



Component Files for App Host

Table of Contents

- [About This Package](#)
 - [Container Environment](#)
 - [Requirements](#)
 - [Installation and Configuration](#)
 - [Message Destination Setup](#)
 - [API Key Permission Setup](#)
 - [Adding Additional Python Files after Deployment](#)
 - [Adding Additional Python Packages](#)
-

Revision History

Version	Date	Notes
1.0.0	07/2020	Initial Release

About This Package:

This package is used to convert existing, single-file Python integrations to use the App Host framework. Today, these single-file integrations are dropped into a directory and referenced from resilient-circuits when the `componentsdir` parameter is added to the app.config file.

To use these files in an App Host environment, a container running only resilient-circuits is used and each file is added through the Resilient App Configuration tab.

[← Apps List](#)

fn_components

Status: Ready For Use!

[Details](#)
[Customizations](#)
[Configuration](#)

App Settings

An app can consist of software code and configuration settings. The app.config file contains the settings that allow communication with the Resilient platform. You can add and edit files. You can delete files except app.config.

New File

Search...

File Name	File Location	File Type	Created At	Last Modified
app.config	/etc/rescircuits	Initialization	2020-07-07 12:48	2020-07-07 14:03
create_note_from_data_table.py	/var/rescircuits/components	Python	2020-07-07 14:02	2020-07-07 14:02
remove_duplicate_artifacts.py	/var/rescircuits/components	Python	2020-07-07 13:33	2020-07-07 13:33
fn_sum.py	/var/rescircuits/components	Python	2020-07-07 12:49	2020-07-07 12:49
cert.cer	/etc/rescircuits	Plain Text	2020-07-07 12:48	2020-07-07 12:48

Container Environment

The container runs resilient-circuits similar to an Integration Server and continues to use the `componentsdir` app.config parameter. The following additional Python packages have been added to the container:

- ldap3
- jinja2
- json2html
- pytz
- requests
- resilient-lib
- six
- tldextract

If you require additional Python packages, refer to the section below on how to [modify the container build environment](#).

Requirements

This App Host package assumes that the message destination, functions, and rules for each single-file integration are already defined in your Resilient server. If you require moving your integrations between Resilient instances, consider converting your single-file integrations to fully packaged Apps using the `resilient-sdk codegen` tool and capability.

For each single-file integration:

- Each file must be Python 3 compatible.
- Have no additional Python packages required other than those specified in the [container environment](#).
- Message destinations, functions, and rules used must already exist on your Resilient server.

Installation and Configuration

With the app-fn_components-x.x.x.zip file downloaded from the AppExchange, navigate to the Apps tab within the Administrative Settings and install the package.

Navigate to the Configuration tab and click the New File button to specify the file name, file path and file contents of your single-file integration. Use `/var/rescircuits/components` for the file path and specify the file type as: **Python**.

Finally, cut and paste the content of the single-file Python code into the File Content window. Repeat these steps for each single-file Python integration.

The screenshot shows the 'App Settings / remove_duplicate_artifacts.py' configuration window. At the top, there are 'Cancel' and 'Save and Push Changes' buttons. Below the title, a note states: 'Edit the settings in the file below. File Path specifies a directory path starting at root. If changing location, the system creates the directory if it does not exist. Use a forward slash (/) only to place the file at the root directory. When done, click Save and Push Changes to implement your changes and restart the app.' To the right of this note, the 'Created Date' and 'Last Modified Date' are both listed as '2020-07-07 13:33'. The 'File Name' field contains 'remove_duplicate_artifacts.py'. The 'File Path' field contains '/var/rescircuits/components'. The 'File Description' field is empty with the placeholder text 'Purpose of the file.' and a 'Show more' link. The 'File Content' section has a 'Text or code as appropriate.' label and a 'Show more' link. Below this, there is a 'Theme' dropdown set to 'light' and a 'File Type' dropdown set to 'Python'. The main content area displays a Python script with a license header and a function definition.

```
1 #!/usr/bin/env python
2 # -*- coding: utf-8 -*-
3 # Resilient Systems, Inc. ("Resilient") is willing to license software
4 # or access to software to the company or entity that will be using or
5 # accessing the software and documentation and that you represent as
6 # an employee or authorized agent ("you" or "your") only on the condition
7 # that you accept all of the terms of this license agreement.
8 #
9 # The software and documentation within Resilient's Development Kit are
10 # copyrighted by and contain confidential information of Resilient. By
11 # accessing and/or using this software and documentation, you agree that
12 # while you may make derivative works of them, you:
13 #
14 # 1) will not use the software and documentation or any derivative
15 # works for anything but your internal business purposes in
16 # conjunction your licensed used of Resilient's software, nor
17 # 2) provide or disclose the software and documentation or any
18 # derivative works to any third party.
```

