select the <a href="#">task instance</a> whose dependencies you want to clear.
<b>Step 2</b>	Click <b>Clear All Dependencies</b> . All dependencies are cleared from the task instance and it is launched normally.

## Clear All Dependencies of a Task Instance within a Workflow from the Workflow Monitor

<b>Step 1</b>	View the Workflow that contains the task instance whose dependencies you want to clear.
<b>Step 2</b>	Select the <a href="#">task instance</a> for which you want to clear predecessor dependencies.
<b>Step 3</b>	Select <b>Commands &gt; Clear All Dependencies</b> . All dependencies are cleared from the task instance and it is launched normally.

## Clear All Dependencies of a Workflow Task Instance from the Workflow Monitor

<b>Step 1</b>	View the Workflow whose dependencies you want to clear.
<b>Step 2</b>	Right-click the Workflow Monitor canvas to display a menu of available actions.
<b>Step 3</b>	Select <b>Commands &gt; Clear All Dependencies</b> . All dependencies are cleared from the Workflow.

## Clearing Time Wait/Delay Specifications of a Task Instance

You can clear all Wait To Start and Delay On Start specifications of a task instance to allow a task instance to run without waiting.

You can clear all Wait To Start and Delay On Start specifications of a task instance while it is in any the following status: Defined, Waiting, Time Wait, Held.

Three methods are available for clearing Time Wait/Delay specifications of task instances:

1. From the Activity Monitor or Task Instances list
2. From the Workflow Monitor
3. From the task instance Details

## Clear All Time Wait/Delay Specifications of a Task Instance from the Activity Monitor or Task Instances List

<b>Step 1</b>	Select the <a href="#">task instance</a> whose Time Wait/Delay specifications you want to clear.
<b>Step 2</b>	Click <b>Clear Time Wait/Delay</b> . The task instance is launched normally.

## Clear All Time Wait/Delay Specifications of a Task Instance from the Workflow Monitor

<b>Step 1</b>	View the Workflow that contains the task instance whose Time Wait/Delay specifications you want to clear.
<b>Step 2</b>	Select the <a href="#">task instance</a> for which you want to clear Time Wait/Delay specifications.
<b>Step 3</b>	Select <b>Commands</b> .
<b>Step 4</b>	Select <b>Clear Time Wait/Delay</b> . The task instance is launched normally.

## Clear All Time Wait/Delay Specifications of a Task Instance from the Task Instance Details

<b>Step 1</b>	Open the Task Instance Details for the task instance that you want to clear Time Wait/Delay specifications.
<b>Step 2</b>	Right-click in the task instance Details and select <b>Clear &gt; Clear Time Wait/Delay</b> .

# Monitoring Activity from the Task Instances List

- [Overview](#)
- [Displaying the Task Instances List](#)
  - [Task Instances List Column Descriptions](#)
- [Task Instances List Task Bar](#)
  - [Time Constraint](#)
  - [Filtering](#)
- [Displaying Task Instance Details](#)
- [Issuing Commands Against Task Instances](#)

## Overview

The Task Instances list displays the same task instance information as the [Activity Monitor](#), but only for task instances for which there has been a status change or a modification to the task instance record.

Also, unlike the Activity Monitor, the Task Instances list is not automatically refreshed.

You also can monitor activity for a specific task by displaying [task-specific Task Instances Details](#).

## Displaying the Task Instances List

From the [Automation Center](#) navigation pane, select **Task Instances > All Task Instances**. The Task Instances list displays.

Instance Name	Type	Status	Invoked By
workflow-regression-one-of-each-tasks-Full	Workflow	Success	Manually Launched
B-09058_WebService_OMS_Servers	Workflow	Success	Workflow: workflow-regression-one-of-each-tasks-Full
B-09058_WebService_OMS_Server_Modify	Web Service	Success	Workflow: B-09058_WebService_OMS_Servers
B-09058_WebService_OMS_Server_List	Web Service	Success	Workflow: B-09058_WebService_OMS_Servers
B-09058_WebService_OMS_Server_Read	Web Service	Success	Workflow: B-09058_WebService_OMS_Servers
B-09058_WebService_OMS_Server_Create	Web Service	Success	Workflow: B-09058_WebService_OMS_Servers
B-09058_WebService_OMS_Server_Delete	Web Service	Success	Workflow: B-09058_WebService_OMS_Servers
workflow-qa-regression-main	Workflow	Success	Workflow: workflow-regression-one-of-each-tasks-Full
B-09067_Triggers_WF-1	Workflow	Success	Workflow: workflow-qa-regression-main
B-09067_Triggers_Email_Mon_WF-B-09067_Email_Mon_Trigger	Workflow	Success	Workflow: B-09067_Triggers_WF-1
B-09067_Triggers_List_Read_Delete_WF-B-09067_Email_Mon_Trigger	Workflow	Success	Workflow: B-09067_Triggers_Email_Mon_WF-1
B-09067_Triggers_Delete-JSON-B-09067_Email_Mon_Trigger	Web Service	Success	Workflow: B-09067_Triggers_List_Read_Delete_WF-1
B-09067_Triggers_Read-JSON-B-09067_Email_Mon_Trigger	Web Service	Success	Workflow: B-09067_Triggers_List_Read_Delete_WF-1
B-09067_Triggers_List-JSON-Email Monitor	Web Service	Success	Workflow: B-09067_Triggers_List_Read_Delete_WF-1
B-09067_Triggers_Read-B-09067_Email_Mon_Trigger	Web Service	Success	Workflow: B-09067_Triggers_List_Read_Delete_WF-1
B-09067_Triggers_List-Email Monitor	Web Service	Success	Workflow: B-09067_Triggers_List_Read_Delete_WF-1
B-09067_Triggered_Task	Timer	Success	Trigger: B-09067_Email_Mon_Trigger
B-09067_Triggers_Commands_WF-B-09067_Email_Mon_Trigger	Workflow	Success	Workflow: B-09067_Triggers_Email_Mon_WF-1
B-09067_Triggers_Trigger_Now-JSON-B-09067_Email_Mon_Trigger	Web Service	Success	Workflow: B-09067_Triggers_Commands_WF-1
B-09067_Triggered_Task	Timer	Success	Trigger: B-09067_Email_Mon_Trigger

## Task Instances List Column Descriptions

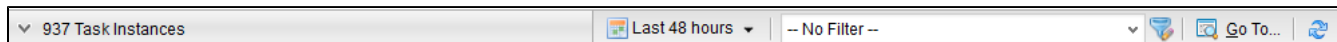
The following table describes the default columns of information displayed on the Task Instances list.

Column	Description
Instance Name	Name of this task instance.
Invoked By	System-supplied; how the task instance was launched. One of the following: <ul style="list-style-type: none"> <li>• Trigger: (Trigger Name) - The instance was launched by the named trigger.</li> <li>• Workflow: (Workflow Name) - The instance was launched by the named workflow.</li> <li>• Manually Launched - The instance was launched by a user. To determine the name of the user:                             <ol style="list-style-type: none"> <li>1. From the Activity Monitor or Task Instances list, click the task instance name to open the record.</li> <li>2. The Execution User field identifies the user who launched the task instance.</li> </ol> </li> </ul>
Start Time	Date and time the task instance started.

