



# Resilient SOAR Platform Integrations

# BigFix Function V1.1.2

Release Date: August 2020

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 V1.1.2 release of the Resilient BigFix Function introduces the following new features and enhancements:

* Added support added for App Host.
* Added proxy support.
* Added selftest functionality.

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 | * Query only. Remediation option not supported. * MS Windows and Linux. * Queries for IP addresses making connections to endpoints in the BigFix environment. |
| Process Name | BigFix Artifact  BigFix Remediation | Example: BigFix Query for Artifact  Example: BigFix Remediate | * MS Windows and Linux. * Case insensitive for MS Windows. * Case sensitive for Linux. |
| Service | BigFix Artifact  BigFix Remediation | Example: BigFix Query for Artifact  Example: BigFix Remediate | * Currently MS Windows only. * Query on ‘Service name ‘or ‘Display name’. * Case insensitive. |
| File path | BigFix Artifact  BigFix Remediation | Example: BigFix Query for Artifact  Example: BigFix Remediate | * MS Windows and Linux. |
| Registry Key | BigFix Artifact  BigFix Remediation | Example: BigFix Query for Artifact  Example: BigFix Remediate | * MS Windows only. * Search for key, key + value + no data or key + value + data. * Delete at key level. * Search for values of type string ONLY. * Remediation of keys at root level and keys with subkeys is disallowed This is a safety measure. |

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)](https://github.com/ibmresilient/resilient-reference/blob/master/developer_guides/Integration%20Server%20Guide.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 while polling BigFix to get query results or the final status of the remediation actions. Default is 30 (Value should be less than the bigfix\_polling\_timeout value).

bigfix\_polling\_timeout. Time in seconds that the integration waits before timing out while polling BigFix to get an initial query result or to get the final status of remediation actions. Default is 600

bigfix\_endpoints\_wait. Time in seconds to wait for all endpoints to respond once an initial query result has been received. Default is 30

bigfix\_hunt\_results\_limit. Limits the number of results sent to the Resilient platform. Default is 200.

# Settings for access to BigFix via a proxy

#http\_proxy=http':'http://proxy:80

#https\_proxy=https':'http://proxy:80

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

1. 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

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

sudo su - integration

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

sudo rm ~/.resilient/resilient\_bigfix\_integration.db

1. 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

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

Testing the integration

Run selftest to test the integration you configured

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

resilient-circuits selftest -l fn-bigfix

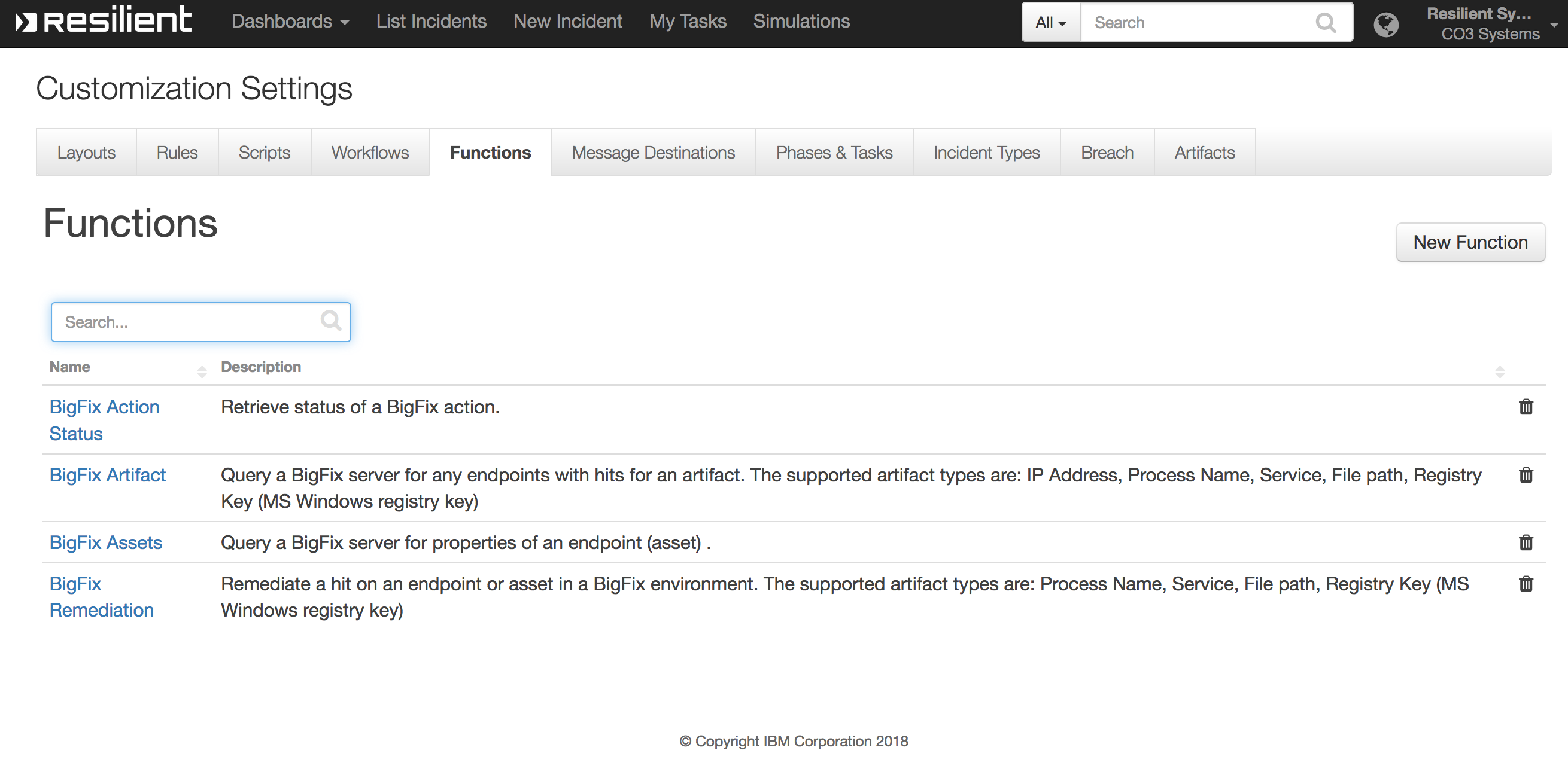
1. The resulting output will show a result indicating success or failure.

fn-bigfix:

selftest: success, Elapsed time: 0.680000 seconds

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.

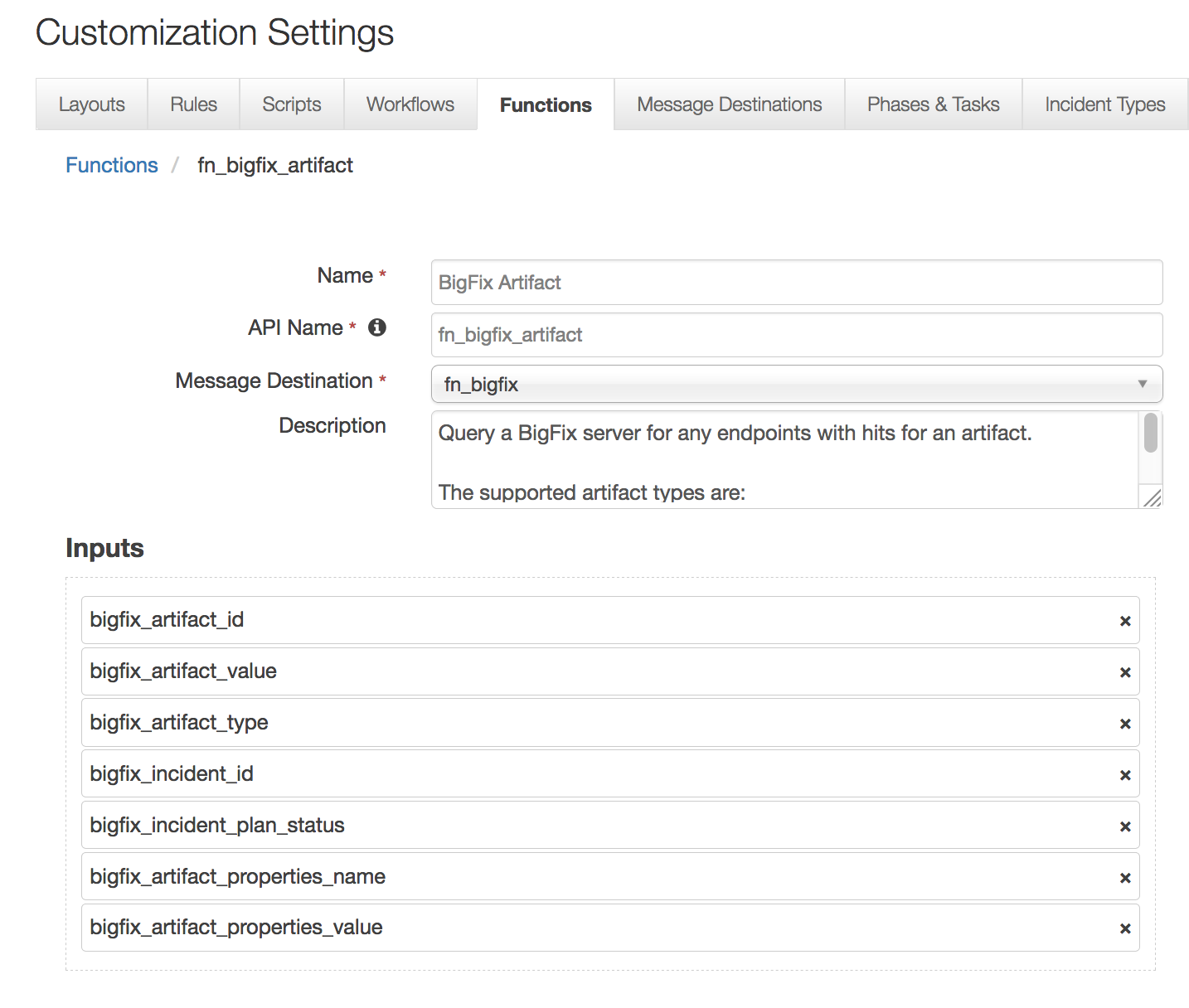


Customizations

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.