Within the app.config file, add the **[resilient]** parameter: `componentsdir=/var/rescircuits/components`

Each single-file integration may have additional sections and parameters to include in this file similar the settings you have already specified on your Integration Server's app.config file.

← Apps List

fn_components

Status: Ready For Use!
DetailsCustomizationsConfiguration

App Settings / app.config

Edit the settings below. You cannot change the name or location. When done, click Test Configuration to verify the settings then click Save and Push Changes to implement your changes and restart the app.

CancelSave and Push Changes

Created Date: 2020-07-07 12:48
Last Modified Date: 2020-07-07 14:03

File Name

app.config

File Path

/etc/rescircuits

File Annotations

Display any configuration comments and variables to be defined.

Show more

File Content

Text or code as appropriate.

Theme light File Type Initialization

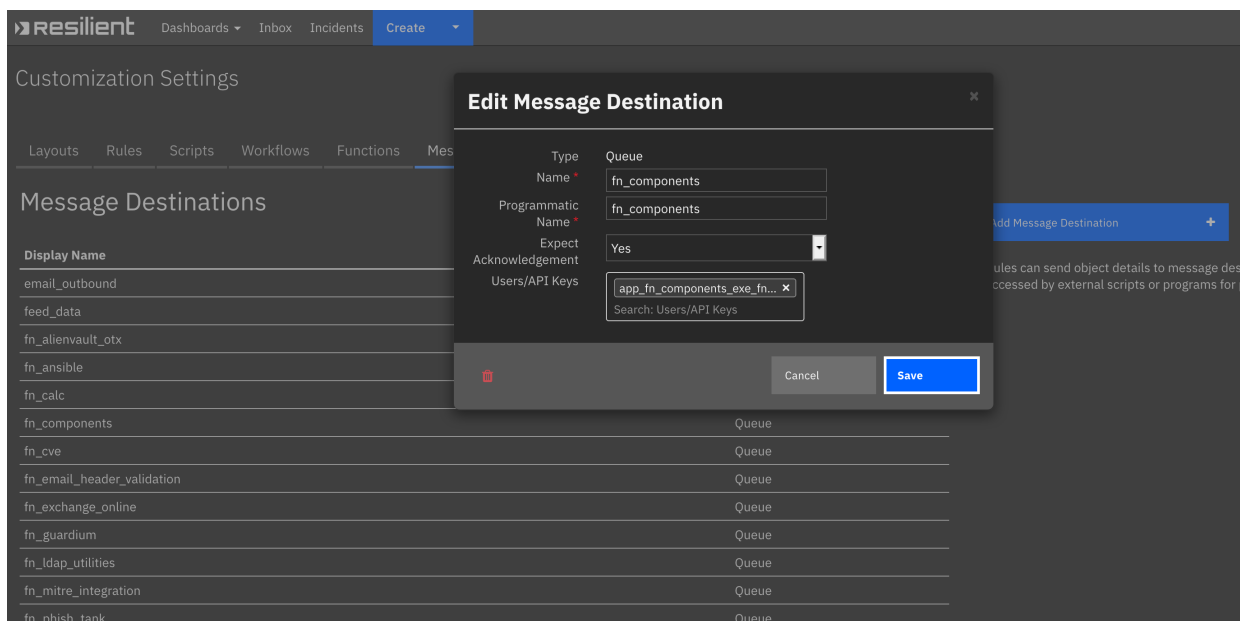
1 [resilient]
2 api_key_id = b90d1285-5a03-4783-8414-62c081ddf0db
3 api_key_secret = kb7HjQ09QGGBwmgY91UoSfqh7QFRnRzw_KUJhah41Fio
4 host = 9.37.29.170
5 port = 443
6 org = testing
7 cafile=false
8 componentsdir=/var/rescircuits/components
9 loglevel=DEBUG
10
11 [remove_duplicate_artifacts]
12 queue=fn_components
13
14 [create_note_from_data_table]
15 queue=fn_components

Once all the single-file integrations and configurations added, return to the Details tab and click on the Deploy button.

