



Universal Controller 6.3.x

Application Monitoring and Control

© 2016 by Stonebranch, Inc. All Rights Reserved.

1. Application Monitoring and Control	3
1.1 Application Monitoring and Control Overview	4
1.2 Application Resources	6
1.3 Application Control Tasks	11
1.4 Application Monitor Triggers	27

Application Monitoring and Control



Application Monitoring and Control

Overview

Application Monitoring Processing Flow

Application Resources

Application Control Tasks

Application Monitor Triggers



The information on these pages also is located in the [Universal Controller 6.3.x Application Monitoring and Control.pdf](#).

Application Monitoring and Control Overview


- [Application Monitoring and Control](#)
- [Processing Flow](#)

Application Monitoring and Control

The Application Monitoring and Control feature of Universal Controller allows you to use it as a network control and monitoring tool. You can use Application Monitoring and Control to start, stop, and query any application running on any machine where you have [Universal Agent](#) installed and running.

Application Monitoring and Control is comprised of three components:

- [Application resource records](#) allow you to define the name and location of your applications, along with the specific commands to control (Start, Stop, and Query) the applications. The [Applications list](#) displays a status for the application that is defined in each Application resource record.
- Three [Application Control tasks](#) are automatically generated when you create an Application resource record: one each for executing the Start, Stop, and Query commands (which you specified in the Application resource record) against the application. You can use these control tasks to schedule the Start, Stop, and Query commands in Workflows and triggers. You also can manually create customized Application Control tasks.

 **Note**
 You can manually run an Application Control task to execute a command specified in an Application resource record, but it is simpler to just execute the command from the Applications list or Application resource record.

- Optional [Application Monitor triggers](#) allow you to launch other tasks based on the status of an application being monitored.

Processing Flow

The following steps show a sample process flow for the manual monitoring (that is, not via a trigger or Workflow) of an application:

Step 1	From the Agents & Connections navigation pane, select System > Applications and create an Application resource record , specifying the name of an application and the start, stop, and query commands to control it. The Controller will automatically create three Application Control tasks that you can use in Workflows and triggers for starting, stopping, and querying the application.
Step 2	Start the application defined in the Application resource record either by: <ul style="list-style-type: none"> • Right-clicking the Application resource record in the Applications list and clicking Start on the displayed Action menu. • Opening the Application resource record and clicking the Start button in the Application Details.
Step 3	The Controller executes the Start Command provided by the user in the Application Details . It puts the application into Starting status, and saves the Start Time . The Start Command has two functions: <ol style="list-style-type: none"> 1. Starts the application. 2. Starts the query process that monitors the application.
Step 4	After 30 seconds, the Controller automatically executes the Query Command provided by the user in the Application Details to determine the status of the application. The Controller continues executing the Query Command every 120 seconds thereafter until the user stops the monitoring by issuing a Stop command from the Controller.

<p>Step 5</p>	<p>The purpose of the Query is to determine whether or not the application is Active. The Controller uses the specifications provided by the user in the Query Exit Code Processing fields in Application Details to make this determination.</p> <ul style="list-style-type: none"> • If the response from the application indicates a successful start-up, the Controller puts the application into Active status. • If the response indicates the Application has not started, the Controller continues executing the Query (keeping track in the Startup Query Attempts field) until it reaches the maximum attempts specified by the user in the Startup Query Maximum field. If the maximum number is reached before achieving an Active status, the Controller puts the application into Impaired status. However, the Controller continues monitoring the application. If the appropriate exit code parameters are eventually returned, the Controller will put the application into Active status. The purpose of the Startup Query Attempts field is to avoid having the application go straight into Impaired status if it takes awhile to start. <p>The Controller writes any Exit Code captured by the Query in the Query Exit Code field of the Application resource record.</p>
<p>Step 6</p>	<p>After starting the application, the Controller continues monitoring by sending out the Query Commands every 120 seconds.</p> <ul style="list-style-type: none"> • If the Controller detects a problem based on the Exit Code parameters, it puts the application into Impaired status. If this occurs, you have several options for handling the problem, with increasing levels of automation: <ol style="list-style-type: none"> 1. The Applications list displays the status of all applications being monitored. You can create a filter for the Applications list that displays only those applications in a specific status, such as Impaired. If you see a problem, troubleshoot the issue and restart the application from outside the Controller. 2. Set up an Application Monitor trigger that monitors the application for Impaired and other problem statuses. When the trigger is satisfied, it launches an Email task that sends emails to support personnel, notifying them of the problem. Several built-in variables are supported that allow you to pass required data into the email message: the application name, type, and status. 3. You also could create a Workflow launched by an Application Monitor trigger looking for Impaired or other problem statuses. The Workflow can include Application Control tasks that attempt to resolve the problem by stopping and then restarting the application. You could also include any other tasks that are specific to troubleshooting the application. • If the Controller fails to get a response to a Query for three minutes, it puts the application into <status>/Query Overdue status, where <status> is the last known status of the application, either Starting, Active, or Impaired. For example, you may see a <status>/Query Overdue status if the Agent went down or there was some other problem on the machine unrelated to the application itself. If this occurs, you should troubleshoot the issue. When you have fixed the problem, the continued queries from the Controller will then return an Active status for the application.
<p>Step 7</p>	<p>To stop monitoring an application, issue the Stop command against it. the Controller stops the application and puts it into Inactive status, which means it is no longer monitoring.</p>

Application Resources

- [Overview](#)
- [Built-In Variables](#)
- [Creating an Application Resource Record](#)
 - [Application Details](#)
 - [Application Details Field Descriptions](#)

Overview

Application resource records are the core component of the Universal Controller [Application Monitoring and Control](#) feature.

These records define:

- Names of the applications being monitored.
- Name and location of the machines where they are running.
- Start, Stop, and Query commands needed to perform the monitoring and control functions.

You can also use Application records and their associated [Application Control tasks](#) to start, stop, and query applications as part of your scheduling processes. You can execute Application Control tasks as you would execute any other task and include them in Workflows where applicable. In addition, you can define [Application Monitor triggers](#) to automatically launch one or more tasks of any type, depending on the status of one or more applications. For example, you might set up an Application Monitor trigger that sends an email to Windows technical support personnel if any Windows application goes to Impaired or Inactive status.

In order for the Controller to access the application, the application must be installed on a machine where [Universal Agent](#) (for Windows, Linux/Unix, or z/OS) is running.

If you set up the Controller to monitor your applications, you should always start and stop the applications from within the Controller. If you stop an application outside the Controller, you must also restart it from outside the Controller. If the Controller detects a problem with an application (the application goes to Impaired status), you should troubleshoot the problem and restart the application outside the Controller. The Controller will continue monitoring and when it detects that the application is back up, it will put the application back into Active status.