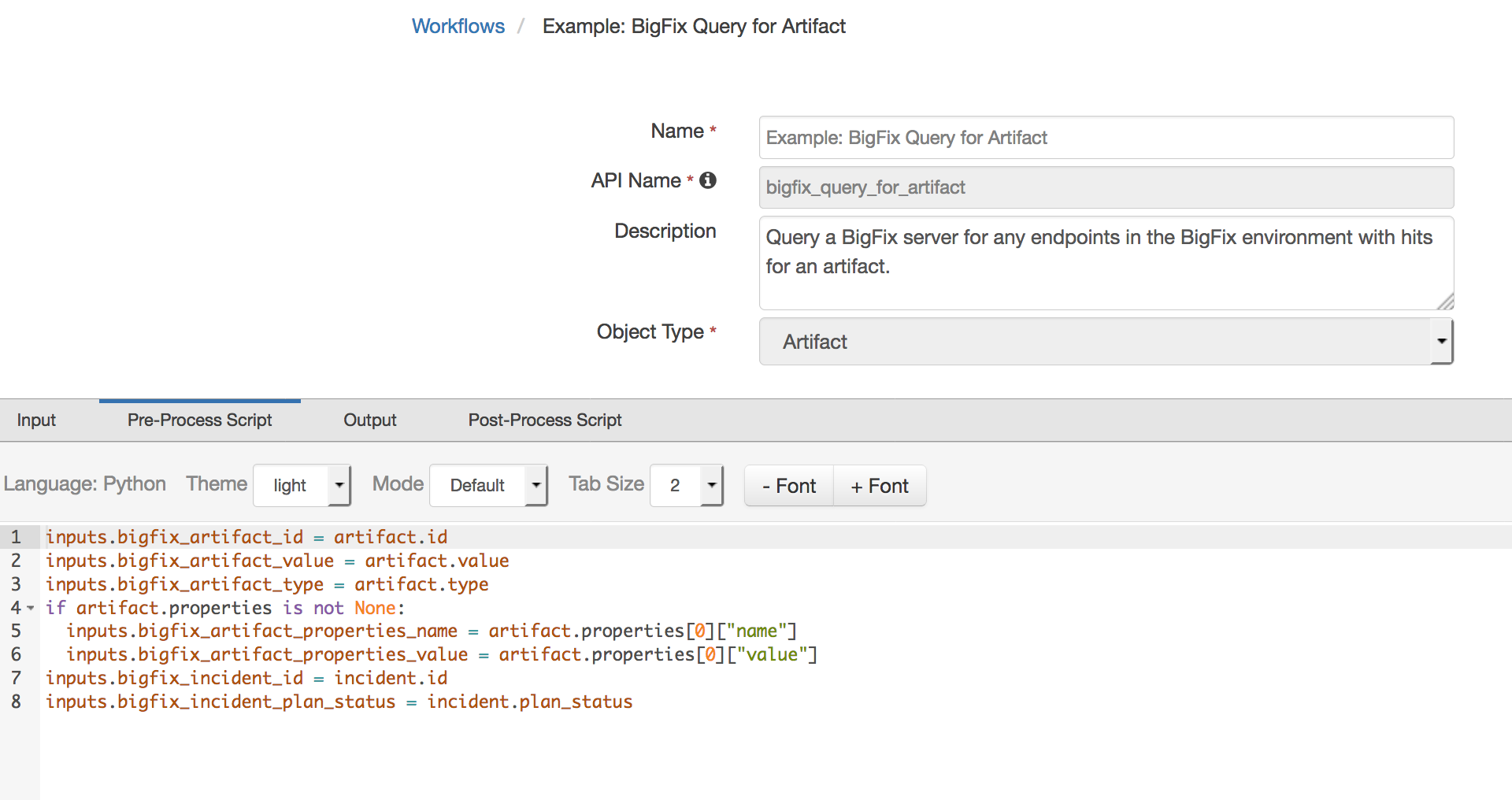


This function takes the following parameters:

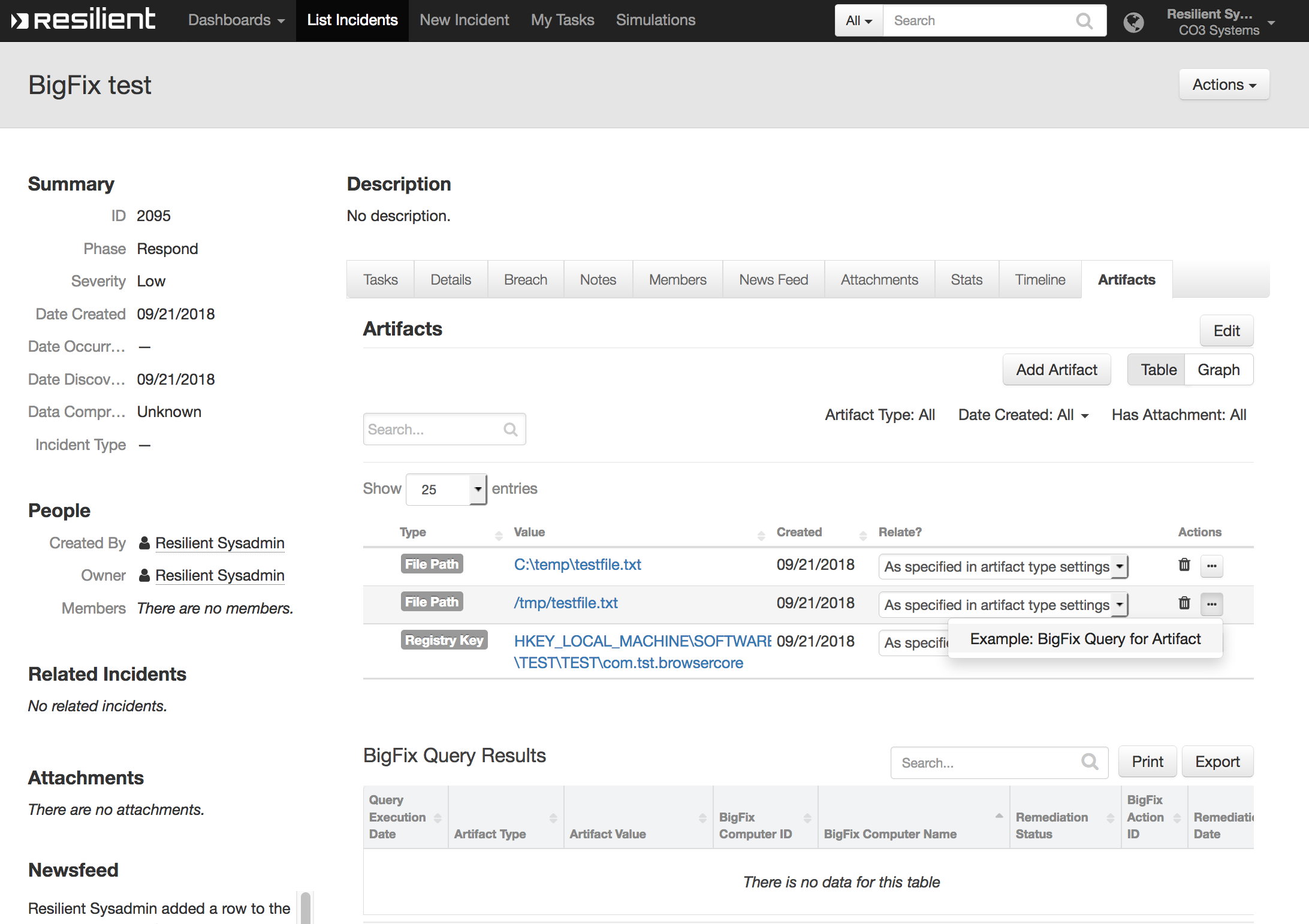
* bigfix\_artifact\_id - Resilient artifact ID
* bigfix\_artifact\_value - Resilient artifact value
* bigfix\_artifact\_type - Resilient artifact type
* bigfix\_incident\_id - Resilient incident ID
* bigfix\_incident\_plan\_status - Resilient incident status
* bigfix\_artifact\_properties\_name - Resilient 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”.

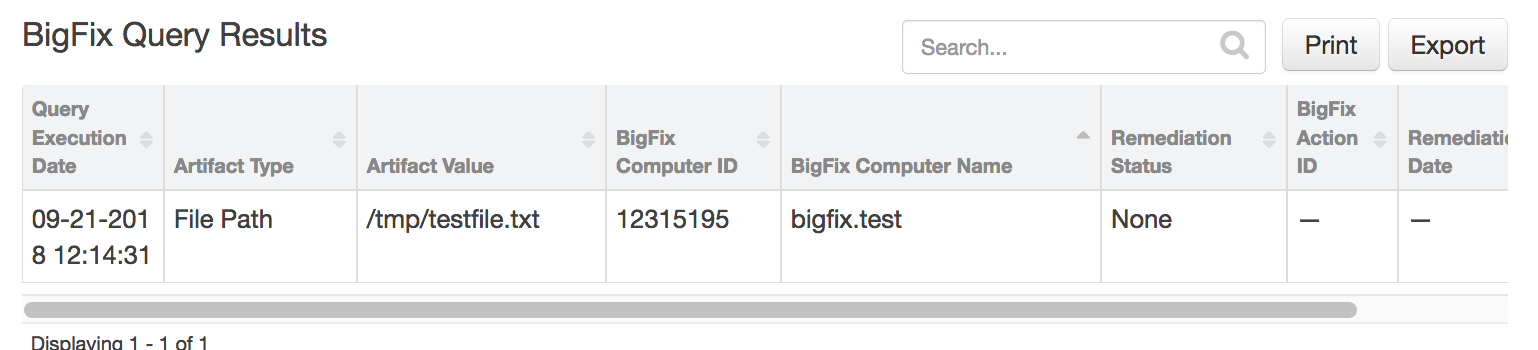
The parameter assignments are done in the Pre-Process Script tab.



