

Build a Custom Action

On this page:

- [Custom Notifications and Custom Actions](#)
- [Creating a Custom Action](#)
- [Contents of the custom.xml File](#)
- [Information Passed to the Custom Action Script from AppDynamics](#)
- [Sample Custom Action Script](#)

Related pages:

- [Actions](#)
- [Custom Actions](#)
- [Policies](#)
- [Configure Policies](#)

You can set up a custom action on the controller instance to integrate notification of AppDynamics health rule violations and events with an alerting or ticketing system. Use a push approach by creating custom notifications that pass the information to your alerting system.

Custom Notifications and Custom Actions

A custom notification lets you integrate alerts about AppDynamics health rule violations and events into your own alerting system. This integration extension requires:

- A `custom.xml` file that provides information about the custom notification
- An executable script that accepts parameters from AppDynamics about the events and health rule violations that trigger an alert
- Configuring AppDynamics events or policies to trigger the custom notification via a custom action

This topic describes how to create the script and the XML file. See the documentation on the Alert & Response features in the *Related pages* above for information on how to trigger the action.

Creating a Custom Action

Create the script

For each custom action that you want to implement, create an executable script (.bat extension for Windows, .sh extension for Linux) that can accept and process the parameters passed to it by AppDynamics. See [Information Passed to the Custom Action Script from AppDynamics](#) for details on the parameters. For each script:

- Set correct executable permissions for the shell scripts in a Linux environment. For example, `chmod 770 script1.sh`.
- Ensure that the script file has the correct character encoding. This is especially important when creating a Unix shell script on a Windows machine.

Install the script on an On-Premises Controller

To install the script on an on-premises controller:

1. At the top level of the Controller installation directory, create a directory named "custom" with a sub-directory named "actions".

```
<controller_home>/custom/actions
```

2. In the `<controller_home>/custom/actions` directory, create a subdirectory for each custom action script that you will install. For example, for an action that interfaces with a JIRA system.

```
<controller_home>/custom/actions/jira
```

3. Move the script to the appropriate subdirectory that you created in step 2.

Create the XML File

1. Create a `custom.xml` file that describes the location and name of your custom action script. See [Contents of the custom.xml File](#).
2. For an on-premises Controller, move the file to the `<controller_home>/custom/actions` directory. For a SaaS Controller, contact your AppDynamics sales representative for instructions.

Verify on the Script on an on-premises Controller

1. After you have installed the script and the `custom.xml` file, restart the Controller.
2. Verify the script manually. To verify the script:
 - a. Open a command-line console on the Controller host machine.
 - b. Execute the script file from the command line console.

Create the Custom Action

Create the custom action in the AppDynamics UI to arrange how the custom action will be triggered. See [Custom Actions](#).

Contents of the custom.xml File

The `custom.xml` file has an `<actions>` element for every custom action on the controller.

The `<type>` element contains the subdirectory that contains the script file.

The `<executable>` element contains the name of the script.

Sample custom.xml file

```

<custom-actions>
  <action>
    <type>jira</type>
    <executable>script1.bat</executable>
  </action>
  <action>
    <type>bugzilla</type>
    <executable>script2.sh</executable>
  </action>
</custom-actions>

```

Information Passed to the Custom Action Script from AppDynamics

The custom action script must handle the parameters that the Controller passes from the health rule violation or other event. The parameter values are passed as an array of strings.

The parameters are passed as \$0 for the script name, then \$1, \$2, . . . \$n. \$1 is the first parameter (application name), \$2 is the application id, and so on in the order in which they are documented in the sections below.

Health rule violations have a different set of parameters from events.

Parameters passed by a health rule violation

The parameters describe the violated health rule violation that triggered the action.

The total number of elements in the array depends on the number of entities evaluated by the health rule and the number of triggered conditions per evaluation entity. Examples of evaluation entities are application, tier, node, business transaction, JMX. For each evaluation entity, the script expects the entity type, entity name, entity id, number of triggered conditions, and for each triggered condition, the set of condition parameters.

The parameter values are passed in the order in which they are described below.