End Time	Date and time the task instance ended.
Status	Current <a href="#">status</a> of the task instance.
Type	Type of task instance.
Updated	Date and time this task instance ended or was last updated.

## Task Instances List Task Bar

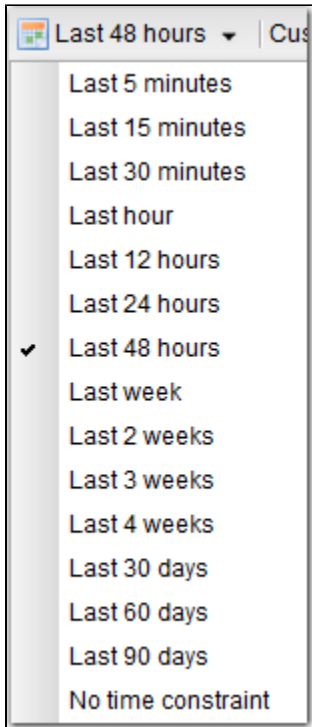
A Task Instances List task bar displays across the top of the [Task Instances list](#), which allows you to select which task instances display on the list.



## Time Constraint

The Time Constraint drop-down list in the Task Instance Selections task bar allows you to select a time frame for which you want task instances to display on the list.

The default time constraint, as specified by the [Task Instances Time Constraint](#) user preference, is 48 hours.



## Filtering

The **Filter** field and **Filter** icon in the Task Instance Selections task bar allows you to filter the list so that only task instances matching the specified criteria within the selected time constraint display on the list.



See [Filters](#) for detailed information on applying and saving filters.

## Displaying Task Instance Details

To display Details for a [specific task instance](#), either:

- Click the **Instance Name** of the task instance to display the Details below the list.
- Click the **Details** icon next to the **Instance Name** of the task instance to display a Details pop-up.

From the task Instance Details, you can also display additional Details about the task instance:

<a href="#">Show Metadata</a>	Displays Metadata for the task instance, including a status history.
<a href="#">Show Details</a>	Displays complete database details for the task instance.
<a href="#">Show Variables</a>	Displays a list of all variables available to the task instance, including any variables inherited from the parent or embedded (sub-Workflow) Workflow of the task instance.

## Issuing Commands Against Task Instances

Where applicable, you can manually intervene in processing by [issuing a command](#) against one or more task instances.

For information about the commands available for each type of task, see [Supported Commands](#).

# Monitoring Activity History

- [Overview](#)
- [Displaying the History List](#)
- [History List Task Bar](#)
  - [Time Constraint](#)
  - [Filtering](#)
- [Displaying History Details](#)
  - [History Details Field Descriptions](#)

## Overview

The History list ([ops\\_history](#)) provides an historical display of all completed task activity. Only task instances with a status in an "end state" (SUCCESS, FINISHED, FAILED, CANCELLED, START FAILURE, SKIPPED) display in the History list.

The list allows you to track information about specific task instances, including multiple runs. For example, Task A may have failed and then was re-run by a user. This task instance will display twice on the History list, first for the time that it ran and failed and again for the time it was re-run to success.

Note



If you want to display task activity for all tasks, for task instances in any status, and issue commands against those task instances, see the [Activity Monitor](#) and/or [Task Instances list](#)

If you want to display task activity for a specific task, for task instances in any status, and issue commands against those task instances, see the [Viewing Task Instances for a Specific Task](#).

## Displaying the History List

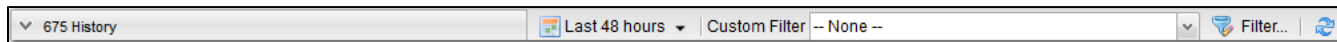
**Step 1** From the [Automation Center](#) navigation pane, select **Task Instances > History**. The History list displays.

Instance Name	Instance Number	Type	Status	Start Time	End Time	Duration	Waited for Reso
B_08587_System_Functions_\$(...)	4	Manual	Skipped		2019-06-28 11:31:08 -0400		No
B_08587_System_Functions_\$(...)	1	Manual	Skipped		2019-06-28 11:31:17 -0400		No
email-task-built-in-variables	1	Email	Success	2019-06-28 11:30:56 -0400	2019-06-28 11:30:56 -0400	0 Seconds	No
ftp-filemon-simple-variable	1	FTP File Monitor	Success	2019-06-28 11:30:58 -0400	2019-06-28 11:31:00 -0400	2 Seconds	No
indesa-task-simple-variables	1	Universal Command	Success	2019-06-28 11:30:23 -0400	2019-06-28 11:30:24 -0400	1 Seconds	No
Linux check for vsFTP#QUERY#	1	Application Control	Failed				No
Linux check for vsFTP#QUERY#	1	Application Control	Finished		2019-06-28 11:31:01 -0400		No
no-filemon-simple-variable	1	File Monitor	Success	2019-06-28 11:30:57 -0400	2019-06-28 11:30:57 -0400	0 Seconds	No
no-task-launch-simple-variables	1	Linux/Unix	Success	2019-06-28 11:30:20 -0400	2019-06-28 11:30:21 -0400	0 Seconds	No
Opswise - Sleep 10	1	Timer	Success	2019-06-28 11:33:47 -0400	2019-06-28 11:33:57 -0400	10 Seconds	No
sap-task-simple-variable	1	SAP	Success	2019-06-28 11:30:25 -0400	2019-06-28 11:30:37 -0400	12 Seconds	No
setvar_rmt_target	1	Universal Command	Success	2019-06-28 11:30:17 -0400	2019-06-28 11:30:18 -0400	1 Seconds	No
Sleep 10	4	Timer	Success	2019-06-28 11:31:42 -0400	2019-06-28 11:31:52 -0400	10 Seconds	No
Sleep 10	2	Timer	Success	2019-06-28 11:31:42 -0400	2019-06-28 11:31:53 -0400	10 Seconds	No
Sleep 10	1	Timer	Success	2019-06-28 11:31:45 -0400	2019-06-28 11:31:55 -0400	10 Seconds	No
Sleep 10	3	Timer	Success	2019-06-28 11:32:01 -0400	2019-06-28 11:32:11 -0400	10 Seconds	No
Sleep 10	5	Timer	Success	2019-06-28 11:32:24 -0400	2019-06-28 11:32:34 -0400	11 Seconds	No
Sleep 10	6	Timer	Success	2019-06-28 11:34:08 -0400	2019-06-28 11:34:18 -0400	10 Seconds	No
Sleep Variable	1	Timer	Success	2019-06-28 11:30:41 -0400	2019-06-28 11:30:48 -0400	7 Seconds	No
sql-task-mysq-select-all	1	SQL	Success	2019-06-28 11:30:48 -0400	2019-06-28 11:30:52 -0400	4 Seconds	No
sql-task-mysq-sproc1	1	Stored Procedure	Success	2019-06-28 11:30:54 -0400	2019-06-28 11:30:55 -0400	0 Seconds	No
stonebranch-filemonfortask-01	1	File Monitor	Finished	2019-07-03 10:06:27 -0400	2019-07-03 10:07:02 -0400	35 Seconds	No
sysmon-diskpace-simple-variable	1	System Monitor	Success	2019-06-28 11:31:01 -0400	2019-06-28 11:31:01 -0400	0 Seconds	No
Taskmon-workflow-simple	1	Task Monitor	Success	2019-06-28 11:30:56 -0400	2019-06-28 11:30:56 -0400	0 Seconds	No
udm-task-simple-variable	1	File Transfer	Success	2019-06-28 11:30:39 -0400	2019-06-28 11:30:40 -0400	1 Seconds	No
variable-monitor-simple	1	Variable Monitor	Success	2019-06-28 11:30:16 -0400	2019-06-28 11:30:16 -0400	0 Seconds	No
web-service-task-simple	1	Web Service	Success	2019-06-28 11:30:52 -0400	2019-06-28 11:30:54 -0400	2 Seconds	No
win-task-launch-simple-variables	1	Windows	Skipped		2019-06-28 11:30:19 -0400		No
workflow-qa-regression-main	1	Workflow	Success	2019-06-28 11:31:02 -0400	2019-06-28 11:37:46 -0400	6 Minutes 44 Seconds	No
workflow-regression-one-of-each-tasks-Full	1	Workflow	Success	2019-06-28 11:30:16 -0400	2019-06-28 11:37:48 -0400	7 Minutes 33 Seconds	No
zos-task-launch-simple-variables	1	z/OS	Skipped		2019-06-28 11:30:19 -0400		No