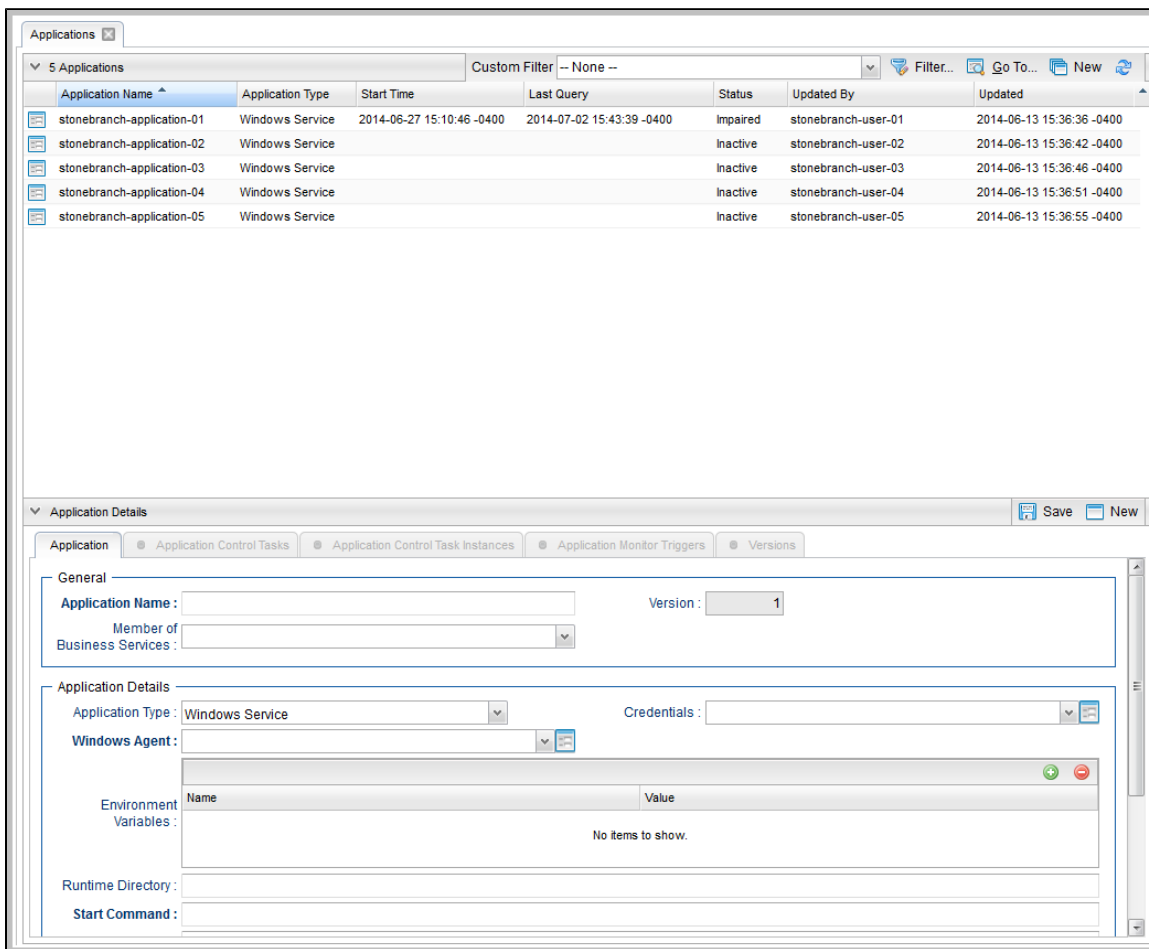
Built-In Variables

[Application Monitor built-in variables](#) are provided to pass information about an application being monitored into the task or tasks being launched by the trigger. You can pass the information into the launched tasks by including the variables in a text field in the task definition.

Creating an Application Resource Record

Step 1 From the [Agents & Connections](#) navigation pane, select **System > Applications**. The Applications list displays a list of all currently defined Application resource records.


Below the list, Application Details for a new Application resource record displays. (You also can click the **New** button to display Application Details for a new Application resource record.)



Step 2 Enter/select Details for a new Application resource record, using the [field descriptions](#) below as a guide.

- Required fields display in **boldface**.
- Default values for fields, if available, display automatically.

To display more of the Details fields on the screen, you can temporarily [hide the list](#).

 **Note**
If you view [Application Details](#) for an existing Application resource record by clicking an Application in the list, and then want to create a new Application resource record, you must click the **New** button that displays above and below the Details.

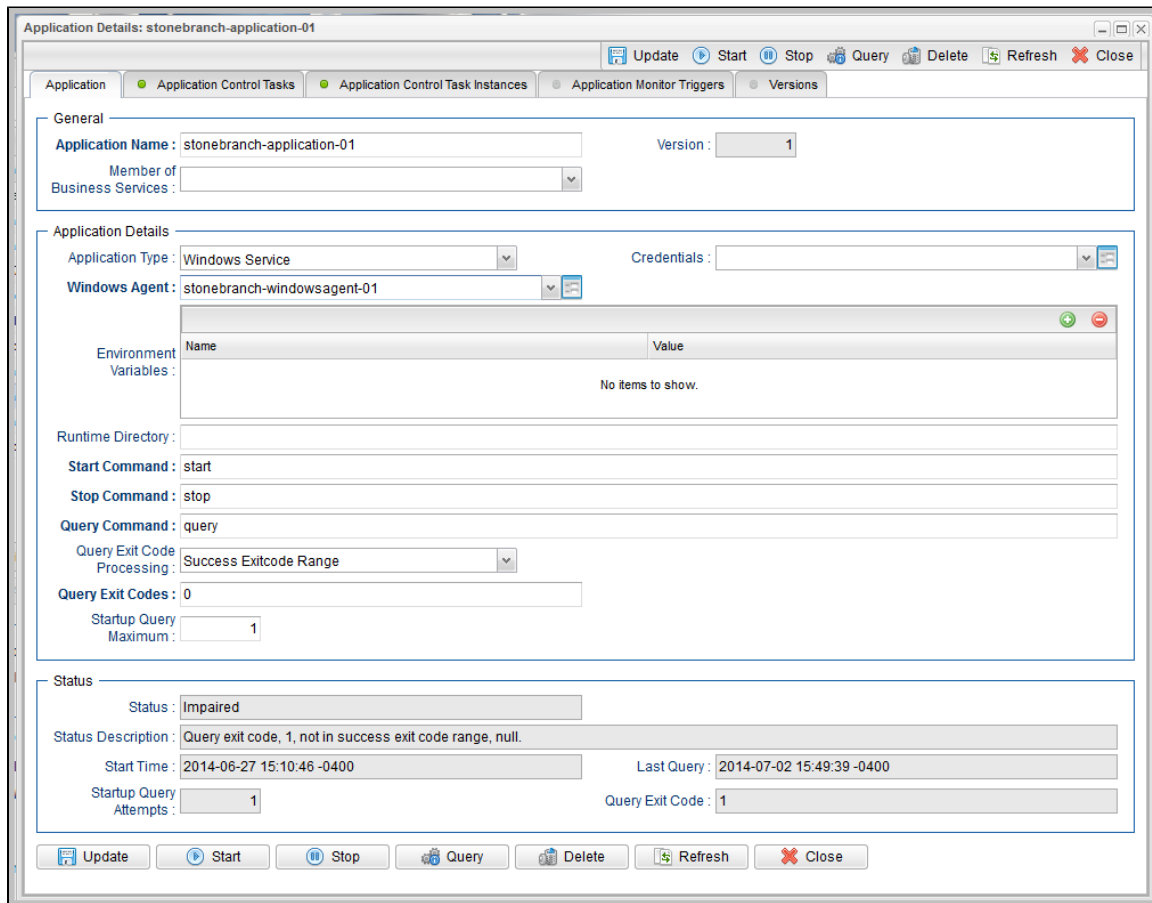
Step 3 Click the **Save** button. The task is added to the database, and all buttons and tabs in the Task Details are enabled.

When you save a new Application resource record, the Controller also automatically creates three related [Application Control Tasks](#), one each for starting, stopping, and querying the application.

Step 4 If appropriate, repeat these steps for any additional Application resource records that you want to add.

Application Details


The following Application Details is for an existing Application resource record. See the [field descriptions](#), below, for a description of all fields that may display in the Application Details.



Application Details Field Descriptions

The following table describes the fields, buttons, and tabs that display in the Application Details.

Field Name	Description
General	This section contains general information about the application.
Application Name	Name used within the Controller to identify this resource. Up to 40 alphanumeric characters. It is the responsibility of the user to develop a workable naming scheme for resources.
Version	Version number of the current record, which is incremented by the Controller every time a user updates a record. Click on the Versions tab to view previous versions. For details, see Record Versioning .
Member of Business Services	User-defined; allows you to select one or more Business Services that this Application resource belongs to. Click on the lock icon to unlock the field and select a Business Service .
Application Details	This section contains assorted detailed information about the application.
Application Type	User-defined; Type of application. Options: <ul style="list-style-type: none"> • Windows Service • Linux/Unix Daemon • z/OS Started Task