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.

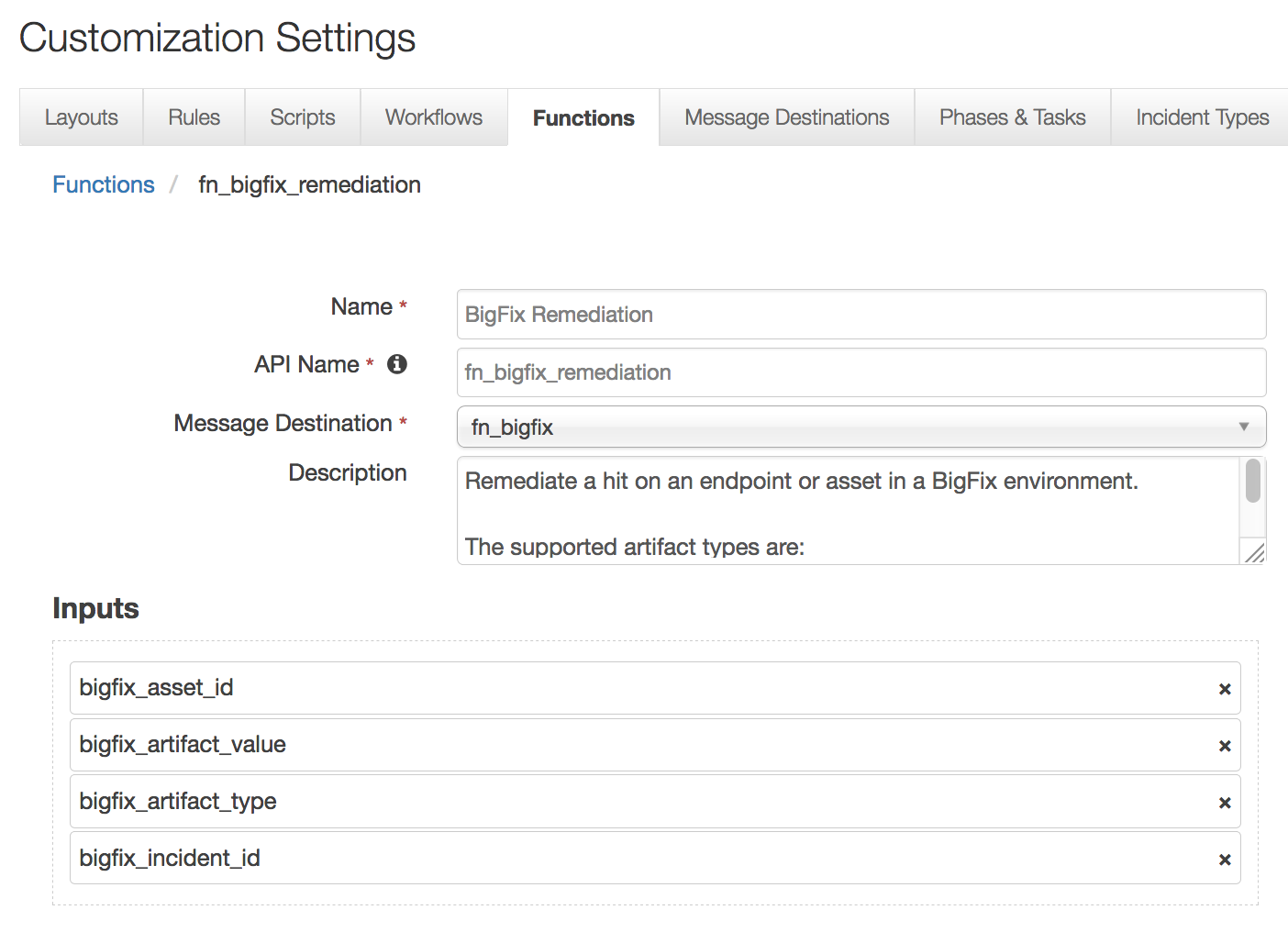


If any endpoints are detected in the BigFix environment with the suspected artifact, entries are added to the data table “BigFix Query Results”.



BigFix Remediation

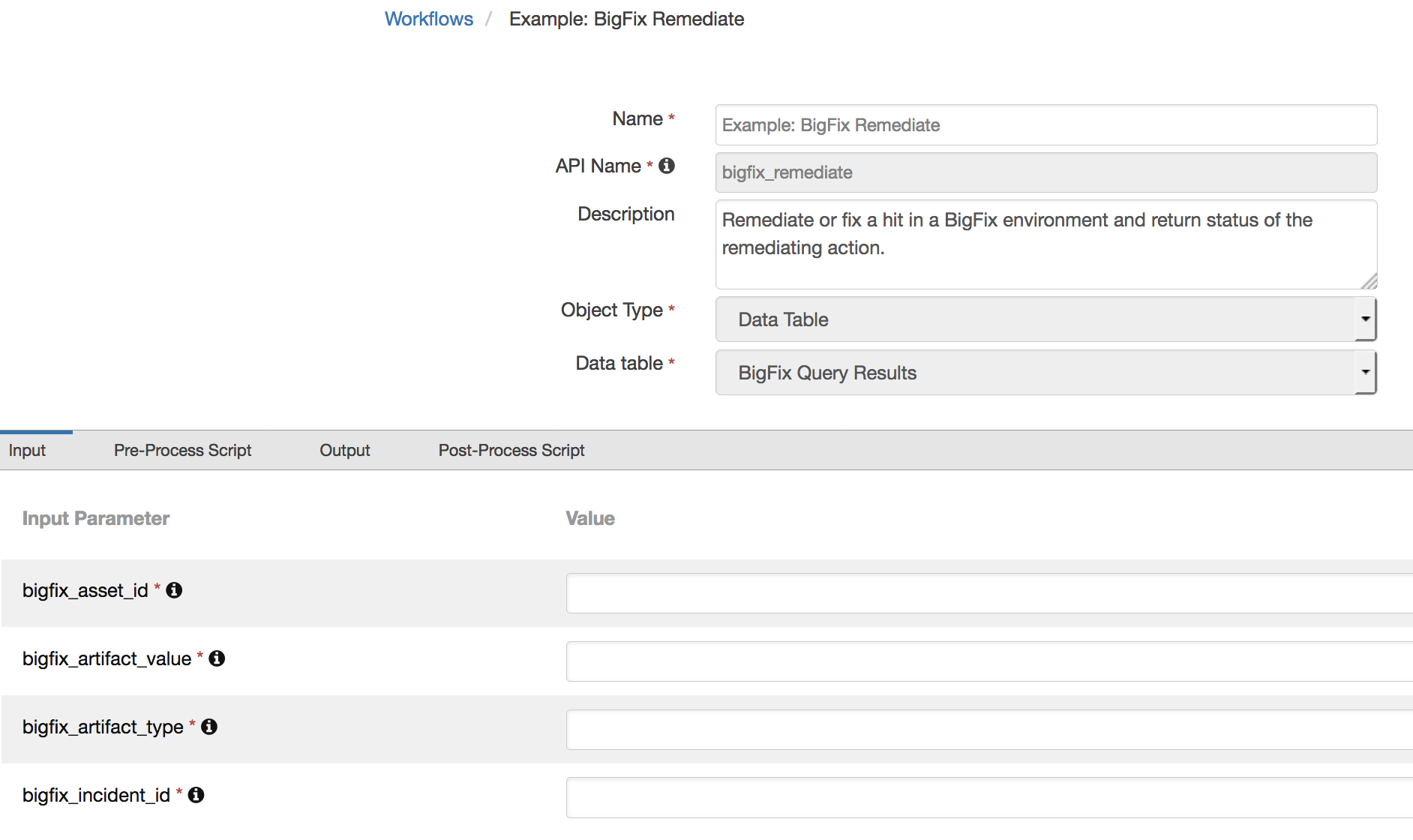
This function creates a BigFix action to remediate a hit found on an endpoint in the BigFix environment.