**Step 2** You can modify the display by filtering, sorting, adding, and removing columns (see [Record Lists](#)).

## History List Task Bar

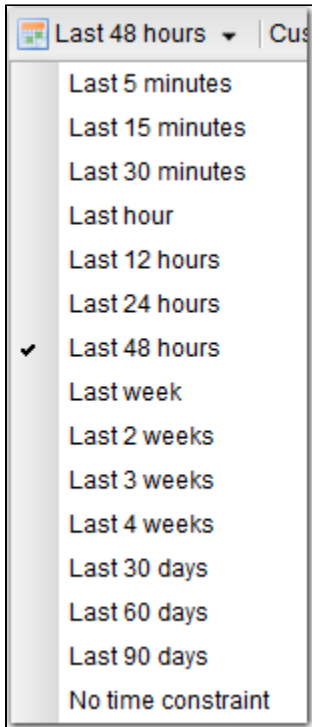
A History List task bar displays across the top of the [History list](#), which allows you to select which task instances display on the list.



## Time Constraint

The Time Constraint drop-down list in the History List Selections task bar allows you to select a time frame for which you want task instances to display on the list.

The default time constraint, as specified by the [History Time Constraint](#) user preference, is 48 hours.



## Filtering

The **Filter** field and **Filter** icon in the History List Selections task bar allows you to filter the list so that only task instances matching the specified criteria within the selected time constraint display on the list.



See [Filters](#) for detailed information on applying and saving filters.

## Displaying History Details

To display execution details about any task instance on the History list, either:

- Click anywhere in the task instance row to display the Details for that task instance below the list.
- Click the icon next to any **Instance Name** to display a pop-up version of the Details.

History Details: stonebranch-agentfilemonitortask-01

History

Details

Instance Name : stonebranch-agentfilemonitortask-01

Description :

Member of Business Services : Accounting

Task Instance : stonebranch-agentfilemonitortask-01 Instance Number : 1

Task : stonebranch-agentfilemonitortask-01

Type : Agent File Monitor

Invoked By : Recurring Task (1): stonebranch-recurringtask-01

Launch Source : Recurring Source Instance : stonebranch-recurringtask-01

Trigger : Agent : qa-cntlr-mysql.stone.branch - qa-cntlr-mysql

Status : Finished Exit Code : 0

Status Description : State was changed from RUNNING to CANCEL PENDING -> State was forced from CANCEL PENDING to FINISHED

Start Time : 2021-05-18 09:56:48 -0400 End Time : 2021-05-25 11:43:22 -0400

Duration : 7 Days 1 Hour 46 Minutes 34 Seconds

Virtual Resource Priority : 10

Waited for Resources :

Waited for Exclusive :

Late Started :

Late Finished :


Status History :

Date	Status
2021-05-18 09:56:47 -0400	Defined
2021-05-18 09:56:47 -0400	Queued
2021-05-18 09:56:48 -0400	Running
2021-05-24 15:17:25 -0400	Queued
2021-05-24 15:17:26 -0400	Running
2021-05-25 11:43:22 -0400	Cancel Pending
2021-05-25 11:43:22 -0400	Finished

## History Details Field Descriptions

The following table describes the fields that display in History Details.

Field Name	Description
Instance Name	Text field; Name of this task instance.

Description	Description of this record. (Maximum = 255 characters.)
Member of Business Services	User-defined; Allows you to select one or more <a href="#">Business Services</a> that this record belongs to. (You also can Check All or Uncheck All Business Services for this record.) You can select up to 62 Business Services for any record type, and enter a maximum of 2048 characters for each Business Service.  If the <a href="#">Business Service Visibility Restricted</a> Universal Controller system property is set to true, depending on your assigned (or inherited) <a href="#">Permissions</a> or <a href="#">Roles</a> , Business Services available for selection may be restricted.
Task Instance	Reference field; Name of this task instance. If the task instance is deleted, this field is empty.
Instance Number	Sequentially assigned number, maintained per task, representing the creation order of the instance.
Task	Name of the task that was run to create this task instance. Click the icon to display Task Details for the task.
Type	Type of task that was run to create this task instance.
Universal Template	If <a href="#">Type</a> = Universal; Name of the Universal Template on which the Universal Task Type is based.
Invoked by	System-supplied; how the task instance was launched.  Options: <ul style="list-style-type: none"> <li>• Trigger: (Trigger Name) Instance was launched by the named trigger.</li> <li>• Workflow: (Workflow Name) Instance was launched by the named workflow.</li> <li>• Manually Launched Instance was launched by a user. To identify the user, check the Execution User column for that task instance on the <a href="#">Task Instances</a> screen or, on most task instance screens, the <a href="#">Execution User</a> field.</li> </ul>
Launch Source	Source from where this task instance's task was launched.
Source Instance	Task instance of the source task that launched this task instance's task.  Note  Only task instances with <a href="#">Launch Source</a> = Recurring will have a value for this field; otherwise, the field is hidden.
Trigger	Name of the trigger that launched this task.
Agent	Name of the <a href="#">Agent</a> resource that identifies the machine where the operation will run. If you do not specify an Agent, you must specify an <a href="#">#Agent Cluster</a> .
Status	System-supplied; see <a href="#">Task Instance Statuses</a> .
Exit Code	System-supplied; the exit code captured by the Agent when executing the task (for example, a command or script).

Status Description	System-supplied; additional information, if any, about the status of the task instance.
Start Time	System-supplied; Date and time the task instance started.
End Time	System-supplied; Date and time the task instance completed.
Duration	System-supplied; amount of time the task instance took to run.
Waited for Resources	Indication of whether or not the task instance ran only after it waited for a resource to become available.
Virtual Resource Priority	Priority for acquiring a resource when two or more tasks are waiting for the resource. This priority applies to all resources required by the task. Options: 1 (high) - 100 (low). Default is 10.
Waited for Exclusive	Indication of whether or not the task instance ran only after it waited for one or more tasks with which it was mutually exclusive to finish.
Late Started	If checked, the task started later than the time specified in the Late Start fields.
Late Finished	If checked, the task finished later than the time or duration specified in the Late Finish fields.
Status History	History of statuses for this task instance.