Credentials	<p>Login credentials that the Controller will use to access the remote machine.</p> <div style="background-color: #ffffcc; padding: 5px; margin-top: 10px;">  Note For z/OS application resources, make sure the credentials are in upper case. </div>
Windows Agent	If Application Type = Windows Service; Name of the Windows agent where the application will run.
Linux/Unix Agent	If Application Type = Linux/Unix Daemon; Name of the Linux/Unix agent where the application will run.
z/OS Agent	If Application Type = z/OS Started Task; Name of the z/OS agent where the application will run.
Environment Variables	<p>If Application Type = Windows Service or Linux/Unix Daemon; Allows you to enter environment variables needed by the program to run.</p> <p>To add a variable, click the + icon and enter a Name and Value. To delete a variable, select in the list of variables and click the - icon.</p> <p>You can add a maximum of 4,000 characters for the combined Names and Values of all variables. The variable is listed in the space underneath.</p>
Run as sudo	If Application Type = Linux/Unix Daemon; Instruction to run the command as Sudo (superuser do).
Runtime Directory	Directory where the application executes. Variables supported.
Start Command	Command used to start the application. This can be any process or command that starts the application. If you try to start an application monitor that is already started, you will see the message: Application already monitored with <status> status.
Stop Command	Command used to stop the application. This can be any process or command that stops the application.
Query Command	Command used to query the application. This can be any process or command that queries the application. You must first start the application monitor from the Controller before you can query the application.
Query Exit Code Processing	<p>Specifies how the Controller should determine whether or not the application is running.</p> <p>Options:</p> <ul style="list-style-type: none"> • Success Exitcode Range - Application goes to or remains in Active status if its exit code falls within the range specified in the Query Exit Codes field (see below). Otherwise it has Impaired status. • Failure Exitcode Range - Application goes to or remains in Impaired status if its exit code falls within the range specified in the Exit Codes field (see below). Otherwise it has Active status. • Success Output Contains - Application goes to or remains in Active status if its output contains the text specified in the Scan Output For field (see below). Otherwise it has Impaired status. • Failure Output Contains - Application goes to or remains in Impaired status if its output contains the text specified in the Scan Output For field (see below). Otherwise it has Active status.
Query Exit Code	System-supplied if Query Exit Code Processing = Success Exitcode Range or Failure Exitcode Range ; the most recent exit code returned by the application in response to a query.
Output Type	<p>If Query Exit Code Processing = Success Output Contains or Failure Output Contains; type of output.</p> <p>Options:</p> <ul style="list-style-type: none"> • Standard Output (STDOUT) • Standard Error (STDERR) • File
Scan Output For	If Query Exit Code Processing = Success Output Contains or Failure Output Contains ; string that the Controller should scan for in the output.
Output File	If Output Type = File ; path and name of the file.
Startup Query Maximum	Maximum number of Query attempts to be made on the specified application.
Status	This section contains information about the current status of the task application.

Status	<p>System-supplied; indicates the current status of the application.</p> <p>Options:</p> <ul style="list-style-type: none"> • Inactive - Application is not being monitored by the Controller. • Start Failure - Application failed to start. This may occur, for example, if you have problems with credentials or the start command itself is incorrect. When this occurs, the Controller is not monitoring the application. You should troubleshoot the problem and restart the application from the Controller. • Starting - Start command has been executed. • Active - Application has successfully started and is running, based on the parameters specified in the Exit Code processing fields. • Impaired - An application that is being monitored returned a response that, based on the specified exit code parameters, indicates it is not running. If this occurs, you should troubleshoot the problem and restart the application from outside the Controller. Unless you issue a stop command, the Controller continues monitoring during this process. When the application comes back up, the query process will recognize this and return the application to Active status.
Status Description	System-supplied; a more detailed status message describing why a status change occurred, in the format: "Query exit code <in-not in> <success-failure> exit code range. Query <success-failure> output not found."
Start Time	System-supplied; Date and time that the application was last started by the Controller.
Last Query	System-supplied; date and time of the last query response received from the application.
Startup Query Attempts	System-supplied; Number of queries that were executed before the Application went into Active or Impaired status.
Query Exit Code	<p>Required if Query Exit Code Processing = Success Exitcode Range or Failure Exitcode Range; range of exit codes. Format is numeric.</p> <p>Use commas to list a series of discontinuous exit codes; use hyphens to specify a series of continuous exit codes. For example: 1,5,11, 22-30.</p>
Buttons	This section identifies the buttons displayed above and below the Application Details that let you perform various actions.
Save	Saves a new record in the Controller database.
Update	Saves updates to the record.
New	Displays empty (except for default values) Details for creating a new application.
Start	Executes the Start command associated with this Application resource and begins querying.
Stop	Executes the Stop command associated with this Application resource. the Controller stops the application and stops querying (monitoring).
Query	Executes the Query command associated with this Application resource. This allows you to get immediate status of the application instead of waiting for the next automated query.
Delete	Deletes the current record.
Refresh	Refreshes any dynamic data displayed in the Details.
Close	For pop-up view only; closes the pop-up view of the Details.
Tabs	This section identifies the tabs across the top of the Application Details that provide access to additional information about this Application resource.
Application Control Tasks	Lists all Application Control tasks associated with this Application resource.
Application Control Task Instances	Lists all Application Control task instances associated with this Application resource.
Application Monitor Triggers	Lists all Application Monitor triggers associated with this Application resource.

Application Control Tasks

- [Overview](#)
- [Built-In Variables](#)
- [Creating an Application Control Task](#)
 - [Application Control Task Details](#)
 - [Application Control Task Details Field Descriptions](#)
- [Viewing an Application Control Task Instance](#)
 - [Application Control Task Instance Details](#)
 - [Application Control Task Instance Details Field Descriptions](#)
- [Running an Application Control Task](#)
- [Monitoring Task Execution](#)

Overview

Application Control tasks allow you to execute a Start, Stop, or Query command against an application in the Universal Controller network.

Three Application Control tasks are created automatically when you create an [Application record](#) – one each for starting, stopping, and querying the application. (The Application Control tasks list [below](#) shows automatically created tasks for five different Application records.)

Each of these automatically created tasks is stored as a separate record in the Controller database and can be executed independently or added to a workflow, as with any other task. These tasks cannot be deleted.

Built-In Variables

The following [built-in variables](#) can be used in an Application Control task to pass data where appropriate:

- [Task Instance variables](#)

Creating an Application Control Task


Step 1 From the [Automation Center](#) navigation pane, select **Tasks > Application Control Tasks**. The Application Control Tasks list displays a list of all currently defined Application Control tasks.

Below the list, Application Control Task Details for a new Application Control task displays. (You also can click the **New** button to display Application Control Task Details for a new Application Control task.)

Step 2 Enter/select Details for a new Application Control task, using the [field descriptions](#) below as a guide.

- Required fields display in **boldface**.
- Default values for fields, if available, display automatically.

To display more of the Details fields on the screen, you can temporarily [hide the list](#).

 **Note**
If you view [Application Control Task Details](#) for an existing Application Control task by clicking a task in the list, and then want to create a new Application Control task, you must click the **New** button that displays above and below the Details.

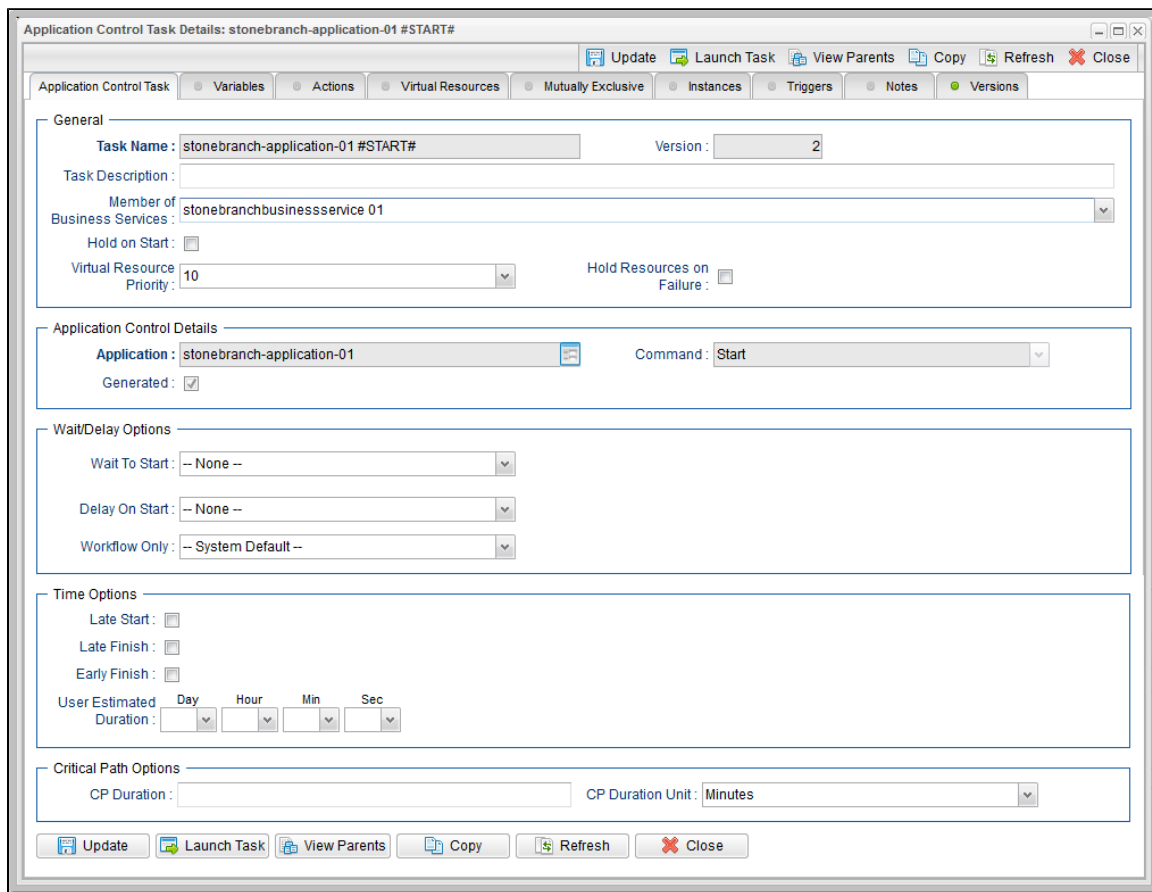
Step 3 Click the **Save** button. The task is added to the database, and all buttons and tabs in the Task Details are enabled.

