TC Exchange™ App Development: The Install Configuration File

User Guide

Software Version 4.2 or Newer

July 30, 2020

30031-02 EN Rev. A

ThreatConnect, Inc.
3865 Wilson Blvd., Suite 550, Arlington, VA 22203

P: 1.800.965.2708 | F: .703.229.4489
www.ThreatConnect.com

Table of Contents

OVERVIEW

This document describes the **install.json** file and the powerful features in the declarative configuration JSON format used in all apps run in ThreatConnect®. A TC Exchange app can use this file to facilitate the app installation and job configuration process for users. The file also allows developers to control how the app is configured to reduce the risk of overloading a ThreatConnect instance.

PURPOSE

The purpose of the **install.json** file is to serve as an installation descriptor to ThreatConnect. There are two main components to the file: a Standard section and a Parameter Array section.

NOTE: The *install.json* ***file format is required for all apps installed in ThreatConnect. The file specification outlined in this document applies to Python®, Java®, and Spaces apps.***

STANDARD SECTION

The Standard section defines required and optional properties to all apps in ThreatConnect. The required properties are properties that must be provided for any packaged app installed through ThreatConnect. The optional properties provide additional information based on the type of target app.

Table 1 lists all of the properties of the Standard section.

Table 1

Property	Required?	Allowed Values	Description
programVersion	Yes	Any	This property is the version for this app as it should be displayed on the **System Settings** screen under **Apps**.
programLanguage	Yes	JAVA PYTHON NONE	This property is the language runtime environment used by the ThreatConnect Job Executor. It is relevant for apps that run on the Job Execution Engine and can be set to **NONE** for Spaces apps.
programMain	Yes for Python and Java Apps	Any	This property is the entry point into the app. For Python apps, it is generally the **.py** file (or exclude the extension if running it as a module). For Java apps, it is the main class the Job Execution Engine should use when calling the app using the Java Runtime Environment.
languageVersion	No	Any	This property is used purely for tracking purposes and does not affect the version of Python or Java used by the Job Execution Engine.

runtimeLevel	Yes	Organization SpaceOrganization System	This property describes the type of app and how it should be used within ThreatConnect. For further details on this property, see the “Runtime Level” section.
runtimeContext	No	Array of Strings: Url, Host, Address, EmailAddress, File, Threat, Incident, Email, Document, Signature, Tag, Adversary, Victim, Menu, Search	This property is relevant for SpaceOrganization apps only. This array of Strings enables Spaces apps to be context aware. For further details on this property, see the “Runtime Context” section.
repeatingMinutes	No	Array of Integers Example: `[ 15,30,60,120, 240,360 ]`	This property is a list of minute increments to display in the **Repeat Every…** field on the **Schedule** section of the Job Wizard. This property is relevant only for Python and Java apps for which the developer wants to control how frequently an app can be executed. If this property is not defined, the default listing is as follows: `[ 60, 120, 240, 360, 720 ]`
allowOnDemand	Yes	Boolean	This property allows an app to display the **Run Now** ▶ button in ThreatConnect when configured as a Job.

Runtime Level

The **runtimeLevel** property allows three distinct values that dictate how the app is used within ThreatConnect, as detailed in Table 2.

Table 2

Value	Description
Organization	This value is a Python or Java app that is run by the Job Execution Engine. This type of app must be provisioned to specific organizations (or **Allow All Orgs** must be selected) by the System Admin. Once provisioned, the app can be scheduled to run as part of a Job.
SpaceOrganization	This value is a Spaces app that is run within ThreatConnect as a Space. This type of app must be provisioned to specific organizations (or **Allow All Orgs** must be selected) by the System Admin. Once provisioned, the app can be added as a Spaces app by any user belonging to the Organization.
System	Although not commonly used, the System level is a Python or Java app that is strictly visible by the System Admin. This app can be scheduled only in a System Job.

Runtime Context

The **runtimeContext** property enables Spaces apps to be context aware. Users are able to add Contextually Aware Spaces apps to their Spaces in the respective **Details** page in ThreatConnect. Because this property is an array of Strings, the app can be displayed in multiple Spaces within ThreatConnect, including the **Menu** and **Search** pages.

NOTE: Contextually Aware Spaces apps are passed contextual information via the URL query string when the app is displayed in ThreatConnect. The details of those parameters are out of scope for this document.

PARAMETER ARRAY SECTION

The Parameter Array section of the **install.json** file is the mechanism used by the Job Execution engine and the Spaces framework to pass configuration data at runtime. For Java and Python apps, the entries defined in this section dictate the **Parameters** section of the Job Wizard in ThreatConnect. Spaces apps have their own configuration screen as part of the user's Space for each app.

