

# IBM Resilient



## Resilient SOAR Platform Integrations

### BigFix Function V1.1.1

Release Date: August 2019

Resilient Functions simplify development of integrations by wrapping each activity into an individual workflow component. These components can be easily installed and then used and combined in Resilient workflows. The Resilient platform sends data to the function component that performs an activity and then returns the results to the workflow. The results can be actioned by scripts, rules, and workflow decision points to dynamically orchestrate the security incident response activities.

This guide describes the BigFix Integration Function.

## What's New

The V1.1.1 release of the Resilient BigFix Function introduces the following new features and enhancements:

- Added new configuration option, 'bigfix\_endpoints\_wait'. This option allows the user to increase the amount of time (in seconds) that the integration will wait for all endpoints to respond to queries.
- Renamed configuration option 'hunt\_results\_limit' to 'bigfix\_hunt\_results\_limit'.

## Overview

BigFix is an endpoint management tool that allows users to keep systems or endpoints in an environment under its control, updated, compatible and free of security issues. It allows for the identification and remediation of a vulnerable endpoint from a central console.

The BigFix integration with the Resilient platform allows querying of a BigFix environment using the REST APIs, where the returned results can be used to remediate issues or hits, such as a malicious path or filename, a service or process name, or a registry key.

The four functions supplied in this Resilient package support the following use cases.

- Beginning with an Indicator of Compromise (IOC) such as a malicious path or filename, service or process name, registry key, or IP address, the BigFix integration allows you to search a BigFix environment for all affected endpoints with a hit, and then update a data table with this information where it can be displayed on the Resilient platform.
- Allows you to query BigFix for all available BigFix properties of an endpoint with a hit, and then attach an XML file with these properties to the Resilient incident.

- Allows you to execute BigFix remediation procedures from the Resilient platform against an endpoint with a hit. These procedures include killing a process, stopping a service, deleting a registry key (Microsoft Windows only) and deleting a file.
- Allows you to query and update the status of a BigFix remediation action from the Resilient platform on an endpoint with a hit.

## Supported artifact types

Artifact type	Associated Resilient Functions	Associated Resilient workflows	Support notes
IP Address	BigFix Artifact	Example: BigFix Query for Artifact	<ul style="list-style-type: none"> <li>• Query only. Remediation option not supported.</li> <li>• MS Windows and Linux.</li> <li>• Queries for IP addresses making connections to endpoints in the BigFix environment.</li> </ul>
Process Name	BigFix Artifact BigFix Remediation	Example: BigFix Query for Artifact Example: BigFix Remediate	<ul style="list-style-type: none"> <li>• MS Windows and Linux.</li> <li>• Case insensitive for MS Windows.</li> <li>• Case sensitive for Linux.</li> </ul>
Service	BigFix Artifact BigFix Remediation	Example: BigFix Query for Artifact Example: BigFix Remediate	<ul style="list-style-type: none"> <li>• Currently MS Windows only.</li> <li>• Query on 'Service name' or 'Display name'.</li> <li>• Case insensitive.</li> </ul>
File path	BigFix Artifact BigFix Remediation	Example: BigFix Query for Artifact Example: BigFix Remediate	<ul style="list-style-type: none"> <li>• MS Windows and Linux.</li> </ul>
Registry Key	BigFix Artifact BigFix Remediation	Example: BigFix Query for Artifact Example: BigFix Remediate	<ul style="list-style-type: none"> <li>• MS Windows only.</li> <li>• Search for key, key + value + no data or key + value + data.</li> <li>• Delete at key level.</li> <li>• Search for values of type string ONLY.</li> <li>• Remediation of keys at root level and keys with subkeys is disallowed This is a safety measure.</li> </ul>

The remainder of this document describes the included functions, how to configure example custom workflows, and any additional customization options.

# Installation

You download the function package to a Resilient integration server, and from there you deploy the functions and components to a Resilient platform. These procedures are provided in the [Resilient Integration Server Guide \(PDF\)](#).