# Viewing Task Instances for a Specific Task

- [Introduction](#)
- [Displaying Task Instance Details](#)
- [Displaying Additional Task Instance Details](#)
- [Show Task Instance Variables](#)
  - [Show / Hide Global Variables](#)

## Introduction

From the Task Details of any task, you can display a list of task instances for that task and the Details of any specific task instance.

The list will display all task instances for which there has been a status change or a modification to the task instance record within the last 30 days.

You also can display Details of any specific task instance from the:

- [Activity Monitor](#)
- [Task Instances list](#)

## Displaying Task Instance Details

**Step 1** From the [Automation Center](#) navigation pane, select the task from the **Tasks > All Tasks** or **Tasks ><task type> Tasks** list. The Task Details for that task displays.

For example:

Linux/Unix Task Details: stonebranch-linuxunixtask-01

Update Copy Launch Task View Parents Delete Refresh Close

Linux/Unix Task Variables Actions Virtual Resources Mutually Exclusive Instances Triggers Notes Versions

**General**

Name: stonebranch-linuxunixtask-01 Version: 16

Description:

Member of Business Services:

Resolve Name Immediately:  Time Zone Preference: -- System Default --

Hold on Start:

Virtual Resource Priority: 10 Hold Resources on Failure:

**Agent Details**

Cluster:

Agent: qa-ctrl-mysql.stone.branch - qa-ctrl-mysql Agent Variable:

Credentials: Credentials Variable:

Run as sudo:

**Linux/Unix Details**

Command or Script: Command

Command: dir

Parameters:

Runtime Directory:

Environment Variables:

Name	Value
No items to show.	

**Result Processing Details**

Exit Code Processing: Success Exitcode Range

Exit Codes: 0

Automatic Output Retrieval: -- None --

**Retry Options**

Retry Exit Codes:

Maximum Retries: 0 Retry Indefinitely:

Retry Interval (Seconds): 60 Suppress Intermediate Failures:

**Wait/Delay Options**

Wait To Start: -- None --

Delay On Start: -- None --

Workflow Only: -- System Default --

**Time Options**

Late Start:  Late Start Type: Time

Late Start Time: Hour: 00 Min: 00

Late Start Day Constraint: -- None --

Late Finish:

Early Finish:

User Estimated Duration: Day: Hour: Min: Sec:

**Critical Path Options**

CP Duration: CP Duration Unit: Minutes

**Workflow Execution Options**

Execution: -- None --



**Step 2** Click the **Instances** tab. A list of instances for that task displays.

You can change the default time constraint for the display of task instances on the tab via the [Task Instances Tab Time Constraint](#) user preference.

Instance Name	Type	Status	Invoked By	Start Time	End Time	Updated
stonebranch-linuxunixtask-01	Linux/Unix	Finished	Workflow: stonebranch-workflow-01	2014-07-09 13:03:05 -0400	2014-07-09 13:03:05 -0400	2014-07-09 13:03:05 -0400
stonebranch-linuxunixtask-01	Linux/Unix	Finished	Workflow: stonebranch-workflow-01	2014-07-09 10:52:20 -0400	2014-07-09 10:52:20 -0400	2014-07-09 10:52:20 -0400
stonebranch-linuxunixtask-01	Linux/Unix	Finished	Workflow: stonebranch-workflow-01	2014-07-09 10:47:34 -0400	2014-07-09 10:47:34 -0400	2014-07-09 10:47:34 -0400
stonebranch-linuxunixtask-01	Linux/Unix	Finished	Workflow: stonebranch-workflow-01	2014-07-09 09:44:59 -0400	2014-07-09 09:45:00 -0400	2014-07-09 09:45:00 -0400

**Step 3** Click the Details icon next to any **Instance Name** on the list to display the Details for that task instance.

Linux/Unix Task Instance
Virtual Resources
Exclusive Requests
Output
Notes

**General**

Instance Name:  Instance Number:

Description:

Member of Business Services:

Task:  Source Version:

Launch Source:  Source Instance:

Invoked By:  Execution User:

Calendar:  Time Zone Preference:

Virtual Resource Priority:  Hold Resources on Failure:

**Status**

Status:  Exit Code:

Status Description:

Operational Memo:

Trigger Time: <input type="text"/>	Launch Time: <input type="text" value="2021-05-12 14:04:37 -0400"/>
Wait Until Time: <input type="text"/>	Queued Time: <input type="text" value="2021-05-12 14:12:45 -0400"/>
Start Time: <input type="text" value="2021-05-12 14:12:46 -0400"/>	End Time: <input type="text" value="2021-05-12 14:12:46 -0400"/>
Duration: <input type="text" value="0 Seconds"/>	CPU Time: <input type="text" value="242"/>
Process ID: <input type="text" value="17620"/>	

**Agent Details**

Cluster:

Agent:  Agent Variable:

Credentials:  Credentials Variable:

Run as sudo:

**Linux/Unix Details**

Command or Script:

Command:

Parameters:

Runtime Directory:

Environment Variables: 

Name	Value
No items to show.	

**Result Processing Details**

Exit Code Processing:

Exit Codes:

Automatic Output Retrieval:

**Retry Options**

Retry Exit Codes:

Maximum Retries:  Retry Indefinitely:

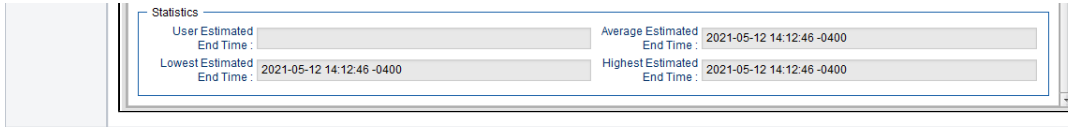
Retry Interval (Seconds):  Suppress Intermediate Failures:

Current Retry Count:

**Wait/Delay Options**

Wait To Start:

Delay On Start:



## Displaying Additional Task Instance Details

You can display additional Details for a task instance by right-clicking anywhere in the task instance to display an Action menu, and then selecting any of the following options from that menu:

<a href="#">Show Metadata</a>	Displays Metadata for the task instance, including a status history.
<a href="#">Show Details</a>	Displays complete database details for the task instance.
<a href="#">Show Variables</a>	Displays a list of all variables available to the task instance, including any variables inherited from the parent or embedded (sub-Workflow) Workflow of the task instance.

## Show Task Instance Variables

The [Action menu](#) for every task instance record [Details](#) contains a Details sub-menu. If you click **Show Variables** on this sub-menu, a Show Variables tab opens.

The Show Variables tab lists all variables available to the task instance. Variables inherited from the parent or embedded (sub-Workflow) Workflow of a task instance are listed in their own sections.