Table 3 highlights the Parameter Array properties (the **params** array).

Table 3

Property	Required?	Allowed Values	Description
name	Yes	Any	This property is the internal parameter name taken from the Job Wizard and passed to the app at runtime. It is the effective command-line argument name passed to the app.
label	Yes	Any	This property is a description of the parameter displayed in the ThreatConnect Job Wizard or Spaces Config dialog box.
sequence	No	Integer	This property is the number used to control the ordering of the parameters in the Job Wizard or Spaces Config dialog box. If it is not defined, the order of the parameters in the **install.json** file is used.

required	No	Boolean	This property designates this parameter as a required field that must be populated to save the Job. Required parameters would fail an app at runtime or cause unexpected results.
default	No	Any	This property is the default value pre-populated for new Jobs or Spaces. The purpose of a default value is to provide the user with a guidance while allowing edits based on preference.
type	No	String, Choice, MultiChoice, Boolean	Data types enable the UI to display relevant components and allow the Job Executor to adapt how parameters are passed to an app at runtime. For further details on this parameter, see the “Type Parameter” section.
encrypt	No	Boolean	This property designates this parameter as an encrypted value. Parameters defined as encrypted will be managed by the Keychain feature that encrypts password while at rest. This flag should be used with the **String** type and will render a password input textbox in the Job and Spaces configuration.

allowMultiple	No	Boolean	The value of this property is automatically set to **true** if the **MultiChoice** type is used. If a **String** type is used, this flag allows the user to define multiple values in a single input field delimited by a pipe (|) character.
validValues	No	String Array	This property is used with the **Choice** and **MultiChoice** types to restrict the possible values a user can select. For instance, to define a **loggingLevel** parameter, this field could have the following values: ["FATAL", "ERROR", "WARN", "INFO", "DEBUG", "TRACE"].
hidden	No	Boolean	If this property is set to **true**, this parameter will be hidden from the Job Wizard. Hidden parameters allow the developer to persist parameters between job executions without the need to render the values in the Job Wizard. This option is valid only for Python and Java apps. Further details on persisting parameters from the app directly are out of scope for this document.

setup	No	Boolean	This property is reserved for the App Profiles feature. Further details on this feature are out of scope for this document.

NOTE: In Python, parameters are called by using the "--param <value>" syntax handled by the argparse library. For Java apps, the system environment arguments are passed by using the "-Dparam=<value>" syntax. Discussion of app argument parsing is out of scope for this document.

Type Parameter

The **type** parameter serves a dual purpose in ThreatConnect, depending on the actual type defined. Table 4 lists the available types and how they affect elements within ThreatConnect.

Table 4

Type	Description
String	• This type renders an HTML Input textbox in the Job Wizard or Spaces configuration dialog box. This allows the user to enter free-form text as a parameter. • Values are passed as a String to Python and Java apps.
Choice	• This type renders an HTML Select option in the Job Wizard or Spaces configuration dialog box. This allows the user to select predefined text values as a parameter. (See the description of the **validValues** string array property in Table 3.) • Values are passed as a String to Python and Java apps.

MultiChoice	• This type renders an HTML Multi-Checkbox Select option in the Job Wizard or Spaces configuration dialog box. This allows the user to select multiple predefined text values as a parameter. (See the description of the **validValues** string array property in Table 3.) • The same parameter is passed multiple times for a Python app. Python apps should use the argparse “action='append'” option to receive the parameters as an array. Java and Spaces apps will receive the parameter as a single value separated by a | character. • Values are passed as a String to Python and Java apps. • This selection must be used together with the **allowMultiple** flag defined as **true**.
Boolean	• This type renders an HTML Checkbox in the Job Wizard or Spaces configuration dialog box. This allows the user to turn on a flag as a parameter. • Values are passed as a “--flag” style parameter to Python apps. (See the “action='store_true'” option in the argparse module.) Java and Spaces apps receive the actual Boolean value **true** or **false**. These apps should parse the string to resolve the Boolean flag value.

VARIABLE EXPRESSION

The variable-expression feature enables developers to reference “$” style variables in the **install.json** file and have ThreatConnect resolve the values when displayed in the Job Wizard or Spaces configuration dialog box. The external-variables component can go one step further by resolving the value at the time a Job executes. Variable expressions are allowed only in the **params** section of the **install.json** file.