The functions included this package have the following requirements, which are above and beyond those listed in the *Resilient Integration Server Guide*.

- Resilient platform is version 31 or later.
- BigFix version must be 9.5 patch 2, or later.
- A designated BigFix Console Operator account, with the Create Custom Content permission enabled. This account must be configured to access all those endpoints that you wish to have accessible to the Resilient platform.

The following sections provide the procedures for a new installation, an upgrade to an existing installation or if you are currently running the legacy BigFix Integration (not the function).

## New installation

After installing the package on the integration server, Resilient Circuits creates a new section, `fn_bigfix`, in the `app.config` file. You need to edit the following settings in that section.

```
bigfix_url. URL of your BigFix server; for example: https://bigfix-url.com
bigfix_port. Port number of your BigFix server.
bigfix_user. Username of the BigFix Console Operator account used for this
integration.
bigfix_pass. Password for the BigFix Console Operator account.
bigfix_polling_interval. Time in seconds that the integration waits between
polling BigFix to get query results or the final status of the remediation
actions. Default is 30
bigfix_polling_timeout. Time in seconds that the integration waits before
timing out while polling BigFix to get query results or the final status of the
remediation actions. Default is 600
bigfix_hunt_results_limit. Limits the number of results sent to the Resilient
platform. Default is 200.
bigfix_endpoints_wait. Optional setting, with time in seconds, to wait for all
endpoints to respond. Default is 30.
```

## Upgrade

If you have a previous version of the BigFix function, perform the following steps to upgrade the configuration:

1. Stop the integration.
2. Open the resilient-circuits configuration file (`app.config`) in an editor.
3. In the `[fn_bigfix]` section, rename the configuration setting 'hunt\_results\_limit' to 'bigfix\_hunt\_results\_limit'.
4. Also in the `[fn_bigfix]` section, add the configuration setting 'bigfix\_endpoints\_wait' and set it to the desired value. For example:

```
bigfix_endpoints_wait=30
```

5. Restart the integration.

## Convert from the BigFix integration

If a legacy version of the BigFix integration was previously deployed in the Resilient environment this version needs to be uninstalled before attempting installation of the latest version, as follows:

1. Ensure all current BigFix operations initiated from the Resilient platform have completed.
2. Stop Resilient Circuits.
3. Uninstall the Resilient Circuits component:

```
sudo pip uninstall bigfix-integration
```

4. Using sudo, switch to the integration user as follows:

```
sudo su - integration
```

5. Backup the existing resilient-circuits configuration file then edit and remove the [bigfix] section.
6. Backup, if required, then remove the Resilient Circuits BigFix database file.

```
sudo rm ~/.resilient/resilient_bigfix_integration.db
```

7. From the Resilient platform Customizations page, remove the following legacy BigFix objects.

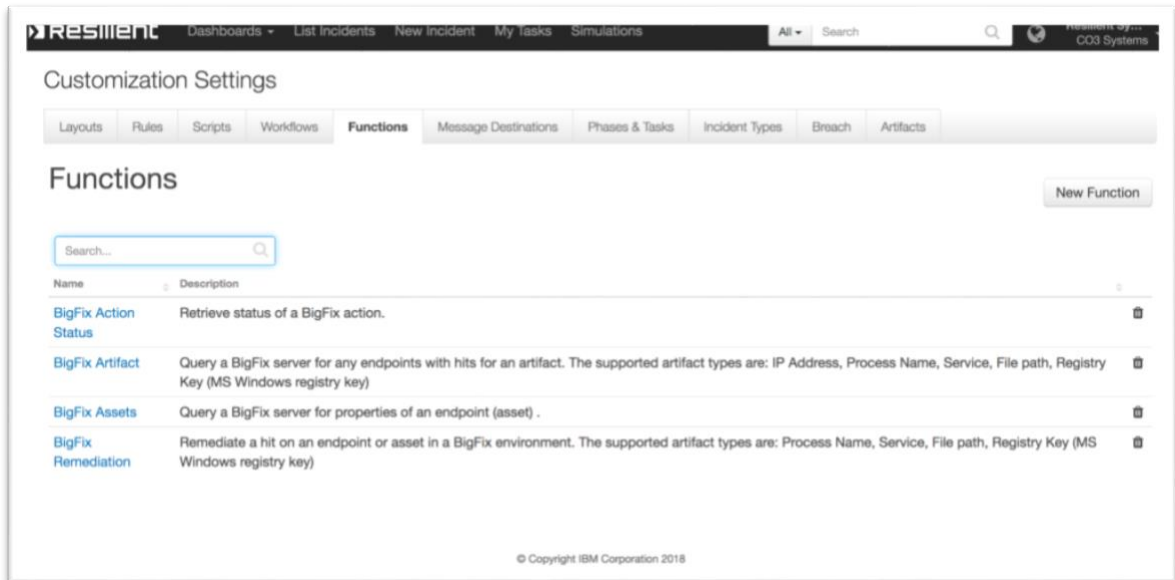
```
Message destinations:
    bigfix_artifact
    bigfix_asset
    bigfix_remediation

Rules:
    BigFix Delete File
    BigFix Delete Registry Key
    BigFix Kill Process
    BigFix Stop Service
    Query BigFix for Artifact
    Retrieve BigFix Resource Details
```