Application Control Task Details

The following Application Control Task Details is for an automatically generated Application Control task.

Some of the fields are protected and the **Generated** field is pre-selected, indicating that this task was generated automatically.

Depending on the values that you enter / select for these fields, and whether or not the Application Control task has ever been launched, more (or less) fields may display. See the [field descriptions](#), below, for a description of all fields that may display in the Application Control Task Details.



Application Control Task Details Field Descriptions

The following table describes the fields, buttons, and tabs that display in the Application Control Task Details.

Field Name	Description
General	This section contains general information about the task.
Task Name	User-defined name of this task, containing a maximum 40 alphanumeric characters; variables supported. It is the responsibility of the user to develop a workable naming scheme for tasks.
Version	System-supplied; version number of the current record, which is incremented by the Controller every time a user updates a record. Click the Versions tab to view previous versions. For details, see Record Versioning .
Task Description	User-defined; description of this record. (Maximum = 200 characters.)
Member of Business Services	User-defined; allows you to select one or more Business Services that this record belongs to.
Hold on Start	If enabled, when the task is launched it appears in the Activity Monitor with a status of Held . The task runs when the user releases it.
Hold Reason	Information about why the task will be put on hold when it starts.

Virtual Resource Priority	<p>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.</p> <p>Options: 1 (high) - 20 (low).</p> <p>Default is 10.</p>
Hold Resources on Failure	<p>If enabled, the task instance will continue to hold Renewable resources if the task instance fails. Renewable resources will be returned only if the task instance status is either Complete, Finished, or Skipped.</p>
Application Control Details	<p>This section contains assorted detailed information about the task.</p>
Application	<p>Protected if auto-generated; name of the Application resource record. The Application resource defines where the software application is running; it also defines the start, stop, and query commands for the application. Enter the name of an existing Application, select an Application from the drop-down list, of all existing Applications, or click the Details icon to create a new Application.</p>
Command	<p>Protected if auto-generated; command that this task is executing against the software application.</p> <p>Options:</p> <ul style="list-style-type: none"> • Query • Start • Stop
Generated	<p>System-supplied; protected. If selected, indicates that this Application Control task was generated automatically when the Application resource record was submitted.</p>
Wait / Delay Options	<p>This section contains specifications for waiting to start and/or delaying on start the task.</p>
Wait To Start	<p>Amount of time to wait before starting a task from the time that it was launched.</p> <p>Options are:</p> <ul style="list-style-type: none"> • – None – • Time • Relative Time • Duration • Seconds
Wait Time	<p>If Wait To Start = Time or Relative Time; Number of hours and minutes to wait before starting the task.</p>

<p>Wait Day Constraint</p>	<p>If Wait Time = Time or Relative Time; Specification for whether or not to advance the wait time to another day.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • -- None -- <ul style="list-style-type: none"> • If Wait To Start = Time; Advance to the next day if calculated wait time is before the current time. • If Wait To Start = Relative Time; Advance to the next day if calculated wait time is before the Trigger/Launch Time. • Same Day Do not advance day. • Next Day Advance to the next day. • Next Business Day Advance to the next business day. • Sunday If today is not Sunday, advance to next Sunday. • Monday If today is not Monday, advance to next Monday. • Tuesday If today is not Tuesday, advance to next Tuesday. • Wednesday If today is not Wednesday, advance to next Wednesday. • Thursday If today is not Thursday, advance to next Thursday. • Friday If today is not Friday, advance to next Friday. • Saturday If today is not Saturday, advance to next Saturday. <p>Default is – None --.</p>
<p>Wait Duration</p>	<p>If Wait To Start = Duration; Number of days, hours, minutes, and seconds to wait before starting the task.</p>
<p>Wait Duration In Seconds</p>	<p>If Wait To Start = Seconds; Number of seconds to wait before starting the task.</p>
<p>Delay On Start</p>	<p>Amount of time to delay the start of a task, after it has been launched, from the time that it is eligible to start; that is, all dependencies have been met. For example: it is not being held, it is not waiting on any predecessors, or there is no wait time specified.</p> <p>Options are:</p> <ul style="list-style-type: none"> • – None – • Duration • Seconds
<p>Delay Duration</p>	<p>If Delay On Start = Duration; Number of days, hours, minutes, and seconds to delay after starting the task.</p>
<p>Delay Duration In Seconds</p>	<p>If Delay On Start = Seconds; Number of seconds to delay after starting the task.</p>
<p>Workflow Only</p>	<p>Specification for whether or not to apply the Wait To Start and Delay On Start specifications only if the task is in a Workflow.</p> <p>Options are:</p> <ul style="list-style-type: none"> • -- System Default -- Apply the Wait To Start and Delay On Start specifications as defined by the System Default Wait/Delay Workflow Only system property. (Default is yes.) • Yes Apply the Wait To Start and Delay On Start specifications only if the task is in a Workflow. • No Apply the Wait To Start and Delay On Start specifications whether or not the task is in a Workflow.

Time Options	This section contains time-related specifications for the task.
Late Start	If enabled, and if the task instance starts after the time or period specified, the task instance is flagged as late. You can specify a time or duration to determine a late start (see Late Start Type). To determine whether a task instance started late, open the task instance and locate the Started Late field; the field is checked if the instance started after the specified time. The Started Late field displays in the task instance Details only if the user specified a Late Start in the task Details.
Late Start Type	Required if Late Start is enabled. Options: <ul style="list-style-type: none"> • Time - Flag the task if it starts after the specified time. • Duration - Flag the task if it starts a certain amount of time after the programmed start time. The task must have a specific start time.
Late Start Time	If Late Start Type = Time; Time after which the task start time is considered late. Use HH:MM, 24-hour time.
Late Start Duration	If Late Start Type = Duration; Duration (amount of relative time) after which the task is considered to have started late. For a task within a workflow, the duration is the period between the time the workflow starts and the time the task itself starts. For example, a task might have a Late Start Duration of 60 minutes. If the workflow starts at 9:00 a.m. but the task itself does not start until 10:30, the task has started late. For a task that is not within a workflow, Late Start Duration has meaning only if the task has been held upon starting. For example, if a task has a Late Start Duration of 60 minutes and the Hold on Start field is enabled, if the task is not released from hold within the amount of time specified in the Late Start Duration field, the task has started late.
Late Finish	If enabled, and if the task instance finishes after the time or period specified, the task instance is flagged as late. You can specify a time or duration to determine a late finish (see Late Finish Type). To determine whether a task instance finished late, open the task instance and locate the Finished Late field; the field is checked if the instance finished after the specified time or lasted longer than expected. This field only appears on the task instance if the user specified a Late Finish in the task definition.
Late Finish Type	Required if Late Finish is enabled. Options: <ul style="list-style-type: none"> • Time - Flag the task if it finishes after the specified time (see Late Finish Time). • Duration - Flag the task if it finishes a certain amount of time after the programmed finish time (see Late Finish Duration). The task must have a specific finish time.
Late Finish Time	If Late Finish Type = Time; Time after which the task finish time is considered late. Use HH:MM, 24-hour time.
Late Finish Duration	If Late Finish Type = Duration; Longest amount of time this task instance should take to run.
Early Finish	If enabled, and if the task instance finishes before the time or period specified, the task instance is flagged as early. You can specify a time or duration to determine an early finish (see Early Finish Type). To determine whether a task instance finished early, open the task instance and locate the Finished Early field; the field is checked if the instance finished before the specified time or did not last as long as expected. This field only appears on the task instance if the user added Early Finish specifications to the task definition.