Internal Variables

Internal variables are predefined (reserved) variables that can be explicitly declared in the **install.json** file. Apps declaring these variables will direct the Job Wizard and Spaces configuration dialog box to convert the variables into literal values. Internal variables should be used only with the **Choice** and **MultiChoice** types. They should be defined in the **validValues** property, as in the following parameter definition example:

```
{
    "name": "owner",
    "label": "Owner",
    "type": "choice",
    "validValues": ["${OWNERS}"]
}
```

The variables listed in Table 5 are internal variables understood by ThreatConnect.

Table 5

Variable	Resolves as Type	Example of Usage	Description
OWNERS	String Array	["${OWNERS}"]	The **OWNERS** variable resolves to the available owners to which the current user has access. Since this determination is dynamically resolved at runtime, the owners rendered depend on the user. This variable is useful when an app needs to have a defined owner passed as a parameter. The string value of the owner(s) is passed as an argument to the app.

ATTRIBUTES	String Array	["${ATTRIBUTES:Address}"]	The **ATTRIBUTES** variable resolves to attributes the current organization has available. This variable has a second, *optional*, component, `:<type>`, that further refines the attributes resolved to the specific Indicator or group type. This component gives the developer further control over the attribute type values rendered at runtime. The string value of the attribute(s) is passed as an argument to the app.

When the `$ATTRIBUTES` internal variable is used with a `:<type>` suffix, the types can be any of the Indicator or Group types in ThreatConnect.

External Variables

External variables offer the user an additional level of convenience by directing the Job Wizard and Spaces configuration dialog box to take advantage of the Variables feature.

NOTE: The Variables feature in ThreatConnect allows any user to create variable key/value pairs. Once created, these values can be selected by the user in the Job Wizard or Spaces configuration dialog box to reduce the need to copy and paste keys and plain-text data.

Since the variable names are not known by the app developer, the generic form of the variables is referenced instead in a **<level:type>** format. For instance, to allow the user to select one of their plain-text variables from Organization and User levels, the **install.json** file would reference them as follows:

```
"validValues": ["${USER:TEXT}", "${ORGANIZATION:TEXT}"]
```

The left-hand component of the variable is the level. The level can be any of the options listed in Table 6.

Table 6

Level Option	Description
User	This option displays the list of the user's variables in the Job Wizard or Spaces configuration dialog box.
Organization	This option displays the list of Organization variables available to the current user in the Job Wizard or Spaces configuration dialog box.
System	This option displays the list of system variables available to the current user in the Job Wizard or Spaces configuration dialog box.

The right-hand component of the variable is the type. The type can either of the options listed in Table 7.

Table 7

Type Option	Description
Text	This option restricts the values in the level to those variables defined as plain text.
Keychain	This option restricts the values in the level to those variables defined as keychain. These parameters are typically set to **encrypt: true** in the configuration.

Multiple external-variable expressions can be included in string array form.