6. Use the procedure for a new installation to install the BigFix function package.

## Function Descriptions

Once the function package deploys the functions, you can view them in the Resilient platform Functions tab, as shown below. The package also includes example workflows and rules that show how the functions can be used. You can copy and modify these workflows and rules for your own needs.



The screenshot displays the Resilient platform interface. At the top, there is a navigation bar with the Resilient logo and several menu items: Dashboards, List Incidents, New Incident, My Tasks, and Simulations. A search bar is also present. Below the navigation bar, the 'Customization Settings' section is active, with a sub-tab for 'Functions'. The 'Functions' tab contains a search bar and a table of functions. The table has two columns: 'Name' and 'Description'. The functions listed are:

Name	Description
BigFix Action Status	Retrieve status of a BigFix action.
BigFix Artifact	Query a BigFix server for any endpoints with hits for an artifact. The supported artifact types are: IP Address, Process Name, Service, File path, Registry Key (MS Windows registry key)
BigFix Assets	Query a BigFix server for properties of an endpoint (asset) .
BigFix Remediation	Remediate a hit on an endpoint or asset in a BigFix environment. The supported artifact types are: Process Name, Service, File path, Registry Key (MS Windows registry key)

At the bottom of the page, there is a copyright notice: © Copyright IBM Corporation 2018.

# Customizations

In the Customization Settings section of the Resilient platform, you can verify that the following BigFix specific functions, workflows, data-table, and rules are available in the Resilient platform by clicking their respective tabs.

## BigFix Artifact

This function performs a query that retrieves a list of endpoints with hits from a BigFix environment.

The screenshot shows the 'Customization Settings' interface. At the top, there's a navigation bar with tabs: Layouts, Rules, Scripts, Workflows, **Functions**, Message Destinations, Phases & Tasks, and Incident Types. Below the tabs, the breadcrumb 'Functions / fn\_bigfix\_artifact' is visible. The main form contains the following fields:

- Name \***: A text input field containing 'BigFix Artifact'.
- API Name \***: A text input field containing 'fn\_bigfix\_artifact'.
- Message Destination \***: A dropdown menu with 'fn\_bigfix' selected.
- Description**: A text area containing 'Query a BigFix server for any endpoints with hits for an artifact.' and 'The supported artifact types are:'.