Dashboards Activity Show Variables: stonebranch-task-variables-01

Instance Name:	stonebranch-task-variables-01		
Status:	Running		
Status Description:			
Invoked By:	Workflow: stonebranch-workflow-variables-02		
Execution User:	variable_user_a		
UUID:	15011667358096523778SIEIVG8EX3J9		

Name	Value	Resolved Value	Inherited
<b>stonebranch-task-variables-01</b>			
DEFAULT_INIT_TOKEN	c82799be297c4eeb86a8fff5b9db9b8a		
INIT_TIME	2017-07-27 11:43:16 -0400		
ops_attempt	1		
ops_cluster_hostname	qa-cntrl-mysql.stone.branch		
ops_cluster_id	qa-cntrl-mysql.stone.branch:8080-qa_cntrl_mysql		
ops_cluster_ipaddr	192.168.31.42		
ops_cluster_mode	Active		
ops_cluster_name	qa-cntrl-mysql.stone.branch:8080-qa_cntrl_mysql		
ops_cluster_start_time	2017-07-27 10:45:35 -0400		
ops_cluster_uptime	5 Hours 17 Minutes 6 Seconds		
ops_custom_field1			
ops_custom_field2			
ops_execution_user	variable_user_a		
ops_exit_code	0		
ops_launch_time	2017-07-27 11:43:16 -0400		
ops_retry_count	0		
ops_retry_interval	60		
ops_retry_maximum	0		
ops_start_time	2017-07-27 11:43:16 -0400		
ops_status	RUNNING		
ops_status_description			
ops_system_identifier	qa-cntrl-mysql		
ops_task_id	15011667358096523778SIEIVG8EX3J9		
ops_task_name	stonebranch-task-variables-01		
ops_task_ref_count	10		
ops_task_type	Timer		
ops_task_type_value	2		
ops_top_level_workflow_id	1501166735809648377XH54VKQONXY3W		
ops_vertex_id	2		
ops_workflow_id	1501166735809650377X43GQLSDHKCLK		
ops_workflow_name	stonebranch-workflow-variables-02		
RANDOM	\$_random(99,1)}	36	
<b>stonebranch-workflow-variables-02</b>			
WF_2_NO	\$_multiply(\$_STONEBRANCH_SYSTEM_ID};20)}	4440	Yes
WF_2_SYSTEM_AND_DATE	WF_2_\${STONEBRANCH_SYSTEM_NAME}_\$_ WF_2_PRODUCTION_2017-07-27 16:03:38 -0400		Yes
<b>stonebranch-workflow-variables-01</b>			
DEFAULT_INIT_TOKEN	83a8b5133bd54cf6b8f9ea27a3c4e33		No

ops_trigger_name	stonebranch-workflow-variables-01 #IHIGER#	Yes
ops_trigger_time	2017-07-27 11:43:00 -0400	Yes
WF_1_NO	\$_multiply('\${STONEBRANCH_SYSTEM_ID};'10') 2220	Yes
WF_1_SYSTEM_AND_DATE	WF_1_\${STONEBRANCH_SYSTEM_NAME}_\$_... WF_1_PRODUCTION_2017-07-27 16:03:38 -0400	Yes
Global		
A_TOKEN	****	Yes
STONEBRANCH_SYSTEM_DATE_YYYY_MM_DD	\$_date('yyyy-MM-dd') 2017-07-27	Yes
STONEBRANCH_SYSTEM_ID	222	Yes
STONEBRANCH_SYSTEM_NAME	PRODUCTION	Yes

Field	Description
Name	Name of the variable.
Value	<p>Value of the variable.</p> <p>If the variable was defined by using the Variables tab on a trigger, task, or workflow (that is, a <a href="#">Local Variable</a>), and it is a composite of other variables and/or <a href="#">functions</a>, it will remain a composite of those variables and/or functions. Each time that variable is resolved, it will produce a dynamic resolution, which could differ from the previous resolution. The Resolved Value field will show what that variable would resolve to now.</p> <p>For example: <code>unique_id=\${ops_task_name}-\${ops_task_id}-\${ops_task_ref_count}-\${_date()}</code></p> <p>If the variable was defined by using a <a href="#">Set Variable action</a> for a task or workflow, and it is a composite of other variables, the value would be fully resolved.</p>
Resolved Value	<p>If the variable is a composite of other variables and/or functions; value that the variable would resolve to now.</p> <p>For example, using unique_id from the Value field, above: <code>unique_id=stonebranch-task-variables-01-15011667358096523778SIEIVG8EX3J9-10-2017-07-27 16:03:38-0400</code></p>
Inherited	<p>Indication (Yes or No) for whether a variable defined in the workflow hierarchy, or globally, would be inherited by the task instance.</p> <p>For example, if the task instance already has that variable defined within its own scope, or within a more direct workflow scope, the variable would not be inherited.</p> <p>(This field is not applicable for variables defined within the task instances own scope.)</p>
<b>Buttons</b>	This section identifies the buttons displayed below the list of variables.
Print	Allows you to print the Show Variables table.
Refresh	Refreshes the information in the Show Variables table.
Show Global	Shows Global variables in the table.
Hide Global	Hides Global variables in the table.

## Show / Hide Global Variables

You can toggle the visibility of Global variables using the Show Global or Hide Global button.

By default (the [Show Variables Fetch Global Automatically](#) Universal Controller system property default is No), Global variables are not fetched and displayed automatically in the Show Variables tab, since there could be a large number of Global variables, and only a few (or none) could be relevant to a specific task instance.

Furthermore, you may want to use the [Set Variable](#) action on a workflow in [Defined](#) status to populate the workflow instance with its own variables generated from Global variables, which often can be date/time related functions that need to be resolved at Trigger/Launch time.

Any user can change the default visibility of Global variables by using the [Show Variables Fetch Global Automatically](#) User Preference.

The following constraints apply when viewing Global variables from Show Variables.

- The task instance can only use (or Read) global variables for which the task instance Execution User has Variable Read permission for, therefore, the report will only display those global variables.
- If the user running the Show Variables report does not have Variable Read permission for a global variable in the report, the Value field will be masked with \*\*\*\*\*.
- If the user must be able to view a global variable value from the report, an administrator must grant the user the appropriate Variable Read permission.

# Displaying Task Instance Status

- [Displaying Task Instance Status](#)
- [Task Instance Status Types](#)
  - [Agent-Based Task Types](#)
- [Task Instance Status Colors](#)

## Displaying Task Instance Status

You can display the status of one or more task instances from the Activity Monitor, Task Instances list, History list, or Command Line Interface (CLI).

<b>Activity Monitor</b>	<p>To display the status of one or more task instances on the Activity Monitor:</p> <ul style="list-style-type: none"> <li>• From the <a href="#">Automation Center</a> navigation pane, select <b>Task Instances &gt; Activity</b>. The Activity Monitor contains a <b>Status</b> column that identifies the current status of every task instance on the list.</li> </ul>
<b>Task Instances list</b>	<p>To display the status of one or more task instances on the Task Instances list:</p> <ul style="list-style-type: none"> <li>• From the <a href="#">Automation Center</a> navigation pane, select <b>Task Instances &gt; All Task Instances</b>. The Task Instances list contains a <b>Status</b> column that identifies the current status of every task instance on the list.</li> </ul>
<b>History list</b>	<p>To display the status of one or more task instances on the History list:</p> <ul style="list-style-type: none"> <li>• From the <a href="#">Automation Center</a> navigation pane, select <b>Task Instances &gt; History</b>. The History list contains a <b>Status</b> column that identifies the current status of every task instance on the list.</li> </ul>
<b>Command Line Interface (CLI)</b>	<p>To display the status of one or more task instances from the Command Line Interface:</p> <ul style="list-style-type: none"> <li>• Use the <a href="#">ops-task-status</a> function.</li> </ul>

## Task Instance Status Types

The following table describes all possible task instance statuses for all task types.