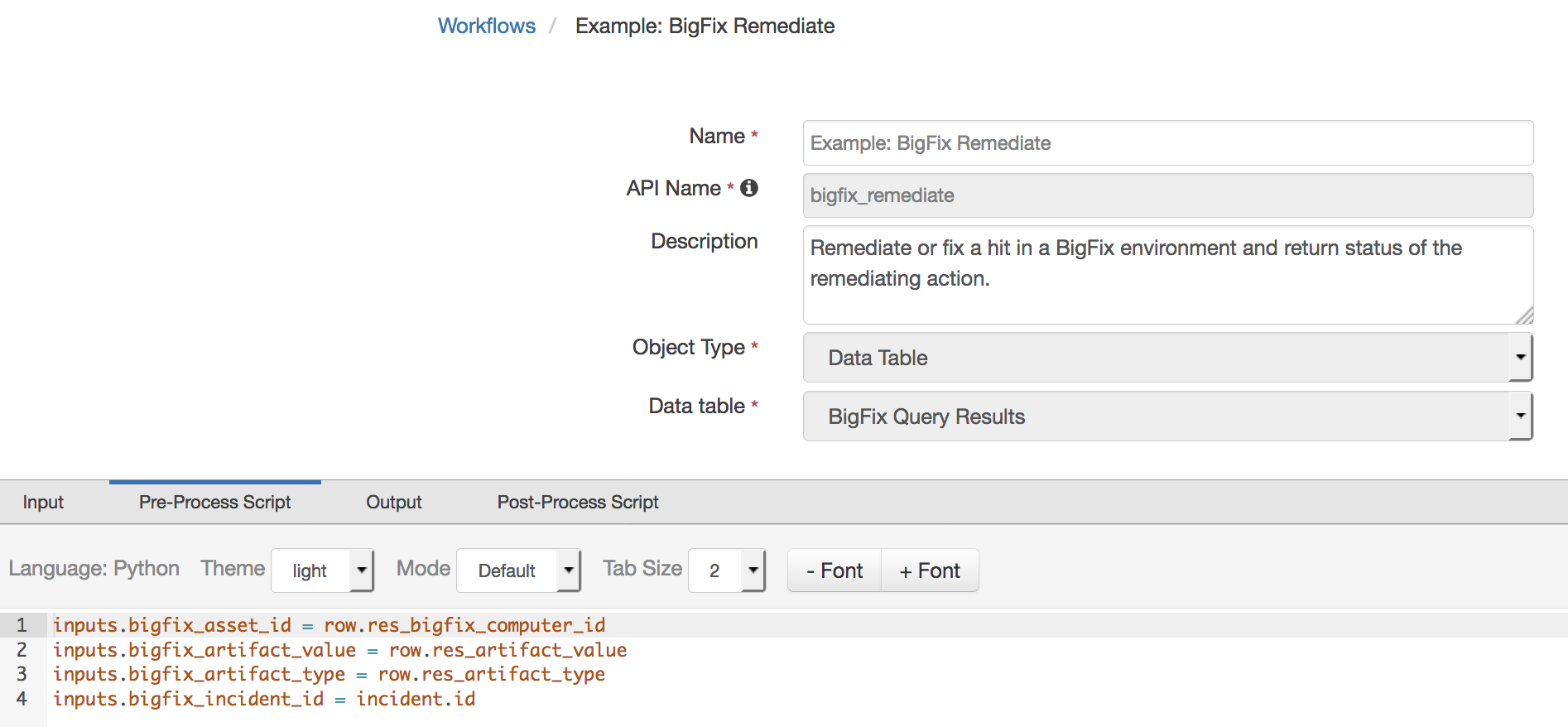
This function takes the following parameters:

* 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

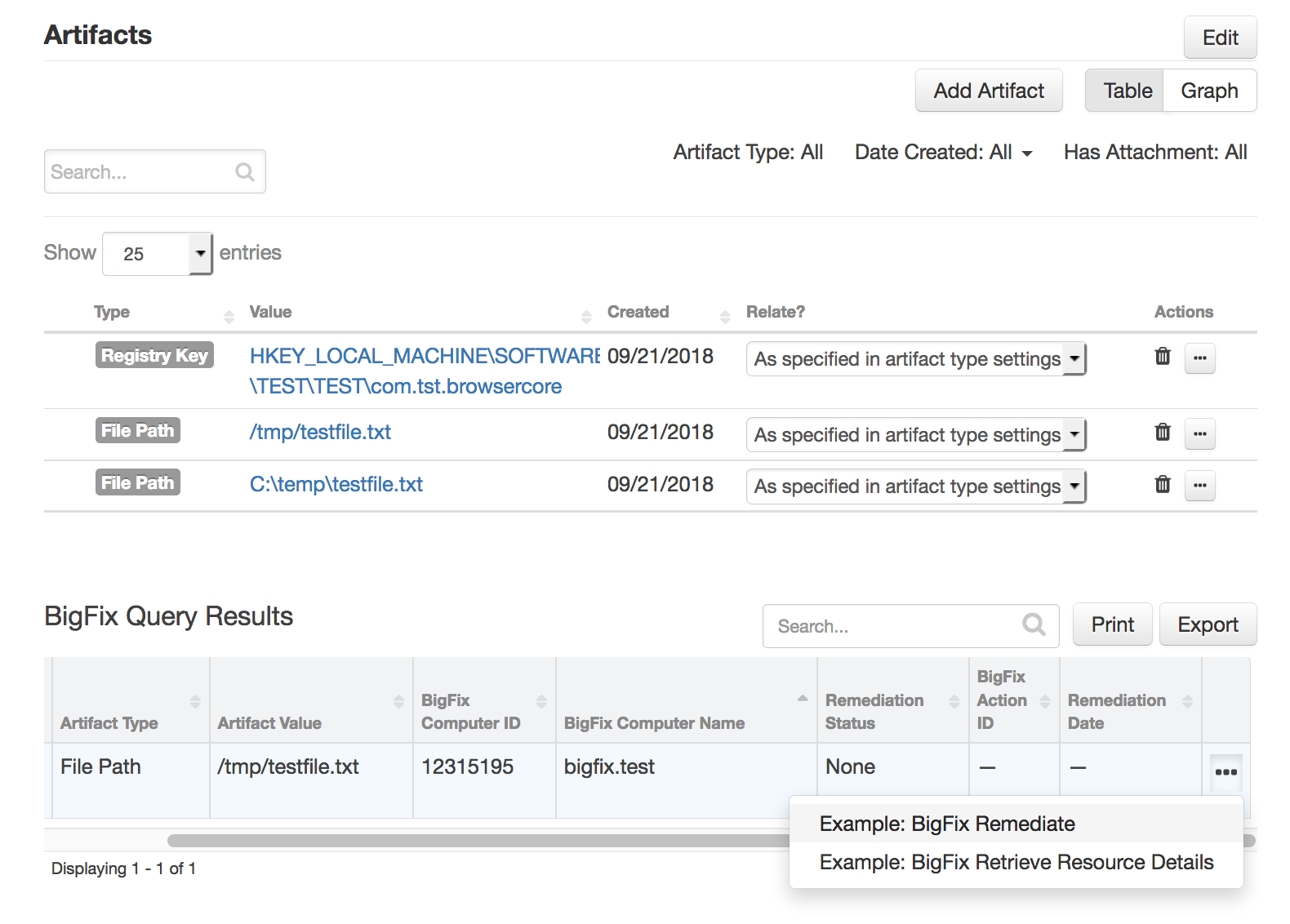
The example workflow (object type = Data Table) that calls this function is “Example: BigFix Remediate”.



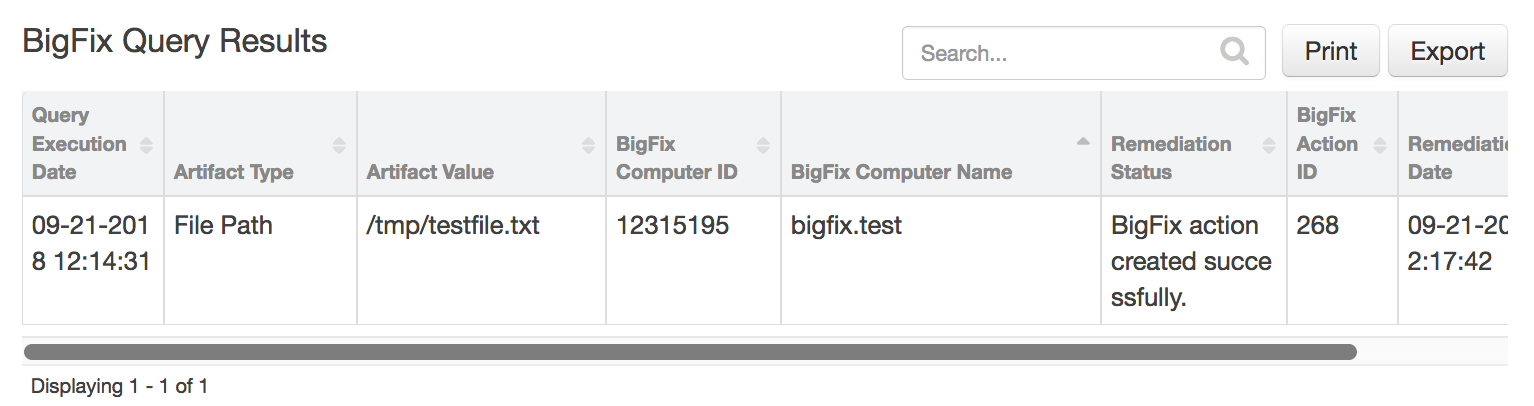
The parameter assignments are done in the Pre-Process Script tab.