Below the form, there's an 'Inputs' section with a list of parameters, each with a delete icon (x) on the right:

- bigfix\_artifact\_id
- bigfix\_artifact\_value
- bigfix\_artifact\_type
- bigfix\_incident\_id
- bigfix\_incident\_plan\_status
- bigfix\_artifact\_properties\_name
- bigfix\_artifact\_properties\_value

This function takes the following parameters:

- bigfix\_artifact\_id - Resilient artifact ID
- bigfix\_artifact\_value - Resilient artifact value
- bigfix\_artifact\_type - Resilient artifact type
- bigfix\_incident\_id - Resilient incident ID
- bigfix\_incident\_plan\_status - Resilient incident status

- bigfix\_artifact\_properties\_name - Resilient artifact properties name; optional, used for registry key value name (MS Windows)
- bigfix\_artifact\_properties\_value - Resilient artifact properties name; optional, used for registry key value data (MS Windows)

The example workflow (object type = Artifact) that calls this function is “Example: BigFix Query for Artifact”.

The parameter assignments are done in the Pre-Process Script tab.

Workflows / Example: BigFix Query for Artifact

Name \* Example: BigFix Query for Artifact

API Name \* bigfix\_query\_for\_artifact

Description Query a BigFix server for any endpoints in the BigFix environment with hits for an artifact.

Object Type \* Artifact

Input Pre-Process Script Output Post-Process Script

Language: Python Theme light Mode Default Tab Size 2 - Font + Font

```

1 inputs.bifix_artifact_id = artifact.id
2 inputs.bifix_artifact_value = artifact.value
3 inputs.bifix_artifact_type = artifact.type
4 if artifact.properties is not None:
5     inputs.bifix_artifact_properties_name = artifact.properties[0]["name"]
6     inputs.bifix_artifact_properties_value = artifact.properties[0]["value"]
7 inputs.bifix_incident_id = incident.id
8 inputs.bifix_incident_plan_status = incident.plan_status

```

A Menu Item rule called “Example: BigFix Query for Artifact” is included. This rule calls the workflow above. A user can invoke the workflow by right-clicking on this rule from the Actions drop-down menu of a suspect artifact.

**BigFix test** Actions ▾

**Summary**

ID 2095

Phase Respond

Severity Low

Date Created 09/21/2018

Date Occurr... —

Date Discov... 09/21/2018

Data Compr... Unknown

Incident Type —

**Description**

No description.

Tasks Details Breach Notes Members News Feed Attachments Stats Timeline **Artifacts**

**Artifacts** Edit

Add Artifact Table Graph

Search...

Artifact Type: All Date Created: All Has Attachment: All

Show 25 entries

Type	Value	Created	Relate?	Actions
File Path	C:\temp\testfile.txt	09/21/2018	As specified in artifact type settings	⋮
File Path	/tmp/testfile.txt	09/21/2018	As specified in artifact type settings	⋮
Registry Key	HKEY_LOCAL_MACHINE\SOFTWARE\TEST\TEST\com.tst.browsercore	09/21/2018	As specified in artifact type settings	⋮

**Related Incidents**

No related incidents.

**Attachments**

There are no attachments.

**Newsfeed**

Resilient Sysadmin added a row to the

**BigFix Query Results** Search... Print Export

Query Execution Date	Artifact Type	Artifact Value	BigFix Computer ID	BigFix Computer Name	Remediation Status	BigFix Action ID	Remediation Date
There is no data for this table							

If any endpoints are detected in the BigFix environment with the suspected artifact, entries are added to the data table “BigFix Query Results”.

**BigFix Query Results** Search... Print Export

Query Execution Date	Artifact Type	Artifact Value	BigFix Computer ID	BigFix Computer Name	Remediation Status	BigFix Action ID	Remediation Date
09-21-2018 12:14:31	File Path	/tmp/testfile.txt	12315195	bigfix.test	None	—	—



## BigFix Remediation

This function creates a BigFix action to remediate a hit found on an endpoint in the BigFix environment.

### Customization Settings

LayoutsRulesScriptsWorkflows**Functions**Message DestinationsPhases & TasksIncident Types

Functions / fn\_bigfix\_remediation

Name \*

API Name \* ⓘ

Message Destination \*

Description

BigFix Remediation

fn\_bigfix\_remediation

fn\_bigfix

Remediate a hit on an endpoint or asset in a BigFix environment.  
The supported artifact types are:

#### Inputs

bigfix\_asset\_id

bigfix\_artifact\_value

bigfix\_artifact\_type

bigfix\_incident\_id

This function takes the following parameters:

- bigfix\_asset\_id – Bigfix endpoint or asset ID
- bigfix\_artifact\_value - Resilient artifact value
- bigfix\_artifact\_type - Resilient artifact type
- bigfix\_incident\_id - Resilient incident ID

The example workflow (object type = Data Table) that calls this function is “Example: BigFix Remediate”.