For a list of commands that you can issue against a task instance in each status, see [Commands Supported for Task Instance Statuses](#).

For a description of each command, see [Issuing Commands Against Task Instances](#).

### Note



The format of multi-word task status names used in the Controller user interface differs from the format of task status names specified in the [Display Task Instance Status](#) CLI function. In the user interface, they are separated by a space; in the [Display Task Instance Status](#) CLI function, the words are separated by an underscore character.

Status Code	Status Name	Task Type	Description
0	Defined	All	The new task instance has been created (the task has been launched).
10	Waiting	All	The task instance has been loaded by a workflow and is waiting on a predecessor.
15	Time Wait	All (except Timer)	The task instance is waiting to start based on a Wait To Start and/or Delay On Start specification.
20	Held	All	The task instance has been put on hold by a user.
22	Exclusive Requested	All	All task instances with a <a href="#">mutually exclusive task</a> defined go immediately to a status of Exclusive Requested. If the task instance is available to run exclusively, the task instance then moves to the next appropriate processing status.
23	Exclusive Wait	All	The task instance is <a href="#">mutually exclusive</a> with one or more other task instances, and it is waiting for those task instances to finish before it will run.
25	Resource Requested	All	All task instances with a <a href="#">virtual resource</a> defined go immediately to a status of Resource Requested. If the resource is available, the task instance then moves to the next appropriate processing status.
30	Resource Wait	All	All task instances with a <a href="#">virtual resource</a> defined go immediately to a status of Resource Requested. If the resource is not available, the task instance goes to a status of Resource Wait. When the resource becomes available, the task instance moves to the next appropriate processing status.
33	Execution Wait	Agent-based	The task instance must wait to be completed; either the Agent/Agent Cluster running the task instance has reached its Task Execution Limit, or the ability of the Agent /Agent Cluster to run tasks has been <a href="#">suspended</a> .
35	Undeliverable	Agent-based	The Agent is unavailable.
40	Queued	Agent-based	The task instance has been queued on a resource.
43	Submitted	z/OS	The task instance has been submitted to the z/OS Job Entry subsystem and scheduled by the z/OS Job Scheduler.
45	Step Restarted	z/OS	The task instance has been re-run starting from a specific <a href="#">z/OS jobstep</a> .
60	Action Required	Manual	When a Manual task launches, the task instance goes into Action Required status, meaning a user must perform some manual activity. For details, see <a href="#">Manual task</a> .
70	Started	Agent-based, Manual	The task instance has started. For Agent-based tasks, this means the Agent has received the task.
80	Running	All	The task instance is running. For Agent-based tasks, the Agent has started running the program.
81	Running /Problems	Workflow	One or more task instances within the workflow has one of the following statuses: <ul style="list-style-type: none"> <li>• Cancelled</li> <li>• Confirmation Required</li> <li>• Failure</li> <li>• In Doubt</li> <li>• Running/Problems (for sub-workflows)</li> <li>• Start Failure</li> <li>• Undeliverable</li> </ul>
99	Cancel Pending	Agent-based	A process running on the Agent needs to be terminated. When the Cancel command is issued, the task instance will go into a Cancel Pending status until the Agent reports back that the process has been cancelled. At that point, the task instance will transition into the Cancelled status.
110	In Doubt	Agent-based, SQL, Stored Procedure, Web Service	The task instance is "in doubt" about the current status of the job. This may occur if an Agent or Agent connection goes down. In this case, the Agent restarts and reviews its data about task instances in progress. If the Agent finds a task instance still running, it resumes normal monitoring. If the Agent cannot find the task instance, this usually indicates that the task instance completed, but the Agent considers the task instance status to be "in doubt."  When the Controller restarts, any <a href="#">SQL</a> , <a href="#">Stored Procedure</a> , and <a href="#">Web Service</a> task instances that are in a Running status will transition into the In Doubt status, since the Controller no longer has any knowledge of them.
120	Start Failure	All	The task instance was unable to start.
125	Confirmation Required	z/OS	If you make JCL changes and restart a z/OS task instance, Universal Controller will put the task instance into Confirmation Required status and prompt you for a confirmation. For detailed processing steps, see <a href="#">Rerunning a z/OS Task</a> .
130	Cancelled	All	The task instance was cancelled by a user.

140	Failed	All (except Workflow)	The task instance ran to a failure status.
180	Skipped	All	The task instance was skipped by Run/Skip criteria, an Execution Restriction, or the Skip command.
190	Finished	All	The task instance was forced by the user to finish. The user may do this in cases where the task instance had a 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 <a href="#">Force Finishing a Task</a> .
200	Success	All	The task instance has completed successfully. Workflows will transition to Success status when all of its task instances have transitioned to Success, Finished, or Skipped status.

## Agent-Based Task Types

The following task types are Agent-based task types:

- Linux/Unix
- Windows
- z/OS
- Universal Command
- SAP
- PeopleSoft
- File Transfer
- Agent File Monitor
- Remote File Monitor
- System Monitor
- Universal

## Task Instance Status Colors

You can change the default color assigned to each task instance status via the [Colors](#), in the **Reporting** navigation pane.

# Retrieving Output

- [Overview](#)
  - [Task Instance Output](#)
- [Retrieving Output Automatically](#)
- [Retrieving Output Manually](#)
- [Retrieve Output Field Descriptions](#)

## Overview

For some [Agent-based](#) task instances where output has been generated (see [Task Instance Output](#), below), you can choose to have the output retrieved [automatically](#) or [manually](#).

## Task Instance Output

The following table identifies the types of [Agent-based tasks](#) whose task instances generate output, and the type of output they generate:

Task Type	Standard Output	Standard Error	z/OS Job Log
Linux/UNIX	✓	✓	
Windows	✓	✓	
z/OS			✓
Universal Command	✓	✓	
SAP	✓	✓	
PeopleSoft	✓	✓	
File Transfer	✓	✓	
Remote File Monitor	✓	✓	
Universal	✓	✓	

Note



For File Transfer tasks, you cannot choose to select automatic or manual output retrieval. The Controller always retrieves its output automatically.

Similarly, although Web Service tasks are not Agent-based tasks, Web Service task instances always produce output, which the Controller always retrieves automatically.