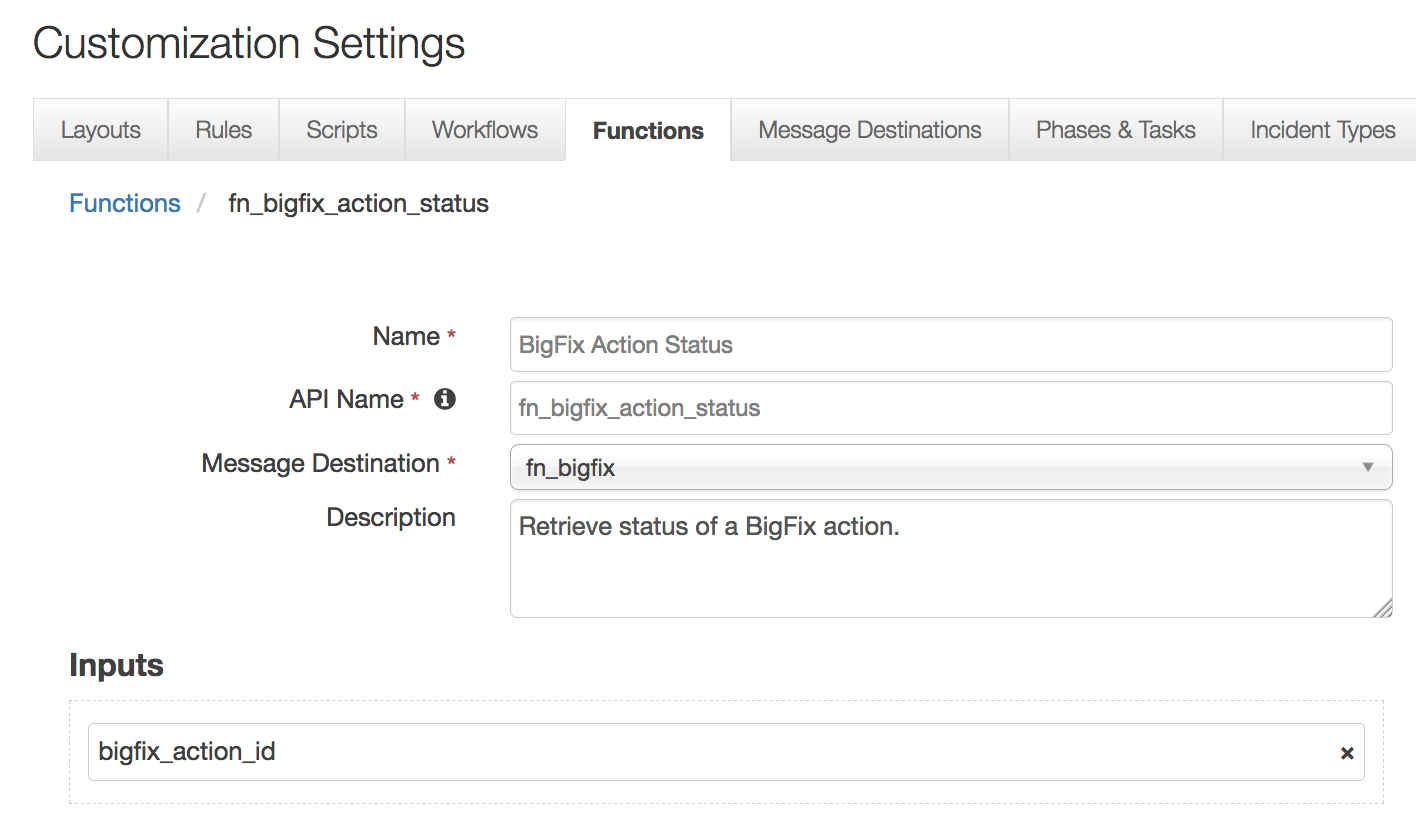
A Menu Item rule called “Example: BigFix Remediate” is also included. This rule calls the workflow. A user can invoke the workflow by right-clicking on this rule from the Actions drop-down or a data table entry for an endpoint with a hit.



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.

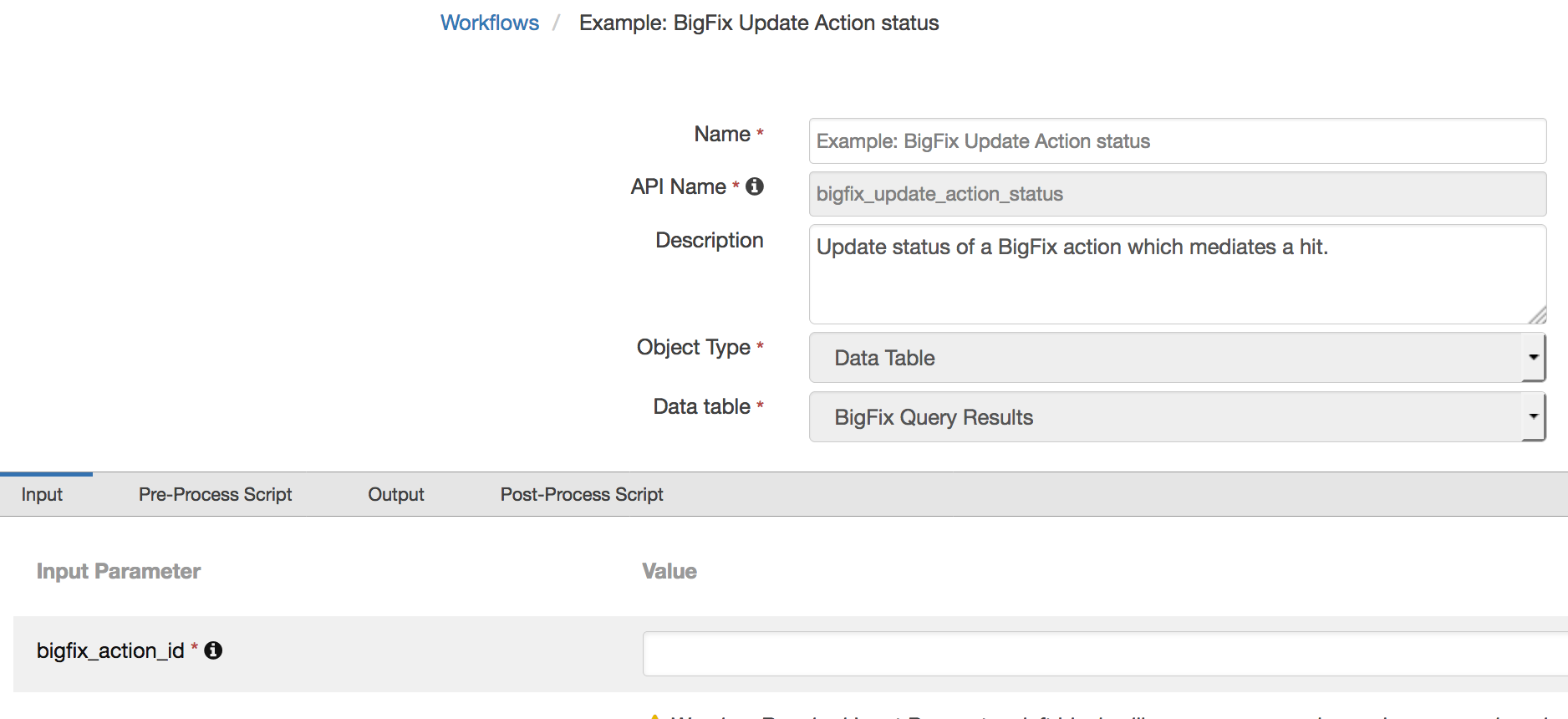


BigFix Action Status

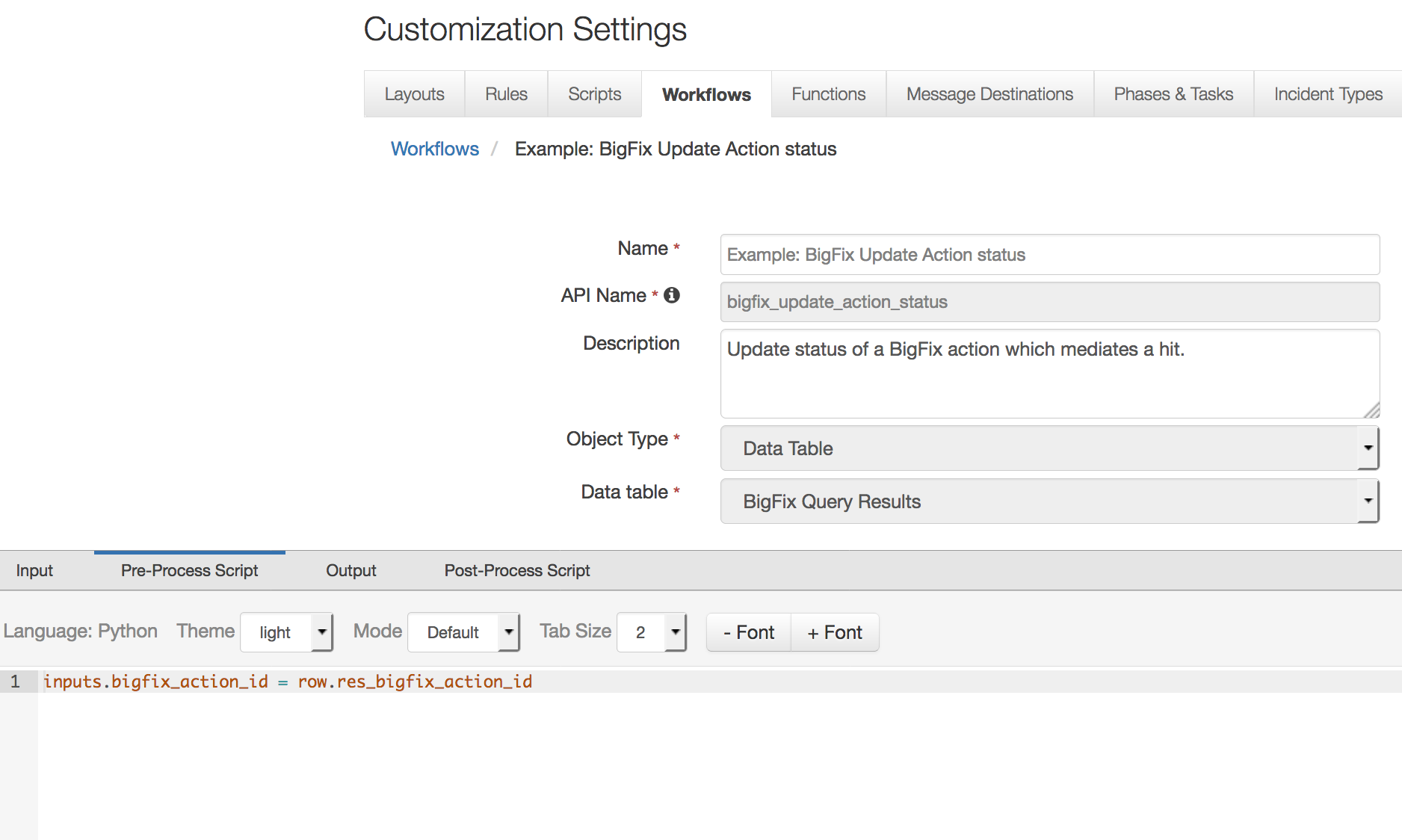


This function takes the following parameter:

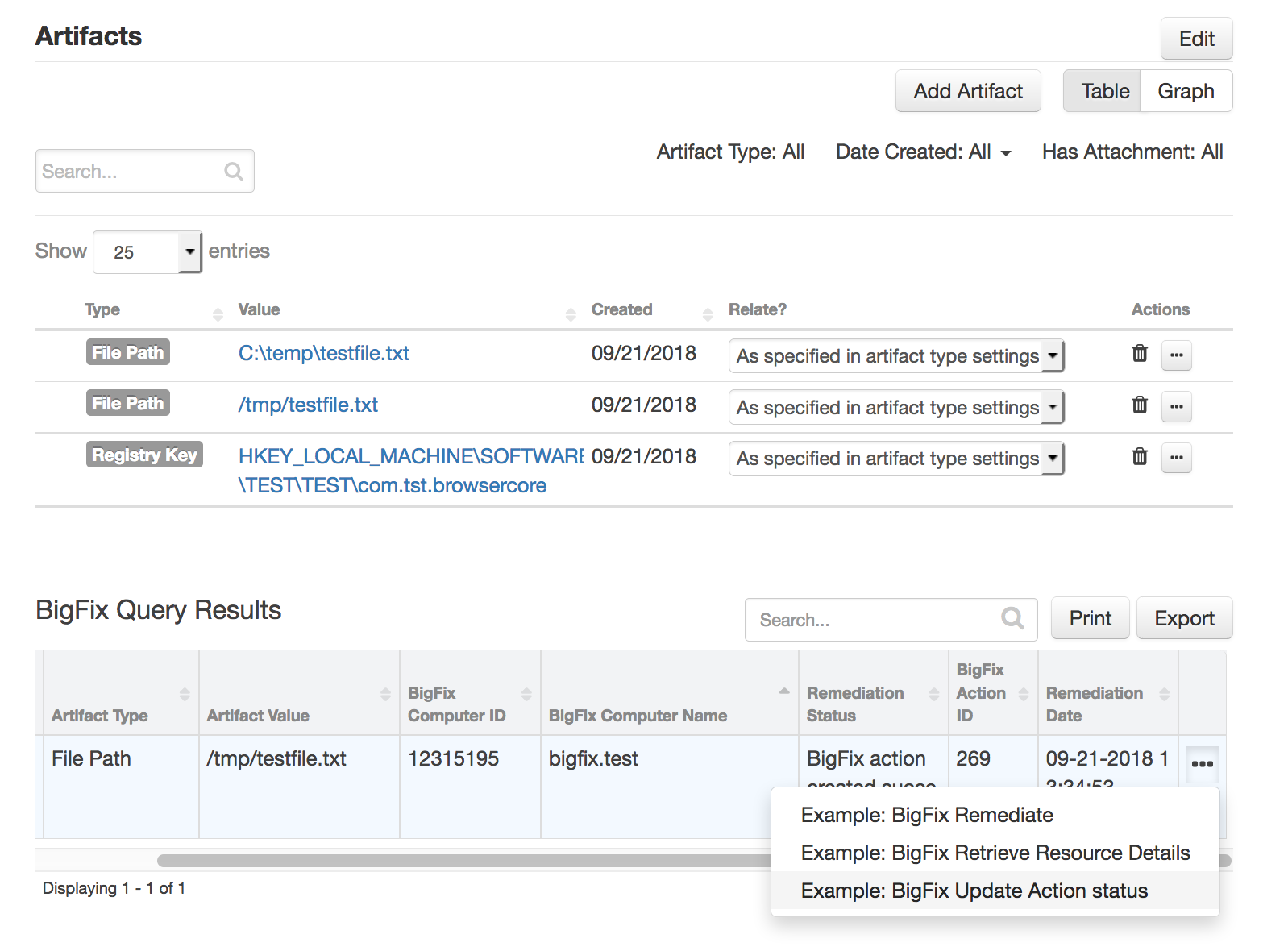
* bigfix\_action\_id – Bigfix action ID
* The example workflow (object type = Data Table) that calls this function is “Example: BigFix Update Action status”.



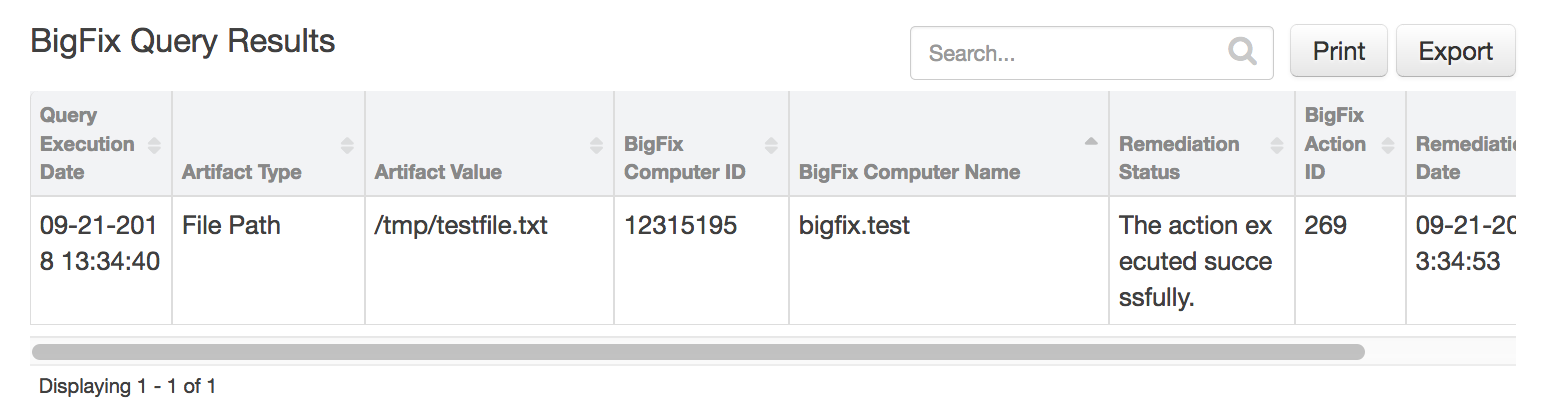
The parameter assignment is done in the Pre-Process Script tab.



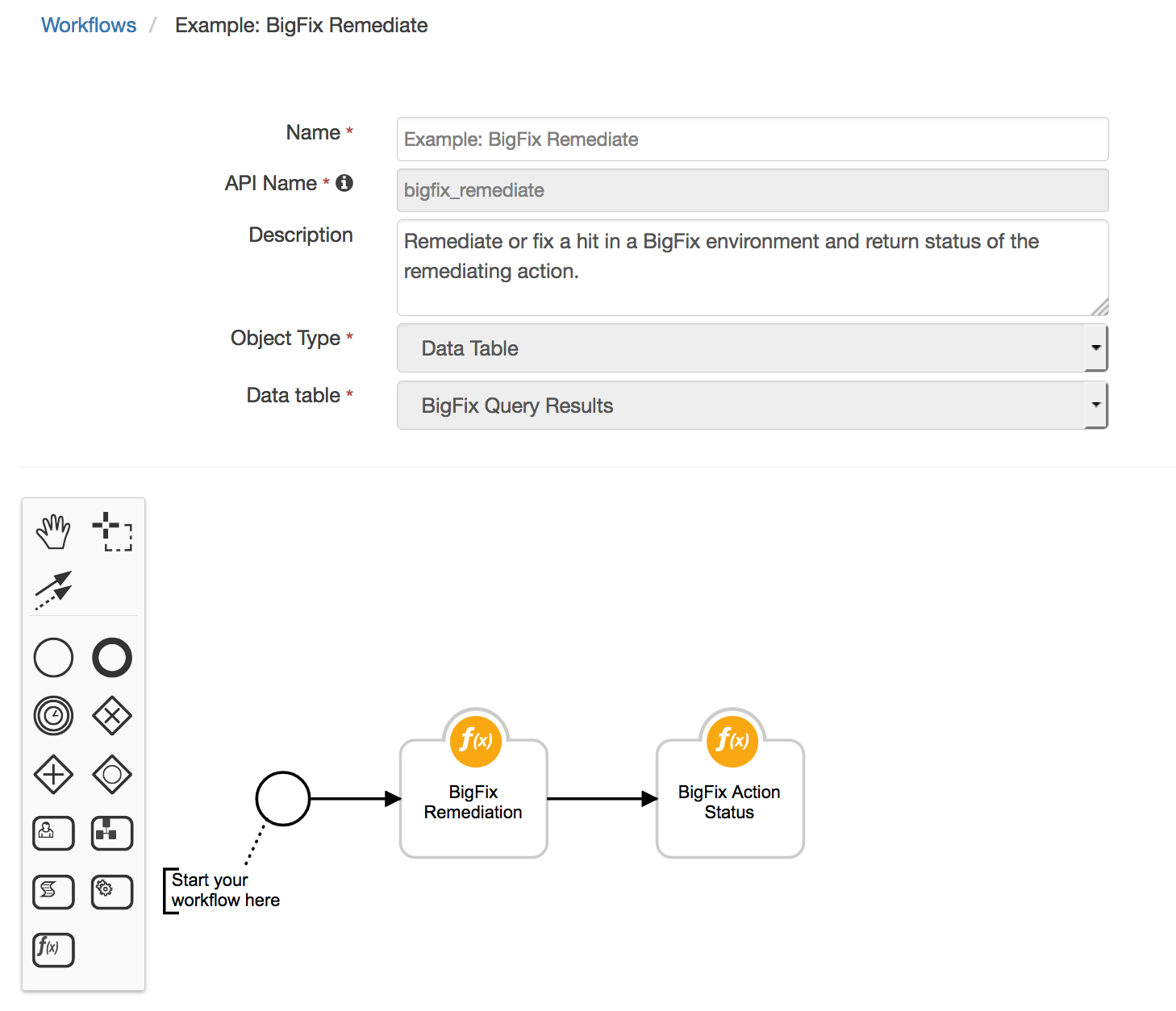
A Menu Item rule called “Example: BigFix Update Action status” is also included. This rule calls the workflow. A user can invoke the workflow by right-clicking on this rule from the Actions drop-down of a data table entry for an endpoint with a hit and where an action ID has been set.



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.