Workflows / Example: BigFix Remediate

Name \* Example: BigFix Remediate

API Name \* ⓘ bigfix\_remediate

Description Remediate or fix a hit in a BigFix environment and return status of the remediating action.

Object Type \* Data Table

Data table \* BigFix Query Results

Input Parameter	Value
bigfix_asset_id * ⓘ	
bigfix_artifact_value * ⓘ	
bigfix_artifact_type * ⓘ	
bigfix_incident_id * ⓘ	

The parameter assignments are done in the Pre-Process Script tab.

Workflows / Example: BigFix Remediate

Name \* Example: BigFix Remediate

API Name \* ⓘ bigfix\_remediate

Description Remediate or fix a hit in a BigFix environment and return status of the remediating action.

Object Type \* Data Table

Data table \* BigFix Query Results

Input Pre-Process Script Output Post-Process Script

Language: Python Theme light Mode Default Tab Size 2 - Font + Font

```
1 inputs.bigfix_asset_id = row.res.bigfix_computer_id
2 inputs.bigfix_artifact_value = row.res.artifact_value
3 inputs.bigfix_artifact_type = row.res.artifact_type
4 inputs.bigfix_incident_id = incident.id
```

A Menu Item rule called “Example: BigFix Remediate” is also included. This rule calls the workflow. A user can invoke the workflow by right-clicking on this rule from the Actions drop-down or a data table entry for an endpoint with a hit.

Artifacts

Edit

Add Artifact

Table

Graph

Search...

Artifact Type: All

Date Created: All

Has Attachment: All

Show 25 entries

Type	Value	Created	Relate?	Actions
Registry Key	HKEY_LOCAL_MACHINE\SOFTWARE\TEST\TEST\com.tst.browsercore	09/21/2018	As specified in artifact type settings	
File Path	/tmp/testfile.txt	09/21/2018	As specified in artifact type settings	
File Path	C:\temp\testfile.txt	09/21/2018	As specified in artifact type settings	

BigFix Query Results

Search...

Print

Export

Artifact Type	Artifact Value	BigFix Computer ID	BigFix Computer Name	Remediation Status	BigFix Action ID	Remediation Date
File Path	/tmp/testfile.txt	12315195	bigfix.test	None	—	—

Displaying 1 - 1 of 1

Example: BigFix Remediate

Example: BigFix Retrieve Resource Details

If a remediating BigFix action is successfully created, the entry in the data table “BigFix Query Results” which the workflow was invoked against, is updated with the status, remediation date and action ID.

BigFix Query Results

Search...

Print

Export

Query Execution Date	Artifact Type	Artifact Value	BigFix Computer ID	BigFix Computer Name	Remediation Status	BigFix Action ID	Remediation Date
09-21-2018 12:14:31	File Path	/tmp/testfile.txt	12315195	bigfix.test	BigFix action created successfully.	268	09-21-2018 12:17:42

Displaying 1 - 1 of 1

## BigFix Action Status

### Customization Settings

LayoutsRulesScriptsWorkflows**Functions**Message DestinationsPhases & TasksIncident Types

Functions / fn\_bigfix\_action\_status

Name \*

BigFix Action Status

API Name \* ⓘ

fn\_bigfix\_action\_status

Message Destination \*

fn\_bigfix

Description

Retrieve status of a BigFix action.

#### Inputs

bigfix\_action\_id

This function takes the following parameter:

- bigfix\_action\_id – Bigfix action ID
- The example workflow (object type = Data Table) that calls this function is “Example: BigFix Update Action status”.