Structure of Parameters Sent by a Health Rule Violation

- APP_NAME
- APP_ID
- PVN_ALERT_TIME
- PRIORITY
- SEVERITY // INFO, WARN, ERROR
- HEALTH_RULE_NAME
- HEALTH_RULE_ID
- PVN_TIME_PERIOD_IN_MINUTES
- AFFECTED_ENTITY_TYPE
- AFFECTED_ENTITY_NAME
- AFFECTED_ENTITY_ID
- NUMBER_OF_EVALUATION_ENTITIES—The following parameters are passed for each evaluation entity:
 - EVALUATION_ENTITY_TYPE
 - EVALUATION_ENTITY_NAME
 - EVALUATION_ENTITY_ID
 - NUMBER_OF_TRIGGERED_CONDITIONS_PER_EVALUATION_ENTITY—The following parameters are passed for each

triggered condition for this evaluation entity:

- SCOPE_TYPE_x
- SCOPE_NAME_x
- SCOPE_ID_x
- CONDITION_NAME_x
- CONDITION_ID_x
- OPERATOR_x
- CONDITION_UNIT_TYPE_x
- USE_DEFAULT_BASELINE_x
- BASELINE_NAME_x
- BASELINE_ID_x
- THRESHOLD_VALUE_x
- OBSERVED_VALUE_x
- SUMMARY_MESSAGE
- INCIDENT_ID
- DEEP_LINK_URL
- EVENT_TYPE
- ACCOUNT_NAME
- ACCOUNT_ID

Definitions of Parameters Sent by a Health Rule Violation

Health Rule Violation Parameter	Definition
APP_NAME	Name of the business application
APP_ID	Application ID number
PVN_ALERT_TIME	Alert time, such as Thu Dec 22 15:03:56 PST 2011
PRIORITY	Integer designating how urgently a health rule violation should be urgent
SEVERITY	INFO, WARN, or ERROR—In the AppDynamics UI they are called out as
HEALTH_RULE_NAME	Name of the health rule that was violated
HEALTH_RULE_ID	Health rule ID
PVN_TIME_PERIOD_IN_MINUTES	Health rule violation time period in minutes
AFFECTED_ENTITY_TYPE	APPLICATION, APPLICATION_COMPONENT (aka Tier), BUSINESS_TRANSACTION, APPLICATION_DIAGNOSTIC_TRANSACTION
AFFECTED_ENTITY_NAME	The affected entity name
AFFECTED_ENTITY_ID	The affected entity ID
NUMBER_OF_EVALUATION_ENTITIES	Number of entities—Business Transactions, Applications, Transactions—violating the health rule conditions
EVALUATION_ENTITY_TYPE	APPLICATION, APPLICATION_COMPONENT (aka Tier), BUSINESS_TRANSACTION, APPLICATION_DIAGNOSTIC_TRANSACTION
EVALUATION_ENTITY_NAME	The evaluation entity name (for JMX it is the counter name)
EVALUATION_ENTITY_ID	The evaluation entity ID or <NULL> for JMX
NUMBER_OF_TRIGGERED_CONDITIONS_PER_EVALUATION_ENTITY	Number of times to loop through the triggered condition parameters. For each condition is triggered, the parameters repeat for each triggered condition
SCOPE_TYPE_x	The scope of the parameter, whether the scope is the application, APPLICATION_COMPONENT, APPLICATION_DIAGNOSTIC_TRANSACTION

SCOPE_NAME_x	The name of the scope, such as ACME Book Store Applicati
SCOPE_ID_x	The scope ID
CONDITION_NAME_x	The health rule condition name
CONDITION_ID_x	The health rule condition ID
OPERATOR_x	Allowed operators: LESS_THAN, LESS_THAN_EQUALS, G EQUALS, NOT_EQUALS
CONDITION_UNIT_TYPE_x	The condition for the threshold parameter: ABSOLUTE, BAS BASELINE_PERCENTAGE, BASELINE_PERCENTILE
USE_DEFAULT_BASELINE_x	A Boolean parameter (true or false) applies only when the cc
BASELINE_NAME_x	Applicable only when the condition unit type is one of the BA meter is false
BASELINE_ID_x	Applicable only when the condition unit type is one of the BA meter is false
THRESHOLD_VALUE_x	Health rule threshold setting
OBSERVED_VALUE_x	Value that violated the health rule threshold
SUMMARY_MESSAGE	Summary of the notification, such as <i>Health rules have been</i>
INCIDENT_ID	The incident identifier number for this health rule violation. In is defined as int(11) which means it takes four bytes of space 2147483647 max value and -2147483648 min value. One bi
DEEP_LINK_URL	Controller deep link URL, such as: http://<controller-host-url>/#location=APP_ID Append the incident ID to the URL to provide a link to the Cc
EVENT_TYPE	POLICY_OPEN_WARNING, POLICY_OPEN_CRITICAL, P POLICY_CLOSE_CRITICAL, POLICY_UPGRADED, POLI NG, POLICY_CANCELED_CRITICAL, POLICY_CONTINU POLICY_CONTINUES_WARNING
ACCOUNT_NAME	Name of the account in which the action was triggered
ACCOUNT_ID	ID of the account in which the action was triggered