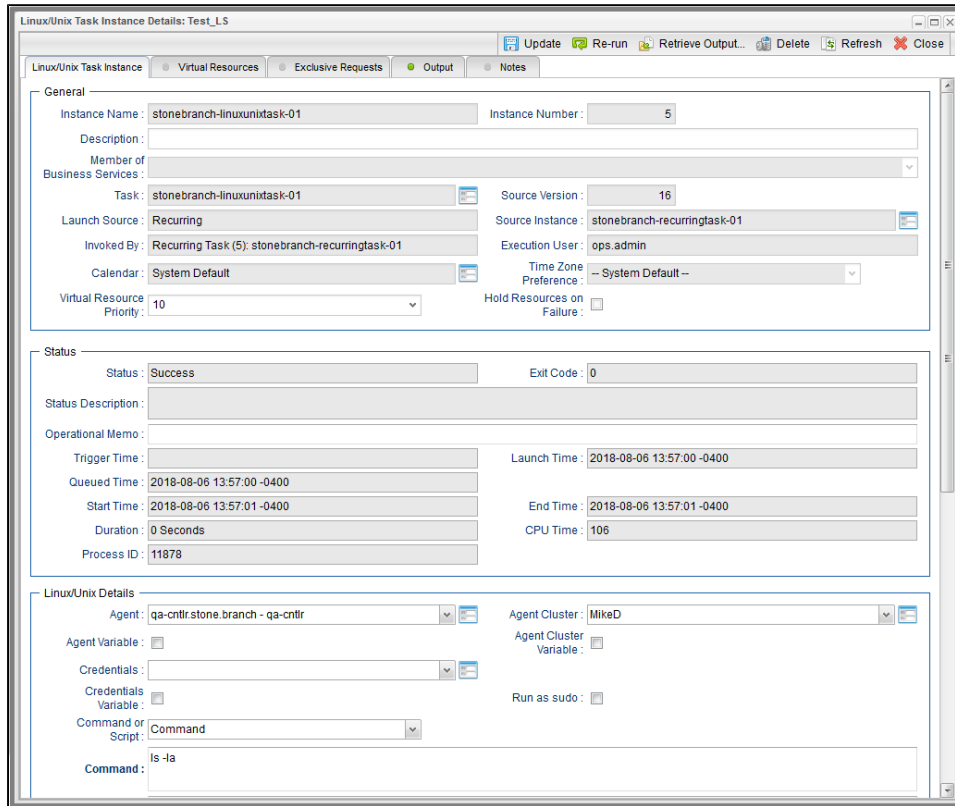
## Retrieving Output Automatically

You can specify that Universal Controller automatically retrieves output from an Agent-based task instance after it has completed running.

<b>Step 1</b>	Specify Automatic Output Retrieval for a task (in its <a href="#">Automatic Output Retrieval</a> field).
<b>Step 2</b>	Run the task; the Controller automatically retrieves that type of output for the task instance.

**Step 3** From the [Automation Center](#) navigation pane, select **Task Instances > Activity** or **Task Instances > Task Instances** to display the [Activity Monitor](#) or [Task Instances list](#).

**Step 4** Open the task instance. The Output tab displays a green icon, indicating that output has been retrieved automatically.



**Step 5** Click the Output tab to view the output that you specified to be automatically retrieved.

Linux/Unix Task Instance Details: Test\_LS

Linux/Unix Task Instance | Virtual Resources | Exclusive Requests | **Output** | Notes

2 Output Retrieve Output...

Type	Attempt	Output	Updated By	Updated
		total 624		
		dr-xr-xr-x. 18 root root 4096 Aug 1 09:22 .		
		dr-xr-xr-x. 18 root root 4096 Aug 1 09:22 ..		
		lrwxrwxrwx. 1 root root 7 Aug 10 2015 bin -> usr/bin		
		dr-xr-xr-x. 4 root root 4096 Aug 10 2015 boot		
		drwxr-xr-x. 21 root root 3160 Mar 28 16:35 dev		
		-rw-rw-rw-. 1 root root 138570 Aug 7 06:39 dev_rfc.trc		
		-rw-r--r--. 1 root root 60 Aug 1 09:22 .err		
		drwxr-xr-x. 82 root root 8192 Jul 12 13:42 etc		
		drwxr-xr-x. 7 root root 74 Jul 12 13:42 home		
		-rw-r--r--. 1 root root 333212 Aug 1 09:22 install.log		
		lrwxrwxrwx. 1 root root 7 Aug 10 2015 lib -> usr/lib		
		lrwxrwxrwx. 1 root root 9 Aug 10 2015 lib64 -> usr/lib64		
STDOUT	1	drwxr-xr-x. 2 root root 6 Jun 9 2014 media	ops.system	2018-08-07 10:05:46 -0400
		drwxr-xr-x. 8 root root 18 Aug 10 2015 mnt		
		drwxr-xr-x. 12 root root 4096 Aug 2 12:16 opt		
		dr-xr-xr-x. 138 root root 0 Mar 28 16:35 proc		
		drwxrwxrwx. 24 qatest root 4096 Aug 7 06:30 qa		
		-rw-rw-rw-. 1 root root 100107 Aug 10 2015 r_cvs.test.pl		
		dr-xr-xr--. 5 root root 4096 Aug 6 11:44 root		
		drwxr-xr-x. 23 root root 680 Jun 12 13:12 run		
		lrwxrwxrwx. 1 root root 8 Aug 10 2015/sbin -> usr/sbin		
		drwxr-xr-x. 2 root root 6 Jun 9 2014 srv		
		dr-xr-xr-x. 13 root root 0 Mar 28 16:35 sys		
		drwxrwxrwt. 11 root root 4096 Aug 7 03:49 tmp		
		drwxr-xr-x. 13 root root 4096 Aug 10 2015 usr		
		drwxr-xr-x. 22 root root 4096 Aug 2 11:40 var		
STDERR	1	[empty]	ops.system	2018-08-07 10:05:46 -0400

**Step 6** You can click the **Retrieve Output...** button to display the Retrieve Output dialog, which lets you select different, or additional, output to retrieve.

If the selected task is a z/OS task, the Retrieve Output dialog contains a single **Output Type** selection:

**Step 7** Using the [field descriptions](#) below as a guide, make your selection. Positioning can be accomplished by using [Start Line](#) or [Scan Text](#) fields to see the [Number of Lines](#) both before and after the first match of text.

Note

 If you want to retrieve data starting at the end of a file, enter **-1** in the [Start Line](#) field and the number of lines to retrieve in the [Number of Lines](#) field.

## Retrieving Output Manually

You can choose to retrieve output manually from a task instance while it is running or after it has completed running.

**Step 1** From the [Automation Center](#) navigation pane, select **Task Instances > Activity** or **Task Instances > Task Instances** to display the [Activity Monitor](#) or [Task Instances list](#).

**Step 2** Open the task instance from which you want to retrieve output. The Output tab displays a gray icon, indicating that output has not been retrieved automatically.