Note: Once deployed, your single-file integrations are enabled for rule execution. It is best to remove these files from your Integration Server and restart resilient-circuits as both will be active otherwise.

Message Destination Setup

The next step is to add the API Key created for this container, **app_fn_components_exe_fn_components**, to each of the message destinations used by your single-file integrations. For convenience, all your single-file integrations can share the same message destination.

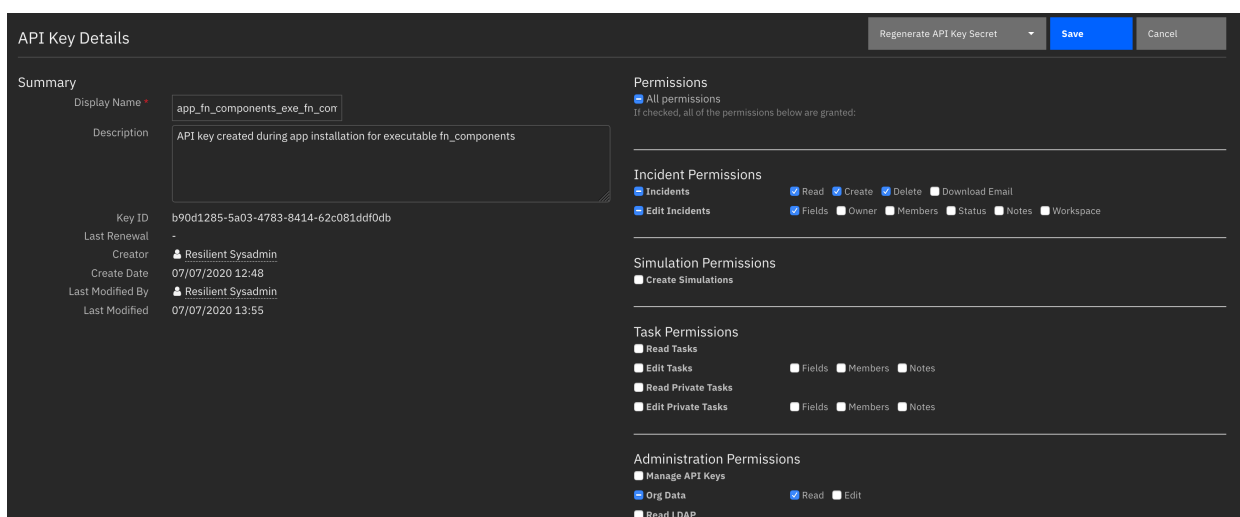


API Key Permission Setup

Since it's not known in advance which API key permissions are required for your single-file integrations, it is necessary to review each integration for the specific API calls performed for their operation. The base permissions for this API key are:

- read and edit incident data.
- create, edit and delete incident elements, such as artifacts, attachments, notes, milestones and tasks.

If your single-file integrations require more or less permissions, edit the key's permissions set as necessary. Insufficient permissions will cause your integration to fail with an error message of **forbidden**.