Early Finish Type	<p>Required if Early Finish is enabled.</p> <p>Options:</p> <ul style="list-style-type: none"> • Time - Flag the task if it finishes before the specified time (see Early Finish Time). • Duration - Flag the task if it finishes a certain amount of time before the programmed finish time (see Early Finish Duration). The task must have a specific finish time.
Early Finish Time	<p>If Early Finish Type = Time; Time before which the task finish time is considered early. That is, enter a time at which the task should still be running. Use HH:MM, 24-hour time.</p>
Early Finish Duration	<p>If Early Finish Type = Duration; Shortest amount of time this task instance should take to run.</p>
User Estimated Duration	<p>Estimated amount of time it should normally take to run this task. The Controller uses this information to calculate the User Estimated End Time on a task instance record.</p>
Critical Path Options	<p>This section contains Critical Path-related specifications for the task.</p>
CP Duration	<p>Optional; Allows you to override the estimated Critical Path Duration of the task when running in a Workflow; used in conjunction with the CP Duration Unit field. In most cases, this field should be left blank, which implies that the Controller will estimate the Critical Path Duration based on historical executions. Valid values are any integer equal to or greater than 0. Variables and Functions are supported.</p>
CP Duration (Resolved)	<p>Displays the current resolved value of the CP Duration field, which may contain variables or functions that will be displayed as unresolved until the task instance starts. The CP Duration (Resolved) field can continue to change value until the task instance starts, at which time CP Duration will display as resolved and CP Duration (Resolved) will no longer be visible unless there was an issue resolving the variables and/or functions contained within CP Duration. If the Controller is unable to resolve CP Duration or it resolves to an invalid value, CP Duration will be ignored and the Controller will estimate the Critical Path Duration based on historical executions.</p>
CP Duration Unit	<p>Type of CP Duration; used in conjunction with the CP Duration field. For example, for a CP Duration of two minutes, specify 2 in the CP Duration field and select Minutes in this field.</p> <p>Options:</p> <ul style="list-style-type: none"> • Seconds • Minutes • Hours <p>Default is Minutes.</p>
Statistics	<p>This section contains time-related statistics for task instances of the task.</p>
First Time Ran	<p>System-supplied; date and time this task first ran.</p>
Last Time Ran	<p>System-supplied; date and time the task last ran.</p>
Last Instance Duration	<p>System-supplied; amount of time the task took to run the last time it ran.</p>
Lowest Instance Time	<p>System-supplied; shortest amount of time this task has taken to run.</p>

Average Instance Time	System-supplied; average amount of time this task takes to run.										
Highest Instance Time	System-supplied; longest amount of time this task has taken to run.										
Number of Instances	System-supplied; number of instances in the database for this task.										
Buttons	This section identifies the buttons displayed above and below the Task Details that let you perform various actions.										
Save	Saves a new task record in the Controller database.										
Update	Saves updates to the record.										
New	Displays empty (except for default values) Details for creating a new task.										
Launch Task	Manually launches the task.										
View Parents	Displays a list of any parent Workflow tasks for this task.										
Copy	Creates a copy of this task, which you are prompted to rename.										
Refresh	Refreshes any dynamic data displayed in the Details.										
Close	For pop-up view only; closes the pop-up view of this task.										
Tabs	This section identifies the tabs across the top of the Task Details that provide access to additional information about the task.										
Variables	Lists all variables associated with this record.										
Actions	<p>Allows you to specify actions that the Controller will take automatically based on events that occur during the execution of this task.</p> <p>Events are:</p> <ul style="list-style-type: none"> • Task instance status • Exit codes • Late start • Late finish • Early finish <p>Actions are:</p> <table border="1" data-bbox="272 1465 1484 1829"> <tr> <td>Abort Action</td> <td>Abort the task if certain events occur. For details, see Abort Actions.</td> </tr> <tr> <td>Email Notification</td> <td>Send an email if certain events occur. For details, see Email Notification Actions.</td> </tr> <tr> <td>Set Variable</td> <td>Used in tasks and workflows to set a variable based on the occurrence of certain events. For details, see Creating a Set Variable Action within a Task or Workflow.</td> </tr> <tr> <td>SNMP Notification</td> <td>Send an email if certain events occur. For details, see SNMP Notification Actions.</td> </tr> <tr> <td>System Operation</td> <td>Run an Universal Controller system operation based on specified conditions. For details, see System Operation Actions.</td> </tr> </table>	Abort Action	Abort the task if certain events occur. For details, see Abort Actions .	Email Notification	Send an email if certain events occur. For details, see Email Notification Actions .	Set Variable	Used in tasks and workflows to set a variable based on the occurrence of certain events. For details, see Creating a Set Variable Action within a Task or Workflow .	SNMP Notification	Send an email if certain events occur. For details, see SNMP Notification Actions .	System Operation	Run an Universal Controller system operation based on specified conditions. For details, see System Operation Actions .
Abort Action	Abort the task if certain events occur. For details, see Abort Actions .										
Email Notification	Send an email if certain events occur. For details, see Email Notification Actions .										
Set Variable	Used in tasks and workflows to set a variable based on the occurrence of certain events. For details, see Creating a Set Variable Action within a Task or Workflow .										
SNMP Notification	Send an email if certain events occur. For details, see SNMP Notification Actions .										
System Operation	Run an Universal Controller system operation based on specified conditions. For details, see System Operation Actions .										
Virtual Resources	Lists all Virtual Resources to which this task is assigned.										

Mutually Exclusive	Lists all tasks that have been set to be mutually exclusive of this task.
Instances	Displays a list of all instances of this Workflow.
Triggers	Lists all triggers that have been defined to launch this task. Also allows you to add new triggers. If you add a new trigger from this location, the Controller automatically constructs a default trigger name as follows: <current task name>#TRIGGER#. You can change the default name if desired. For instructions on creating triggers, see Triggers .
Notes	Lists all notes associated with this record.
Versions	Stores copies of all previous versions of the current record. See Record Versioning .

Viewing an Application Control Task Instance

When an Application Control task is launched, the Controller creates a task instance record of that task.

A task instance contains detailed information about a single execution of that task.

You can access a task instance from:

- **Instances tab** on the [Application Control Task Details](#) for that task
- [Activity list](#)
- [Task Instances list](#)

Application Control Task Instance Details

The following Application Control Task Instance Details contains information on the execution of the task shown in the [Application Control Task Details](#).

Application Control Task Instance Details Field Descriptions

The following table describes the fields, buttons, and tabs that display in Application Control Task Instance Details.

Field Name	Description
General	This section contains general information about the task instance.
Instance Name	Name of this task instance.
Reference Id	System-supplied; the Controller increments this number each time the task is run.
Task	Name of the task that was run to create this task instance. Click the icon to display Task Details for the task.


Invoked by	<p>System-supplied; how the task instance was launched.</p> <p>Options:</p> <ul style="list-style-type: none"> • Trigger: (Trigger Name) Instance was launched by the named trigger. • Workflow: (Workflow Name) Instance was launched by the named workflow. • Manually Launched Instance was launched by a user. To identify the user, check the Execution User column for that task instance on the Task Instances screen or, on most task instance screens, the Execution User field.
Task Description	User-defined; description of this record. (Maximum = 200 characters.)
Member of Business Services	User-defined; allows you to select one or more Business Services that this record belongs to.
Execution User	System-supplied; if the task was launched manually, the ID of the user who launched it.
Virtual Resource Priority	<p>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.</p> <p>Options: 1 (high) - 20 (low).</p> <p>Default is 10.</p>
Hold Resources on Failure	If enabled, the task instance will continue to hold Renewable resources if the task instance fails. Renewable resources will be returned only if the task instance status is either Complete, Finished, or Skipped.
Status	This section contains information about the current status of the task instance.
Status	System-supplied; see Task Instance Statuses .
Status Description	System-supplied; additional information, if any, about the status of the task instance.
Operational Memo	User-defined operational memo.
Critical	Indicates that this task is in the Critical Path of a workflow.
Wait Until Time	Amount of time calculated to wait before the task was started, based on Wait To Start and Delay On Start times.
Queued Time	System-supplied; Date and time the task was queued for processing.
Trigger Time	System-supplied; Date and time the task instance was triggered.
Launch Time	System-supplied; Date and time the task instance was launched.

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.
Application Control Details	This section contains assorted detailed information about the task instance.
Application	Protected if auto-generated; Name of the Application resource record. The Application resource defines where the software application is running; it also defines the start, stop, and query commands for the application.
Command	Protected if auto-generated; command that this task is executing against the software application. Options: <ul style="list-style-type: none"> • Query • Start • Stop
Generated	System-supplied; protected. If selected, indicates that this Application Control task was generated automatically when the Application resource record was submitted.
Wait / Delay Options	This section contains specifications for waiting to start and/or delaying on start the task.
Wait To Start	Amount of time to wait before starting a task from the time that it was launched. Options are: <ul style="list-style-type: none"> • – None – • Time • Relative Time • Duration • Seconds
Wait Time	If Wait To Start = Time or Relative Time; Number of hours and minutes to wait before starting the task.