The screenshot displays the 'Linux/Unix Task Instance Details: Test\_LS' window. At the top, there are tabs for 'Linux/Unix Task Instance', 'Virtual Resources', 'Exclusive Requests', 'Output', and 'Notes'. The 'Output' tab is selected, and a gray icon is visible next to it, indicating that the output has not been retrieved. The window is divided into three main sections: 'General', 'Status', and 'Linux/Unix Details'.  
**General Section:**  
 Instance Name: stonebranch-linuxunitask-01  
 Instance Number: 5  
 Description: (empty)  
 Member of Business Services: (dropdown menu)  
 Task: stonebranch-linuxunitask-01  
 Source Version: 16  
 Launch Source: Recurring  
 Source Instance: stonebranch-recurringtask-01  
 Invoked By: Recurring Task (5): stonebranch-recurringtask-01  
 Execution User: ops.admin  
 Calendar: System Default  
 Time Zone Preference: -- System Default --  
 Virtual Resource Priority: 10  
 Hold Resources on Failure: (checkbox)  
**Status Section:**  
 Status: Success  
 Exit Code: 0  
 Status Description: (empty)  
 Operational Memo: (empty)  
 Trigger Time: (empty)  
 Launch Time: 2018-08-06 13:57:00 -0400  
 Queued Time: 2018-08-06 13:57:00 -0400  
 Start Time: 2018-08-06 13:57:01 -0400  
 End Time: 2018-08-06 13:57:01 -0400  
 Duration: 0 Seconds  
 CPU Time: 106  
 Process ID: 11878  
**Linux/Unix Details Section:**  
 Agent: qa-ctrl.stone.branch - qa-ctrl  
 Agent Cluster: MikeD  
 Agent Variable: (checkbox)  
 Agent Cluster Variable: (checkbox)  
 Credentials: (dropdown menu)  
 Credentials Variable: (checkbox)  
 Run as sudo: (checkbox)  
 Command or Script: Command  
 Command: ls -la

**Step 3** Click the Retrieve Output button to display the Retrieve Output dialog.

If the selected task is a z/OS task, the Retrieve Output dialog contains a single **Output Type** selection:

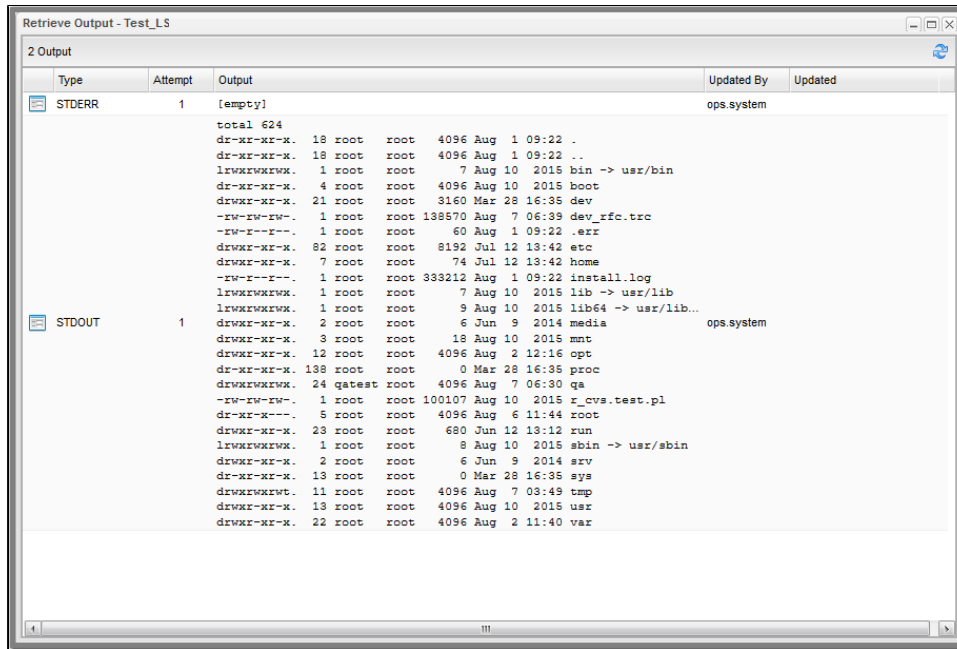
**Step 4** Using the [field descriptions](#) below as a guide, make your selection. Positioning can be accomplished by using [Start Line](#) or [Scan Text](#) fields to see the [Number of Lines](#) both before and after the first match of text.

Note

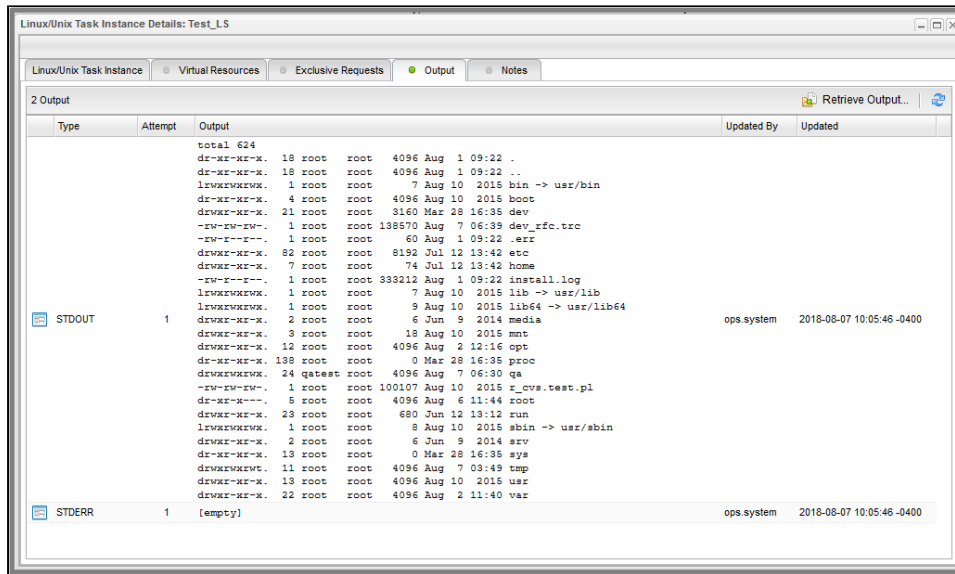
 If you want to retrieve data starting at the end of a file, enter **-1** in the [Start Line](#) field and the number of lines to retrieve in the [Number of Lines](#) field.

**Step 5** Click the **Submit** button. The Controller retrieves the output you specified and displays a Retrieve Output list containing the requested type of output.

(It also writes the output to the **Output** tab of the task instance.)



**Step 6** If you now click the Output tab with the gray icon on the task instance Details, the icon becomes green, and the retrieved output displays.



**Step 7** You can again click the **Retrieve Output...** button to display the Retrieve Output dialog, which lets you select different, or additional, output to retrieve.

## Retrieve Output Field Descriptions

Field Name	Description
Standard Output and Standard Error	Retrieve both standard output and standard error information returned by the program.
Standard Output	Retrieve standard output only.
Standard Error	Retrieve standard error output only.
z/OS Job Log	z/OS tasks only; Retrieve information from the z/OS Job Log.
Start Line	Retrieve data beginning at the line indicated. <ul style="list-style-type: none"> <li>• If a <b>Start Line</b> value is not specified in the task instance Details, the default is 1.</li> <li>• If the <b>Start Line</b> value is -1, data will be retrieved starting at the end of the file.</li> </ul>
Number of Lines	Limit the retrieved data to the <b>Number of Lines</b> value in the task instance Details. If a <b>Number of Lines</b> value is not specified, default is the value of the <a href="#">Retrieve Output Default Number Of Lines</a> Universal Controller system property.

Scan Text:	Regex pattern that the Controller will search for a match for in STDOUT/STDERR (or z/OS Job Log). The Controller will include the Number of Lines above and below the first line matched.  if the Regex pattern is not found, the following message is returned: OPSWISE WARNING - Scan text string not found.
------------	--