Adding Additional Python Files after Deployment

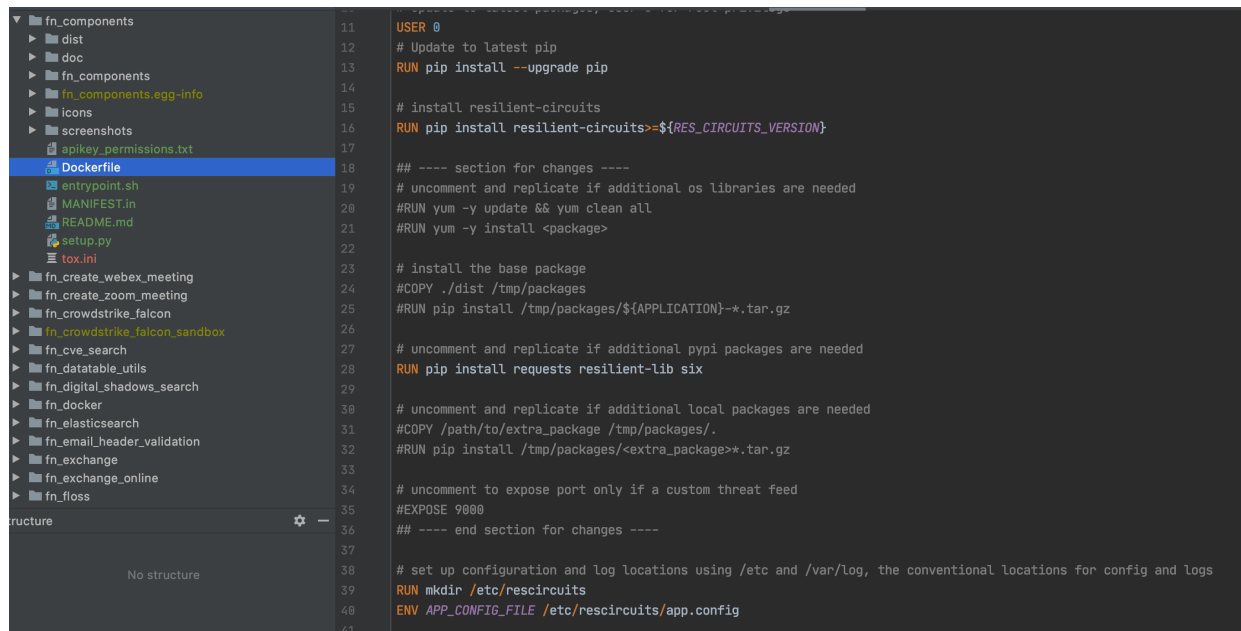
After the initial deployment of your fn_components app, repeat the steps in the [Installation and Configuration](#) section when adding additional single-file Python integrations. Changes to the **app.config** file will trigger the container to restart. Alternatively, click the **Restart** button in the Details tab.

Adding Additional Python Packages

In order to enable the container to include additional Python packages, it is necessary to rebuild the container. This is possible by unzipping the app-fn_components-x.x.x.zip file and then uncompressing the `fn_components-x.x.x.tar.gz` archive. Edit the enclosed `Dockerfile` to include additional Python packages. See the existing RUN command as an example:

RUN pip install requests resilient-lib six

Build the container using either `docker build` or `podman build` in your development environment. You will need to push the new container to your own registry and reference that repository in your App Host. Information on the use of private repositories can be reviewed on the [IBM Knowledge Center](#).



```
11 USER 0
12 # Update to latest pip
13 RUN pip install --upgrade pip
14
15 # install resilient-circuits
16 RUN pip install resilient-circuits>=${RES_CIRCUITS_VERSION}
17
18 ## ---- section for changes ----
19 # uncomment and replicate if additional os libraries are needed
20 #RUN yum -y update && yum clean all
21 #RUN yum -y install <package>
22
23 # install the base package
24 #COPY ./dist /tmp/packages
25 #RUN pip install /tmp/packages/${APPLICATION}/*.tar.gz
26
27 # uncomment and replicate if additional pypi packages are needed
28 RUN pip install requests resilient-lib six
29
30 # uncomment and replicate if additional local packages are needed
31 #COPY /path/to/extra_package /tmp/packages/
32 #RUN pip install /tmp/packages/<extra_package>*.tar.gz
33
34 # uncomment to expose port only if a custom threat feed
35 #EXPOSE 9000
36 ## ---- end section for changes ----
37
38 # set up configuration and log locations using /etc and /var/log, the conventional locations for config and logs
39 RUN mkdir /etc/rescircuits
40 ENV APP_CONFIG_FILE /etc/rescircuits/app.config
```