Workflows / Example: BigFix Update Action status

Name \*

Example: BigFix Update Action status

API Name \* ⓘ

bigfix\_update\_action\_status

Description

Update status of a BigFix action which mediates a hit.

Object Type \*

Data Table

Data table \*

BigFix Query Results

InputPre-Process ScriptOutputPost-Process Script

Input Parameter

Value

bigfix\_action\_id \* ⓘ

The parameter assignment is done in the Pre-Process Script tab.

### Customization Settings

LayoutsRulesScripts**Workflows**FunctionsMessage DestinationsPhases & TasksIncident Types

Workflows / Example: BigFix Update Action status

Name \*Example: BigFix Update Action status

API Name \* ⓘbigfix\_update\_action\_status

DescriptionUpdate status of a BigFix action which mediates a hit.

Object Type \*Data Table

Data table \*BigFix Query Results

InputPre-Process ScriptOutputPost-Process Script

language: PythonTheme: lightMode: DefaultTab Size: 2- Font+ Font

1 inputs.bigfix\_action\_id = row.res\_bigfix\_action\_id

A Menu Item rule called “Example: BigFix Update Action status” is also included. This rule calls the workflow. A user can invoke the workflow by right-clicking on this rule from the Actions drop-down of a data table entry for an endpoint with a hit and where an action ID has been set.

Artifacts

Edit

Add Artifact

Table

Graph

Search...

Artifact Type: All

Date Created: All

Has Attachment: All

Show 25 entries

Type	Value	Created	Relate?	Actions
File Path	C:\temp\testfile.txt	09/21/2018	As specified in artifact type settings	...
File Path	/tmp/testfile.txt	09/21/2018	As specified in artifact type settings	...
Registry Key	HKEY_LOCAL_MACHINE\SOFTWARE\TEST\TEST\com.tst.browsercore	09/21/2018	As specified in artifact type settings	...

BigFix Query Results

Search...

Print

Export

Artifact Type	Artifact Value	BigFix Computer ID	BigFix Computer Name	Remediation Status	BigFix Action ID	Remediation Date	
File Path	/tmp/testfile.txt	12315195	bigfix.test	BigFix action created successfully.	269	09-21-2018 13:34:53	...

Example: BigFix Remediate

Example: BigFix Retrieve Resource Details

Example: BigFix Update Action status

Displaying 1 - 1 of 1

If a remediating BigFix action was executed successfully, the entry in the data table “BigFix Query Results” which the workflow was invoked against, is updated with the new status.

BigFix Query Results

Search...

Print

Export

Query Execution Date	Artifact Type	Artifact Value	BigFix Computer ID	BigFix Computer Name	Remediation Status	BigFix Action ID	Remediation Date
09-21-2018 13:34:40	File Path	/tmp/testfile.txt	12315195	bigfix.test	The action executed successfully.	269	09-21-2018 13:34:53

Displaying 1 - 1 of 1

This function is also included in the “Example: BigFix Remediate” workflow and it is invoked automatically as part of that workflow. This would be the more common method of invocation.

Workflows / Example: BigFix Remediate

Name \* Example: BigFix Remediate

API Name \* ⓘ bigfix\_remediate

Description Remediate or fix a hit in a BigFix environment and return status of the remediating action.

Object Type \* Data Table

Data table \* BigFix Query Results

The workflow diagram shows a sequence of steps. It begins with a start node (a circle) labeled "Start your workflow here". An arrow points from this start node to a function node labeled "BigFix Remediation". This function node has an orange circle with a white 'f(x)' symbol above it. An arrow then points from the "BigFix Remediation" node to another function node labeled "BigFix Action Status". This second function node also has an orange circle with a white 'f(x)' symbol above it. To the left of the workflow diagram is a vertical toolbar containing various icons for workflow design, including a hand, a plus sign, a lightning bolt, a circle, a diamond, a square, a person, a document, a dollar sign, a gear, and a function icon 'f(x)'.

In cases where the “Example: BigFix Remediate” workflow does not receive the status within the specified time, this workflow can be invoked manually at a later time.