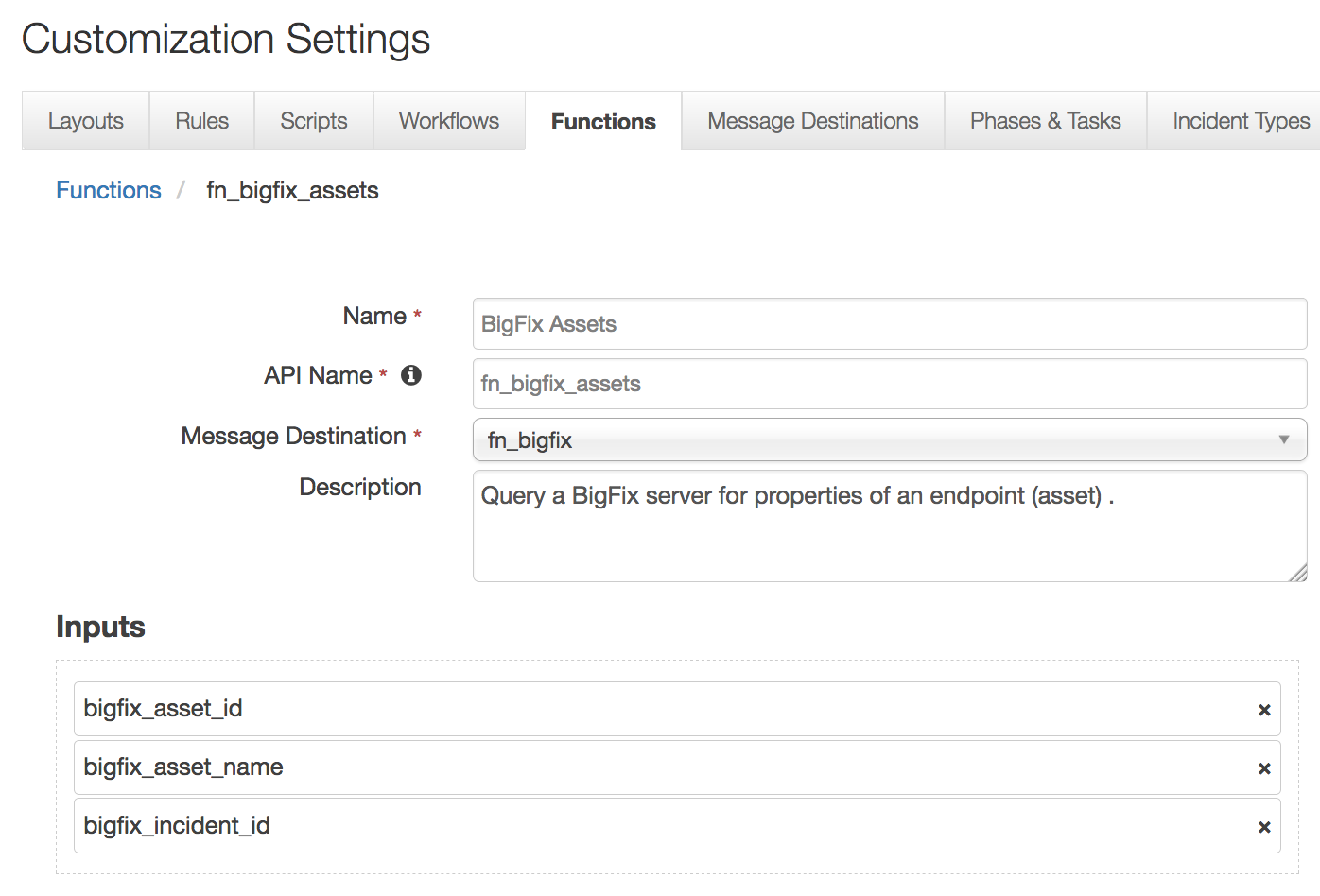
This function is also included in the “Example: BigFix Remediate” workflow and it is invoked automatically as part of that workflow. This would be the more common method of invocation.



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

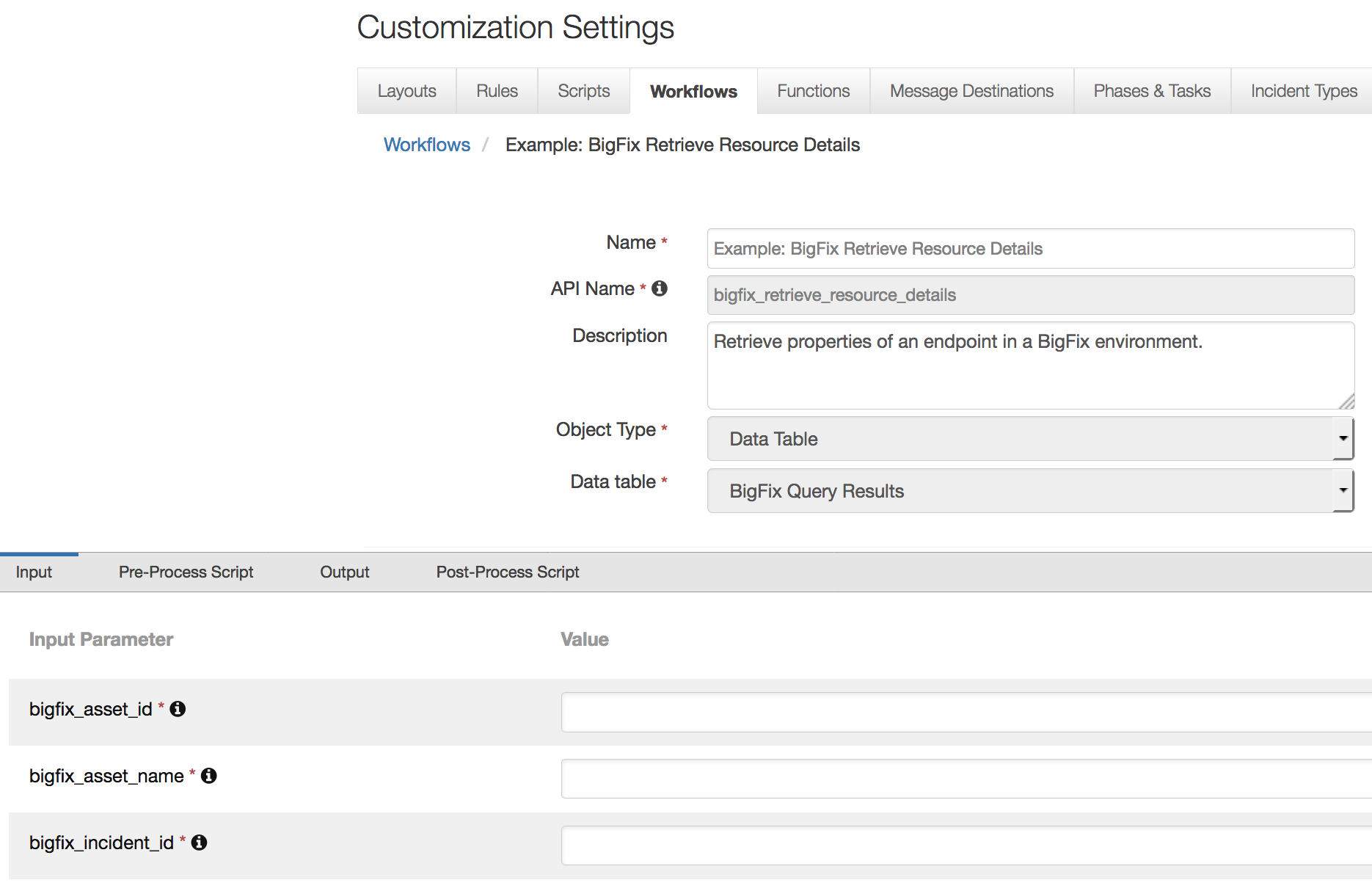
This function performs a query to fetch BigFix properties of an endpoint with a hit from a BigFix environment.



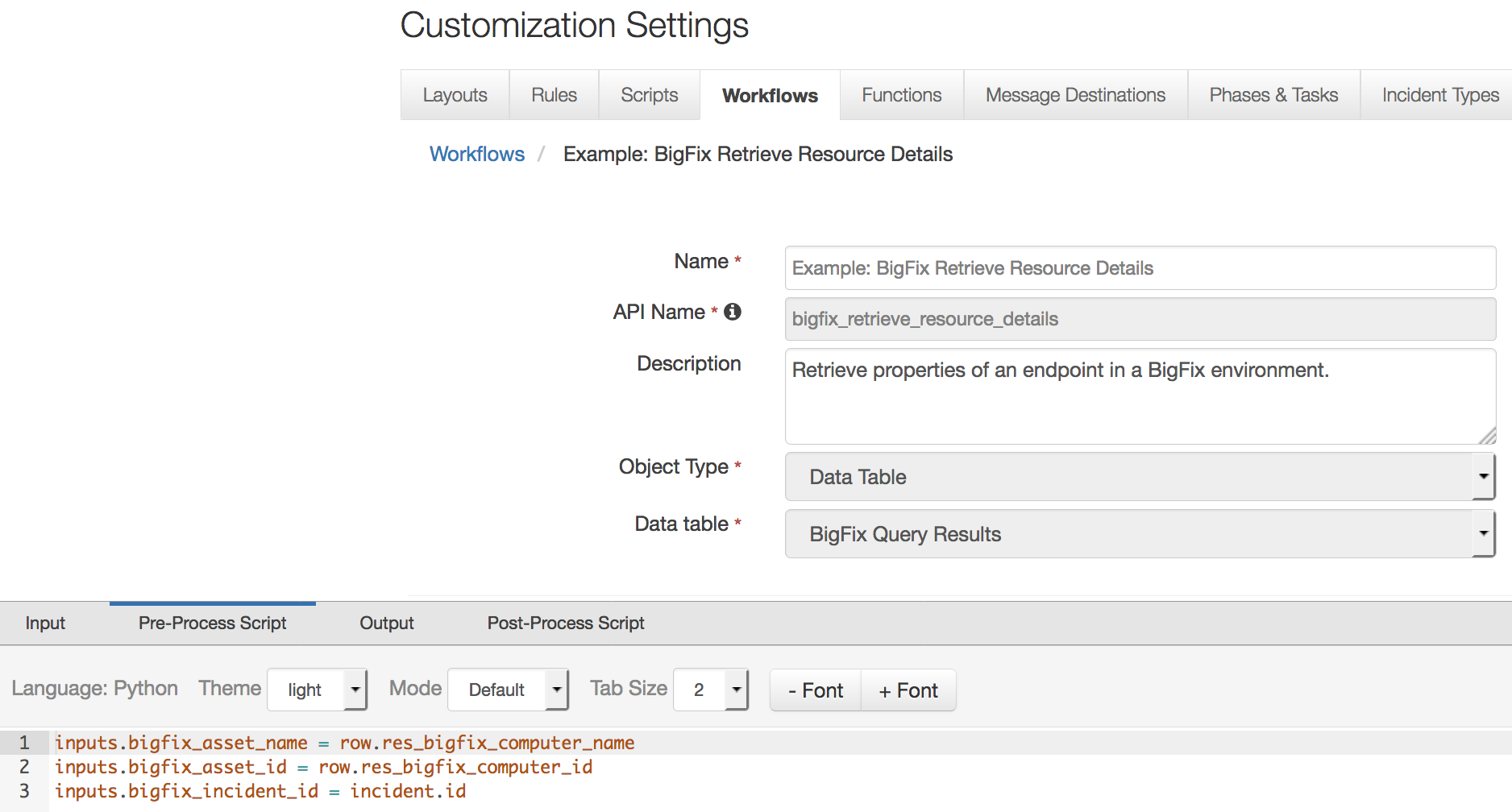
This function takes the following parameter:

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

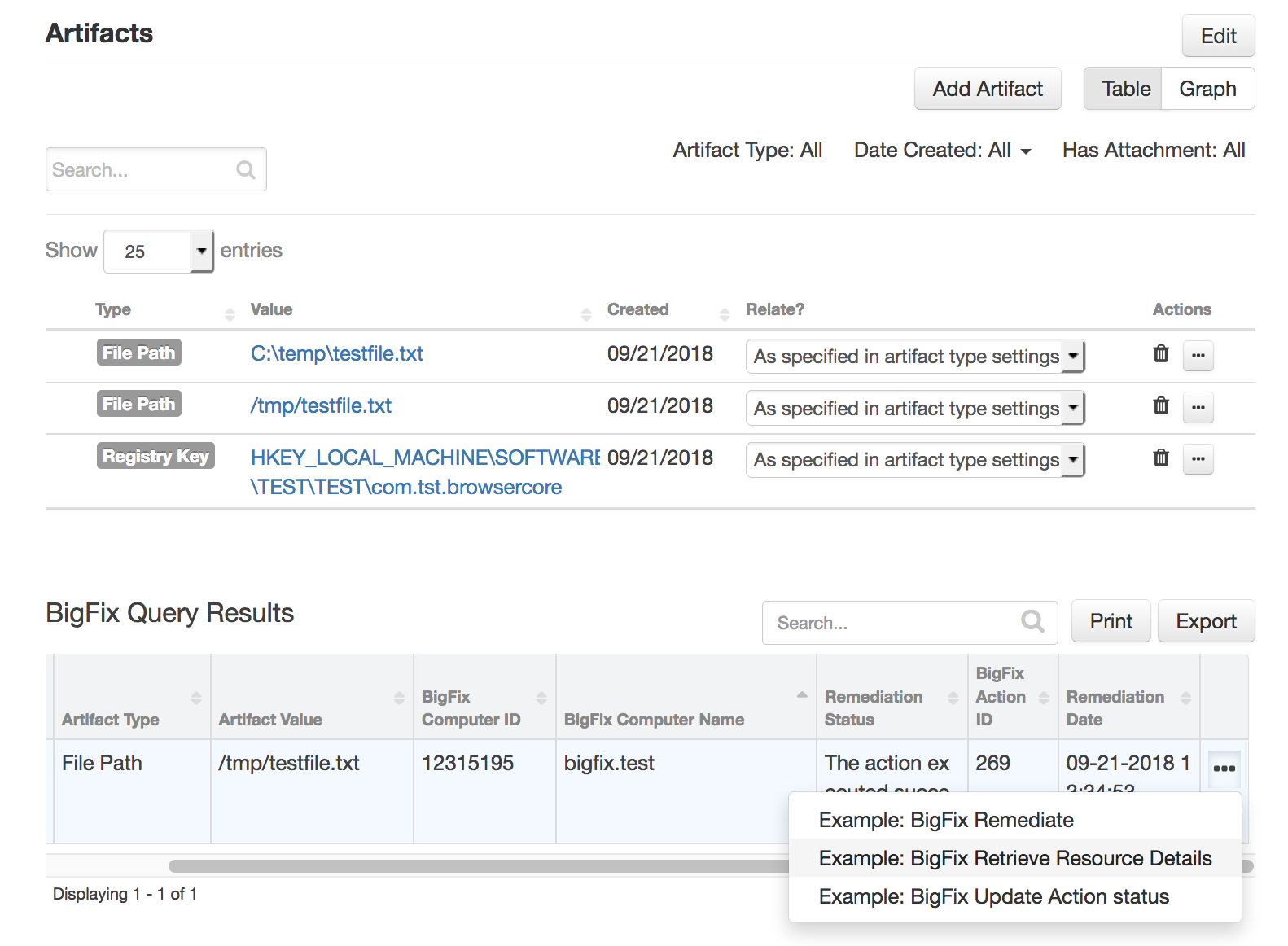
The example workflow (object type = Data Table) that calls this function is “Example: BigFix Retrieve Resource Details”.



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



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.



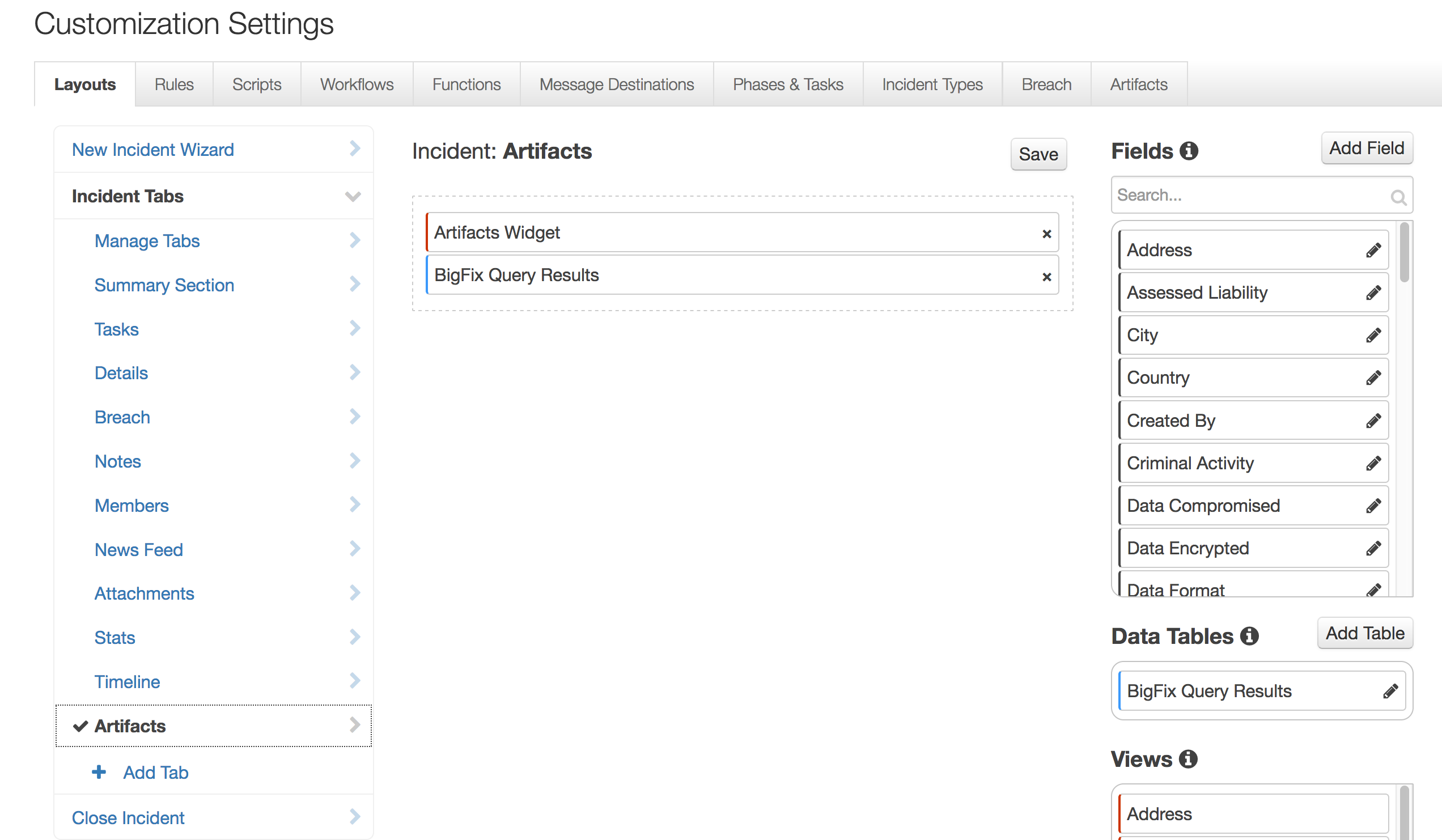
An attachment is added to the incident containing BigFix properties of the targeted endpoint.



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@resilientsystems.com](mailto:support@resilientsystems.com).

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