<p>Wait Day Constraint</p>	<p>If Wait Time = Time or Relative Time; Specification for whether or not to advance the wait time to another day.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • -- None -- <ul style="list-style-type: none"> • If Wait To Start = Time; Advance to the next day if calculated wait time is before the current time. • If Wait To Start = Relative Time; Advance to the next day if calculated wait time is before the Trigger/Launch Time. • Same Day Do not advance day. • Next Day Advance to the next day. • Next Business Day Advance to the next business day. • Sunday If today is not Sunday, advance to next Sunday. • Monday If today is not Monday, advance to next Monday. • Tuesday If today is not Tuesday, advance to next Tuesday. • Wednesday If today is not Wednesday, advance to next Wednesday. • Thursday If today is not Thursday, advance to next Thursday. • Friday If today is not Friday, advance to next Friday. • Saturday If today is not Saturday, advance to next Saturday. <p>Default is – None --.</p>
<p>Wait Duration</p>	<p>If Wait To Start = Duration; Number of days, hours, minutes, and seconds to wait before starting the task.</p>
<p>Wait Duration In Seconds</p>	<p>If Wait To Start = Seconds; Number of seconds to wait before starting the task.</p>
<p>Delay On Start</p>	<p>Amount of time to delay the start of a task, after it has been launched, from the time that it is eligible to start; that is, all dependencies have been met. For example: it is not being held, it is not waiting on any predecessors, or there is no wait time specified.</p> <p>Options are:</p> <ul style="list-style-type: none"> • – None – • Duration • Seconds
<p>Delay Duration</p>	<p>If Delay On Start = Duration; Number of days, hours, minutes, and seconds to delay after starting the task.</p>
<p>Delay Duration In Seconds</p>	<p>If Delay On Start = Seconds; Number of seconds to delay after starting the task.</p>
<p>Time Options</p>	<p>This section contains time-related specifications for the task instance.</p>
<p>Late Start</p>	<p>If enabled, and if the task instance starts after the time or period specified, the task instance is flagged as late. You can specify a time or duration to determine a late start (see Late Start Type). To determine whether a task instance started late, open the task instance and locate the Started Late field; the field is checked if the instance started after the specified time. The Started Late field displays in the task instance Details only if the user specified a Late Start in the task Details.</p>
<p>Started Late</p>	<p>System-supplied; this field is flagged if the task started later than the time specified in the Late Start fields.</p>

<p>Late Start Type</p>	<p>Required if Late Start is enabled.</p> <p>Options:</p> <ul style="list-style-type: none"> • Time - Flag the task if it starts after the specified time. • Duration - Flag the task if it starts a certain amount of time after the programmed start time. The task must have a specific start time.
<p>Late Start Time</p>	<p>If Late Start Type = Time; Time after which the task start time is considered late. Use HH:MM, 24-hour time.</p>
<p>Late Start Duration</p>	<p>If Late Start Type = Duration; Duration (amount of relative time) after which the task is considered to have started late.</p> <p>For a task within a workflow, the duration is the period between the time the workflow starts and the time the task itself starts. For example, a task might have a Late Start Duration of 60 minutes. If the workflow starts at 9:00 a.m. but the task itself does not start until 10:30, the task has started late.</p> <p>For a task that is not within a workflow, Late Start Duration has meaning only if the task has been held upon starting. For example, if a task has a Late Start Duration of 60 minutes and the Hold on Start field is enabled, if the task is not released from hold within the amount of time specified in the Late Start Duration field, the task has started late.</p>
<p>Late Finish</p>	<p>If enabled, and if the task instance finishes after the time or period specified, the task instance is flagged as late. You can specify a time or duration to determine a late finish (see Late Finish Type). To determine whether a task instance finished late, open the task instance and locate the Finished Late field; the field is checked if the instance finished after the specified time or lasted longer than expected. This field only appears on the task instance if the user specified a Late Finish in the task definition.</p>
<p>Finished Late</p>	<p>System-supplied; this field is flagged if the task finished later than the time or duration specified in the Late Finish fields.</p>
<p>Late Finish Type</p>	<p>Required if Late Finish is enabled.</p> <p>Options:</p> <ul style="list-style-type: none"> • Time - Flag the task if it finishes after the specified time (see Late Finish Time). • Duration - Flag the task if it finishes a certain amount of time after the programmed finish time (see Late Finish Duration). The task must have a specific finish time.
<p>Late Finish Time</p>	<p>If Late Finish Type = Time; Time after which the task finish time is considered late. Use HH:MM, 24-hour time.</p>
<p>Late Finish Duration</p>	<p>If Late Finish Type = Duration; Longest amount of time this task instance should take to run.</p>
<p>Early Finish</p>	<p>If enabled, and if the task instance finishes before the time or period specified, the task instance is flagged as early. You can specify a time or duration to determine an early finish (see Early Finish Type). To determine whether a task instance finished early, open the task instance and locate the Finished Early field; the field is checked if the instance finished before the specified time or did not last as long as expected. This field only appears on the task instance if the user added Early Finish specifications to the task definition.</p>
<p>Finished Early</p>	<p>System-supplied; this field is flagged if the task finished earlier than the time specified in the Early Finish fields.</p>
<p>Early Finish Type</p>	<p>Required if Early Finish is enabled.</p> <p>Options:</p> <ul style="list-style-type: none"> • Time - Flag the task if it finishes before the specified time (see Early Finish Time). • Duration - Flag the task if it finishes a certain amount of time before the programmed finish time (see Early Finish Duration). The task must have a specific finish time.

Early Finish Time	If Early Finish Type = Time; Time before which the task finish time is considered early. That is, enter a time at which the task should still be running. Use HH:MM, 24-hour time.
Early Finish Duration	If Early Finish Type = Duration; Shortest amount of time this task instance should take to run.
Critical Path Options	This section contains Critical Path-related specifications for the task.
CP Duration	Optional; Allows you to override the estimated Critical Path Duration of the task when running in a Workflow; used in conjunction with the CP Duration Unit field. In most cases, this field should be left blank, which implies that the Controller will estimate the Critical Path Duration based on historical executions. Valid values are any integer equal to or greater than 0. Variables and Functions are supported.
CP Duration (Resolved)	Displays the current resolved value of the CP Duration field, which may contain variables or functions that will be displayed as unresolved until the task instance starts. The CP Duration (Resolved) field can continue to change value until the task instance starts, at which time CP Duration will display as resolved and CP Duration (Resolved) will no longer be visible unless there was an issue resolving the variables and/or functions contained within CP Duration . If the Controller is unable to resolve CP Duration or it resolves to an invalid value, CP Duration will be ignored and the Controller will estimate the Critical Path Duration based on historical executions.
CP Duration Unit	Type of CP Duration; used in conjunction with the CP Duration field. For example, for a CP Duration of two minutes, specify 2 in the CP Duration field and select Minutes in this field. Options: <ul style="list-style-type: none"> • Seconds • Minutes • Hours Default is Minutes.
Statistics	This section contains time-related statistics for the task instance.
User Estimated End Time	System-supplied; if the user entered information into the User Estimated Duration field in the task Details, the Controller uses this information to calculate an end time for the task instance, based on the date/time the task instance started.
Average Estimated End Time	System-supplied; average estimated end time of the task instance, calculated by the Controller based on the date/time the task instance started.
Shortest Estimated End Time	System-supplied; shortest estimated end time of the task instance, calculated by the Controller based on the date/time the task instance started.
Longest Estimated End Time	System-supplied; longest estimated end time of the task instance, calculated by the Controller based on the date/time the task instance started.
Projected End Time	System-supplied; projected end time of the task instance, calculated by the Controller based on the projected end time of its predecessor (or the maximum projected end time of all its predecessors, if more than one path exists to that task instance) plus its estimated critical path duration .
Buttons	This section identifies the buttons displayed above and below the Task Instance Details that let you perform various actions.
Update	Saves updates to the record.