## BigFix Assets

This function performs a query to fetch BigFix properties of an endpoint with a hit from a BigFix environment.

### Customization Settings

LayoutsRulesScriptsWorkflows**Functions**Message DestinationsPhases & TasksIncident Types

Functions / fn\_bigfix\_assets

Name \*

API Name \* ⓘ

Message Destination \*

Description

BigFix Assets

fn\_bigfix\_assets

fn\_bigfix

Query a BigFix server for properties of an endpoint (asset) .

#### Inputs

bigfix\_asset\_id

bigfix\_asset\_name

bigfix\_incident\_id

This function takes the following parameter:

- bigfix\_asset\_id – Bigfix endpoint or asset ID
- bigfix\_asset\_name - Bigfix endpoint or asset name
- bigfix\_incident\_id - Resilient incident ID



The example workflow (object type = Data Table) that calls this function is “Example: BigFix Retrieve Resource Details”.

Customization Settings

Layouts Rules Scripts **Workflows** Functions Message Destinations Phases & Tasks Incident Types

Workflows / Example: BigFix Retrieve Resource Details

Name \* Example: BigFix Retrieve Resource Details

API Name \* ⓘ bigfix\_retrieve\_resource\_details

Description Retrieve properties of an endpoint in a BigFix environment.

Object Type \* Data Table

Data table \* BigFix Query Results

Input Pre-Process Script Output Post-Process Script

Input Parameter	Value
bigfix_asset_id * ⓘ	
bigfix_asset_name * ⓘ	
bigfix_incident_id * ⓘ	

The parameter assignments are done in the Pre-Process Script tab.

Customization Settings

Layouts Rules Scripts **Workflows** Functions Message Destinations Phases & Tasks Incident Types

Workflows / Example: BigFix Retrieve Resource Details

Name \* Example: BigFix Retrieve Resource Details

API Name \* ⓘ bigfix\_retrieve\_resource\_details

Description Retrieve properties of an endpoint in a BigFix environment.

Object Type \* Data Table

Data table \* BigFix Query Results

Input **Pre-Process Script** Output Post-Process Script

Language: Python Theme light Mode Default Tab Size 2 - Font + Font

```
1 inputs.bifix_asset_name = row.res_bifix_computer_name
2 inputs.bifix_asset_id = row.res_bifix_computer_id
3 inputs.bifix_incident_id = incident.id
```

A Menu Item rule called “Example: BigFix Retrieve Resource Details” is also included. This rule calls the workflow. A user can invoke the workflow by right-clicking on this rule from the Actions drop-down of a data table entry for an endpoint with a hit.

Artifacts

Edit

Add Artifact

Table

Graph

Search...

Artifact Type: All

Date Created: All

Has Attachment: All

Show 25 entries

Type	Value	Created	Relate?	Actions
File Path	C:\temp\testfile.txt	09/21/2018	As specified in artifact type settings	
File Path	/tmp/testfile.txt	09/21/2018	As specified in artifact type settings	
Registry Key	HKEY_LOCAL_MACHINE\SOFTWARE\TEST\TEST\com.tst.browsercore	09/21/2018	As specified in artifact type settings	

BigFix Query Results

Search...

Print

Export

Artifact Type	Artifact Value	BigFix Computer ID	BigFix Computer Name	Remediation Status	BigFix Action ID	Remediation Date	
File Path	/tmp/testfile.txt	12315195	bigfix.test	The action ex	269	09-21-2018 1	...

Displaying 1 - 1 of 1

Example: BigFix Remediate

Example: BigFix Retrieve Resource Details

Example: BigFix Update Action status

An attachment is added to the incident containing BigFix properties of the targeted endpoint.

Attachments

Drag file here

Upload File

Maximum file size: 25 MB

Search...