Parameters passed by an event

The parameters describe the event that triggered the action.

The total number of elements in the array depends on the number of event types and event summaries that triggered the action.

The parameter values are passed in the order in which they are described below.

Structure of Parameters Sent by an Event

- APP_NAME
- APP_ID
- EN_TIME
- PRIORITY
- SEVERITY
- EN_NAME
- EN_ID

- EN_INTERVAL_IN_MINUTES
- NUMBER_OF_EVENT_TYPES
The following parameters are passed for each event type:
 - EVENT_TYPE_x
 - EVENT_TYPE_NUM_x
- NUMBER_OF_EVENT_SUMMARIES
The following parameters are passed for each event summary:
 - EVENT_SUMMARY_ID_x
 - EVENT_SUMMARY_TYPE_x
 - EVENT_SUMMARY_SEVERITY_x
 - EVENT_SUMMARY_STRING_x
- DEEP_LINK_URL
- ACCOUNT_NAME
- ACCOUNT_ID

Definitions of Parameters Sent by an Event

Event Notification Parameter	Definition
APP_NAME	Name of the business application
APP_ID	Application ID number
EN_TIME	Event notification time, for example, Wed Jan 04 09:36:55 PST 2012
PRIORITY	Integer designating how urgently a health rule violation should be fixed, with the lowest number (0) the most urgent
SEVERITY	Allowed values: INFO, WARN, or ERROR. In the AppDynamics UI they are called Info, Warning, and Critical
EN_NAME	Name of the event notification
EN_ID	Event notification ID number
EN_INTERVAL_IN_MINUTES	Event notification interval in minutes
NUMBER_OF_EVENT_TYPES	Determines how many times to loop through the event type map parameters
EVENT_TYPE_x	If there is more than one event type, the parameters repeat for each event type, where x increments the number representing the event type
EVENT_TYPE_NUM_x	Number of events of this type
NUMBER_OF_EVENT_SUMMARIES	Number of event summaries in the notification that determines how many times to loop through the event summary parameters
EVENT_SUMMARY_ID_x	Event summary ID number
EVENT_SUMMARY_TIME_x	Event summary time, for example: Wed Jan 04 09:34:13 PST 2012
EVENT_SUMMARY_TYPE_x	Type of event, such as: APPLICATION_CONFIG_CHANGE, APP_SERVER_RESTART, DIAGNOSTIC_SESSION, STALL
EVENT_SUMMARY_SEVERITY_x	Event severity, such as: INFO, WARN or ERROR. In the AppDynamics UI they are called Info, Warning, and Critical.
EVENT_SUMMARY_STRING_x	Event summary string, such as: Application Server environment variables changed
DEEP_LINK_URL	<a href="http://<controller-host-url>/#location=APP_EVENT_VIEWER_MODAL&eventSummaryId={eventSummaryId}">http://<controller-host-url>/#location=APP_EVENT_VIEWER_MODAL&eventSummaryId={eventSummaryId} Append each event summary ID to the URL to provide a link to the Controller UI for this event
ACCOUNT_NAME	Name of the account in which the action was triggered
ACCOUNT_ID	ID of the account in which the action was triggered

Sample Custom Action Script

See the [CreateServiceNow script](#), for an example of a script that creates ServiceNow tickets triggered by AppDynamics health rule violations.