Force Finish	See Force Finishing a Task .
Hold	Places the task instance on Hold (see Putting a Task on Hold).
Skip	For tasks loaded into the schedule that have not yet run; allows you to tell the Controller to skip this task. See Skipping a Task .
Re-run	See Re-running a Task .
View Parent	Displays the task instance Details for the parent Workflow of this task instance.
Retrieve Output	See Retrieving Output .
Delete	<p>Deletes the current record.</p> <div style="background-color: #ffffcc; padding: 10px; border: 1px solid #ccc;"> <p> Note You cannot delete a task if it is either:</p> <ul style="list-style-type: none"> Specified in an enabled Trigger. The only task specified in a disabled Trigger. </div>
Refresh	Refreshes any dynamic data displayed in the Details.
Close	For pop-up view only; closes the pop-up view of this task instance.
Tabs	This section identifies the tabs across the top of the Task Instance Details that provide access to additional information about the task instance.
Virtual Resources	Lists all Virtual Resources to which this task is assigned.
Exclusive Requests	Lists all records in the Exclusive Requests table (<code>ops_exclusive_order</code>) for this task instance.
Notes	Lists all notes associated with this record.

Running an Application Control Task

You can run an Application Control task:

- Manually, by clicking the [Launch Task](#) or [Launch Task with Variables](#) button in the Application Control Tasks list or Application Control Task Details [Action menu](#).
- As part of a [Workflow](#).
- [Specify triggers](#) that run the task automatically based on times or events.

Monitoring Task Execution

You can monitor all system activity from the [Activity Monitor](#) and can view activity history from the [History list](#).

Application Monitor Triggers

- [Overview](#)
- [Built-In Variables](#)
- [Creating an Application Monitor Trigger](#)
 - [Application Monitor Trigger Details](#)
 - [Application Monitor Trigger Details Field Descriptions](#)

Overview

The Application Monitor Trigger allows you to trigger one or more tasks based on the status of:

- A specific [application resource](#).
- One or more [application resources](#), based on selection criteria you supply.

You can launch any number of tasks when the conditions in the trigger are satisfied.

When creating a trigger, if you specify [Application Monitor Condition](#) = ALL, and select all Application types, the trigger monitors all Application resource records you have defined. Any time any one of them goes to any of the statuses you specified in the [Status\(es\)](#) field, the trigger launches the task(s) specified in the [Task\(s\)](#) field. For example, you might use this trigger to send an email notification to technical support if any of the monitored applications goes into the Start Failure status.

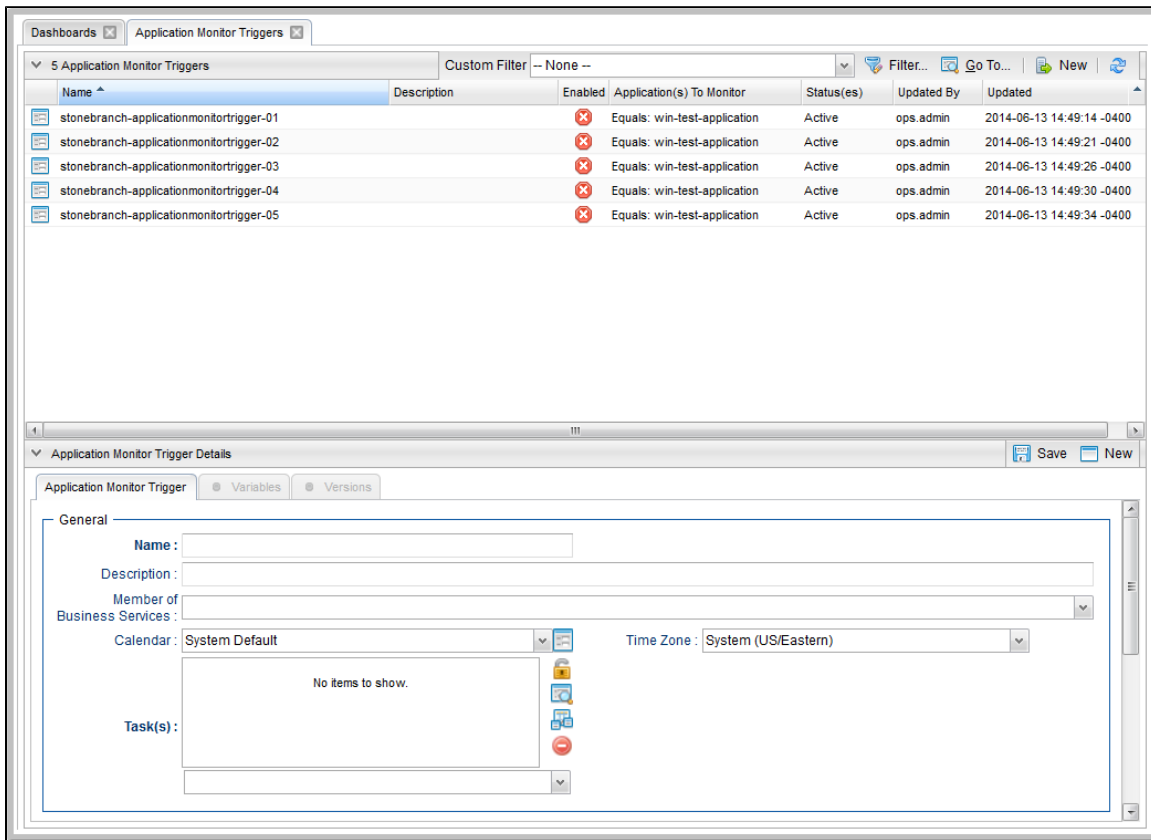
Built-In Variables

[Application Monitor built-in variables](#) are provided to pass information about the Application being monitored into the task(s) being launched by the trigger. You can pass the information into the launched tasks by including the variables in a text field in the task definition.

Creating an Application Monitor Trigger

Step 1 From the [Automation Center](#) navigation pane, select **Triggers > Application Monitor Triggers**. The Application Monitor Triggers list displays.

Below the list, Application Monitor Trigger Details for a new Application Monitor trigger displays. (You also can click the **New** button to display Application Monitor Trigger Details for a new Application Monitor trigger.)



Step 2 Enter/select Details for a new Application Monitor trigger, using the [field descriptions](#) below as a guide.

- Required fields display in **boldface**.
- Default values for fields, if available, display automatically.

To display more of the Details fields on the screen, you can temporarily [hide the list](#).

Note
 If you view [Application Monitor Trigger Details](#) for an existing Application Monitor trigger by clicking a trigger in the list, and then want to create a new Application Monitor trigger, you must click the **New** button that displays above and below the Details.

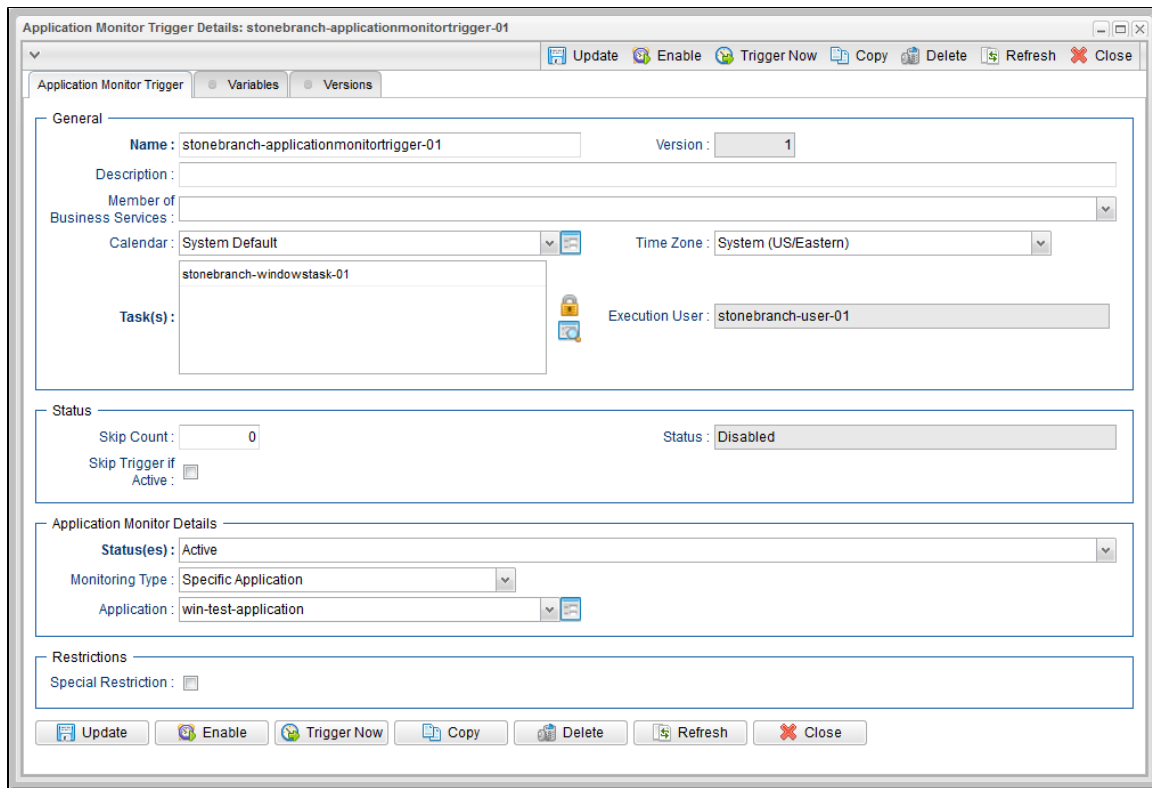
Step 3 Click the **Save** button. The trigger is added to the database, and all buttons and tabs in the Task Details are enabled.