EXAMPLE JSON FILE

This section provides an example of an **install.json** file for a Python app. The key elements are described with line-number references in Table 8, below the example.

```
1   {
 2    "programVersion" : "1.0.0",
 3    "programLanguage" : "PYTHON",
 4    "programMain" : "auto_enrich",
 5    "languageVersion" : "2.7",
 6    "runtimeLevel" : "Organization",
 7    "allowOnDemand" : true,
 8    "params" : [ {
 9      "name" : "api_access_id",
10      "label" : "Local ThreatConnect API Access ID",
11      "sequence" : 1,
12      "required" : true,
13      "validValues": ["${USER:TEXT}", "${ORGANIZATION:TEXT}"]
14    }, {
15      "name" : "api_secret_key",
16      "label" : "Local ThreatConnect API Secret Key",
17      "sequence" : 2,
18      "encrypt" : true,
19      "required" : true,
20      "validValues": ["${USER:KEYCHAIN}", "${ORGANIZATION:KEYCHAIN}"]
21    }, {
22      "name" : "owner",
23      "label" : "Destination Owner",
24      "sequence" : 3,
25      "required" : true,
26      "type": "choice",
27      "validValues": ["${OWNERS}"]
28    }, {
29      "name" : "remote_api_access_id",
30      "label" : "Remote ThreatConnect API Access ID",
31      "sequence" : 4,
32      "required" : true,
33      "validValues": ["${USER:TEXT}", "${ORGANIZATION:TEXT}"]
34    }, {
35      "name" : "remote_api_secret_key",
36      "label" : "Remote ThreatConnect API Secret Key",
37      "sequence" : 5,
    "encrypt" : true,
    "required" : true,
    "validValues": ["${USER:KEYCHAIN}", "${ORGANIZATION:KEYCHAIN}"]
  }, {
    "name" : "remote_api_path",
    "label" : "Remote ThreatConnect API Path",
    "sequence" : 6,
    "required" : true,
    "default" : "https://api.threatconnect.com",
    "validValues": ["${USER:TEXT}", "${ORGANIZATION:TEXT}"]
  }, {
    "name" : "remote_owner",
    "label" : "Remote Owner",
    "sequence" : 7,
    "required" : true
  }, {
    "name" : "apply_threat_assess_rating",
    "label" : "Apply ThreatAssessRating from Remote Owner",
    "type" : "Boolean",
    "sequence" : 8
  }, {
    "name" : "apply_rating",
    "label" : "Apply Rating from Remote Owner if ThreatAssesRating
is not Available",
    "type" : "Boolean",
    "sequence" : 9
  }, {
    "name" : "apply_threat_assess_confidence",
    "label" : "Apply ThreatAssessConfidence from Remote Owner",
    "type" : "Boolean",
    "sequence" : 10
  }, {
    "name" : "apply_confidence",
    "label" : "Apply Confidence from Remote Owner if
ThreatAssessConfidence is not Available",
    "type" : "Boolean",
    "sequence" : 11
  }, {
    "name" : "apply_tags",
    "label" : "Apply Tags from Remote Owner",
    "type" : "Boolean",
    "sequence" : 12
  }, {
    "name" : "apply_auto_enrich_tag",
    "label" : "Apply 'AutoEnriched' Tag",
    "type" : "Boolean",
    "sequence" : 13
  }, {
    "name" : "apply_proxy_tc",
    "label" : "Apply Proxy to Local API Connection",
    "type" : "Boolean",
    "sequence" : 14,
    "default" : false
  }, {
    "name" : "apply_proxy_ext",
    "label" : "Apply Proxy to Remote API Connection",
    "type" : "Boolean",
    "sequence" : 15,
    "default" : false
  }, {
    "name" : "logging",
    "label" : "Logging Level",
    "sequence" : 16,
    "default" : "info",
    "type": "choice",
    "validValues": ["debug", "info", "warning", "error", "critical"]
  } ]
}
```

Table 8

Level Option	Description
2	The **programVersion** is 1.0.0. This value is rendered in the apps listing for System Administrators.
4	The **programMain** will direct the Job Executor to run this app as a main module.
6	The **runtimeLevel** for this app is **Organization**. This app will allow Jobs to be configured only for an Organization (assuming the System Admin has provisioned the Org).
8	This line is the start of the **params** array. The contents in this array are purely for parameter definitions.
9–13	This parameter describes the **api_access_id** argument for the app. The app will be passed an argument called **--api_access_id** at execution time. The label in the Job Wizard will be **Local ThreatConnect API Access ID**. Since the sequence is defined as **1**, this parameter will be the first parameter displayed in the Job Wizard. This parameter is required, and the user can benefit from User- and Organization-level plain-text variables, if defined. Otherwise, the user is allowed to enter freeform text (the default type if no variables are defined).

35–40	This parameter describes the **remote_api_secret_key** argument for the app. The app will be passed an argument called **--remote_api_secret_key** at execution time. The label in the Job Wizard will be **Remote ThreatConnect API Secret Key**. This parameter will be the 5th parameter in the Job Wizard **Parameters** tab. Since the parameter is set to **encrypt**, the input field will be displayed as a password with a masked value. Encrypted parameters will also be stored in encrypted form in the database. At runtime, the decrypted password will be passed to the app. Finally, the user can benefit from User- and Organization-level keychain variables, if defined. Otherwise, the user is allowed to enter freeform password text.
65–68	This parameter describes the **apply_threat_assess_confidence** Boolean argument for the app. The app will be passed an argument called **--apply_threat_assess_confidence** at execution time *only if* the user selects this value in the Job Wizard. The Job Wizard will display a label called **Apply ThreatAssessRating from Remote Owner**, along with a checkbox. The “argparse” style flag (without an argument) and the checkbox displayed in the Job Wizard are dictated by the **Boolean** type in the parameter definition. This parameter will be the 8th parameter in the Job Wizard **Parameters** tab.

98–103	This parameter describes the **logging** argument for the app. The app will be passed a parameter named **--logging** with a string argument. The **Logging Level** label will be displayed in the Job Wizard. This parameter will be the 16th (and last) parameter in the Job Wizard **Parameters** tab. The type for this parameter is **Choice**, and the definition dictates that a valid value for this parameter is one of **debug**, **info**, **warning**, **error**, or **critical**. The user will not be able to edit this dropdown list, and the default value for new Jobs will be logging level **info**.