Show Task Attachments

Uploaded By: All

Date Created: All

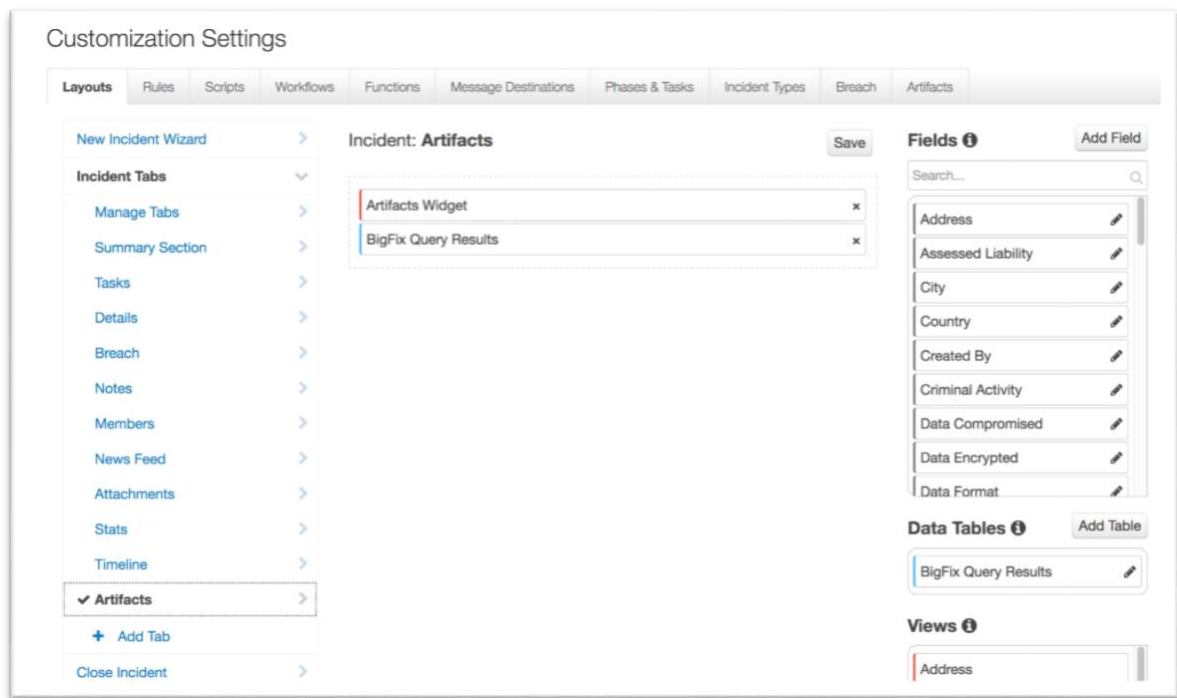
Type	Name	Uploaded By	Date Added	Size	Actions
	bigfix-properties-bigfix.test-20180921.xml	Resilient Sysadmin	09/21/2018	37 KB	

Showing 1 to 1 of 1 entries

## Resilient Platform Configuration

To display query results, users need to manually add the “BigFix Query Results” data table to the Artifacts tab.

1. Navigate to the Customization Settings and select the Layouts tab.
2. Select Artifacts.
3. Drag the “BigFix Query Results” data table to your Artifacts tab.
4. Click Save.



## Troubleshooting

There are several ways to verify the successful operation of a function.

- Resilient Action Status

When viewing an incident, use the Actions menu to view Action Status. By default, pending and errors are displayed. Modify the filter for actions to also show Completed actions. Clicking on an action displays additional information on the progress made or what error occurred.

- Resilient Scripting Log

A separate log file is available to review scripting errors. This is useful when issues occur in the pre-processing or post-processing scripts. The default location for this log file is:  
`/var/log/resilient-scripting/resilient-scripting.log`

- Resilient Logs

By default, Resilient logs are retained at `/usr/share/co3/logs`. The `client.log` may contain additional information regarding the execution of functions.

- Resilient-Circuits

The log is controlled in the `.resilient/app.config` file under the section `[resilient]` and the property `logdir`. The default file name is `app.log`. Each function will create progress information. Failures will show up as errors and may contain python trace statements.

## Support

For additional support, contact [support@resilientystems.com](mailto:support@resilientystems.com).

Including relevant information from the log files will help us resolve your issue.