Step 4 If appropriate, repeat these steps for any additional triggers you want to add.

Step 5 Enable the trigger(s) as desired.

Application Monitor Trigger Details

The following Application Monitor Trigger Details is for an existing Application Monitor trigger. See the [field descriptions](#), below, for a description of the fields that display in the Application Monitor Trigger Details.



Application Monitor Trigger Details Field Descriptions

The following table describes the fields, buttons, and tabs that display in the Application Monitor Trigger Details.

Field Name	Description
General	This section contains general information about the trigger.
Name	Name used within the Controller to identify this trigger. It can contain a maximum of 255 alphanumeric characters. It is the responsibility of the user to develop a workable naming scheme for triggers.
Version	System-supplied; version number of the current record, which is incremented by the system every time a user updates a record. Click the Versions tab to view previous versions. For details, see Record Versioning .
Description	User-defined; description of this trigger.
Member of Business Services	User-defined; allows you to select one or more Business Services that this record belongs to.
Calendar	Calendar that defines the business days, holidays, and other special days that determine the run dates for the task(s) specified in the trigger. Select a Calendar from the drop-down list of all existing Calendars. To display detailed information about a selected calendar, click the Details icon next to the Calendar field.
Time Zone	User-defined; allows you to specify the time zone that will be applied to the time(s) specified in the trigger. For example, if you specify 23:00 and a time zone of Canada/Central, the task will run at 11:00 p.m. Central Canada time.

Task(s)	Name of the task(s) being triggered when this trigger is satisfied. When selecting tasks from the Details, click on the lock icon to unlock the field and select tasks .
Execution User	Read-only; execution user that has been selected (via the Assign Execution User action) to override the execution user of task instances being launched by the trigger.
Status	This section contains information about the current status of the trigger.
Skip Count	User-defined; allows you to specify that the Controller should skip the next <i>N</i> times this task is triggered.
Skip Trigger if Active	User-defined; allows you to specify that the Controller should skip the next run of the specified task(s) if the previous run has not gone to a Complete status (that is, it is still active).
Status	User-defined; indication of whether the trigger is enabled (checked) or disabled (not checked). The user enables and disables the trigger by clicking the Enable / Disable buttons. Only enabled triggers are processed by the Controller.
Enabled By	System-supplied. If Status = Enabled, ID of the user who most recently enabled this trigger.
Application Monitor Details	This section contains assorted detailed information about the trigger time.
Status(es)	System-supplied; application status being monitored for. One or more of the following: <ul style="list-style-type: none"> • Inactive - The initial state of the Application. The Application is stopped and unmonitored. • Start Failure - The Agent experienced a failure while attempting to execute the Start command. • Starting - The Start command was executed and the Controller is waiting for Query command response. • Active - The Query command response is reporting that the Application is Active. • Impaired - The Query command response is reporting that the Application is experiencing a problem and is possibly down. • Query Overdue - The Agent is late sending the Controller an updated Query command response.
Monitoring Type	Indicates whether you are monitoring one specific Application resource or want to provide selection parameters to monitor multiple Application resources. See Applications for information about setting up Application resources. Options: <ul style="list-style-type: none"> • Specific Application - Use the Application field to browse for and select the Application resource you want to monitor. • General Applications - Use the Application Monitor Condition and Application Type(s) fields to provide parameters for selecting which Application resources you want to monitor.
Application	If Monitoring Type = Specific Application, name of a specific application resource to be monitored.
Application Monitor Condition	If Monitoring Type = General Application(s), allows you to specify selection parameters: <ul style="list-style-type: none"> • ALL - Monitor all Application resources. • Starts With - Monitor all Application resources whose name starts with the string you provide in the Condition Value field. • Contains - Monitor all Application resources whose name contains the string you provide in the Condition Value field. • Ends With - Monitor all Application resources whose name ends with the string you provide in the Condition Value field.
Condition Value	If Application Monitor Condition = Starts With, Contains, or Ends With; specifies the search string.

Application Type(s)	<p>If Monitoring Type = General Application(s), type(s) of applications to monitor. Options:</p> <ul style="list-style-type: none"> • Windows Service • Linux/Unix Daemon • z/OS Started Task
Restrictions	This section specifies any restrictions that apply to the trigger.
Special Restriction	<p>Enable this field in order to specify additional parameters that tell the Controller how to handle exceptions, such as when the trigger is satisfied on a holiday or non-business day. You can specify simple and/or complex restrictions. For example, you can specify a Simple Restriction that disables the trigger if it is satisfied on a holiday identified in the calendar and/or a Complex Restriction that disables the trigger on the last business day of every month.</p>
Simple Restriction	<p>If enabled, allows you to specify an action (see Action field) such as Do Not Trigger on a non-business day or holiday (see Situation field). For example, do not trigger on a non-business day.</p>
Situation	<p>If Simple Restriction is enabled, allows you to select the situation that causes the system to initiate the action specified in the Action field.</p> <p>Options:</p> <ul style="list-style-type: none"> • On Non Business Day • On Holiday
Action	<p>If Special Restriction is enabled, allows you to select an action to take on a non-business day or holiday (see Situation field).</p> <p>Options:</p> <ul style="list-style-type: none"> • Do Not Trigger • Next Day (run on the next day) • Next Business Day (run on the next business day, as defined in the calendar) • Previous Day (run on the previous day) • Previous Business Day (run on the previous business day, as defined in the calendar)
Complex Restriction	<p>If enabled, allows you to specify a set of parameters that determine one or more situations when this trigger should not be satisfied. Used in conjunction with the following fields: Restriction Mode, Restriction Adjective, Restriction Noun, Restriction Qualifier. For example, you may specify that you do not want to satisfy this trigger on the last business day of the year or the first day of each month.</p>
Restriction Mode	<p>If both Simple Restriction and Complex Restriction are enabled, specifies whether you want to use both restriction types (AND) or one or the other (OR).</p> <p>Options:</p> <ul style="list-style-type: none"> • And • Or
Restriction Adjective	<p>If Complex Restriction is enabled, the type of selection.</p> <p>Options:</p> <ul style="list-style-type: none"> • Every • 1st • 2nd • 3rd • 4th • Last <p>Example: The last business day of the month.</p>

<p>Restriction Noun</p>	<p>If Complex Restriction is enabled, the day you want to select.</p> <p>Options:</p> <ul style="list-style-type: none"> • Sunday through Saturday • Day • Business Day • Custom Day <p>Example: The last business day of the month.</p>
<p>Restriction Qualifier</p>	<p>If Complex Restriction is enabled, the period you are selecting from.</p> <p>Options:</p> <ul style="list-style-type: none"> • Month • Year • January through December • Custom period <p>Example: The last quarter of the year.</p>
<p>Buttons</p>	<p>This section identifies the buttons displayed above and below the Trigger Details that let you perform various actions.</p>
<p>Save</p>	<p>Saves a new task record in the Controller database.</p>
<p>Update</p>	<p>Saves updates to the record.</p>
<p>New</p>	<p>Displays empty (except for default values) Details for creating a new trigger.</p>
<p>Enable</p>	<p>Activates this trigger and writes your User ID to the Enabled By field.</p>
<p>Disable</p>	<p>Deactivates this trigger.</p>
<p>Trigger Now</p>	<p>Immediately triggers all the tasks specified in this trigger.</p>
<p>Copy</p>	<p>Creates a copy of this trigger, which you are prompted to rename.</p>
<p>Delete</p>	<p>Deletes the current record.</p>
<p>Refresh</p>	<p>Refreshes any dynamic data displayed in the Details.</p>
<p>Close</p>	<p>For pop-up view only; closes the pop-up view of this trigger.</p>
<p>Tabs</p>	<p>This section identifies the tabs across the top of the Trigger Details that provide access to additional information about the trigger.</p>
<p>Variables</p>	<p>Lists all variables associated with this record.</p>
<p>Versions</p>	<p>Stores copies of all previous versions of the current record. See Record Versioning.</p>