**Overview**

The task is used to tokenize environment specific configuration that is maintained in XML or any other text documents. This task allows translating XML documents with environment specific values by setting values based on XPath and/or replace the tokens that match the regular expression \_\_<variable>\_\_

A configuration Json document is provided as an input that contains a section **ConfigChanges** to provide **KeyName** the XPath to identify a particular node in the XML document, **Attribute** name that needs to be set and **Value** to be set. And this configuration can be maintained for multiple environments.

Below is the sample for the Json document that can be provided as input. There can be multiple sections for each <environment>

{

"<environment>": {

"CustomVariables": {

"Variable1": "value1",

"Variable2": "value2",

},

"ConfigChanges": [

{

"KeyName": "/configuration/appSettings/add[@key='ServiceURL']",

"Attribute":"value",

"Value":"https://ServiceURL"

}

,

{

"KeyName": "/configuration/appSettings/add[@key='EnableDebugging']",

"Attribute":"value",

"Value":"false"

}

,

{

"KeyName":“/configuration/connectionStrings/add[@name='databaseentities']”,

"Attribute": "connectionString",

"value": "Integrated Security=True;Persist Security Info=False;Initial Catalog=DB;Data Source=servername"

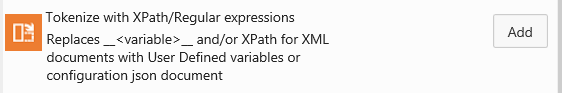
}

]

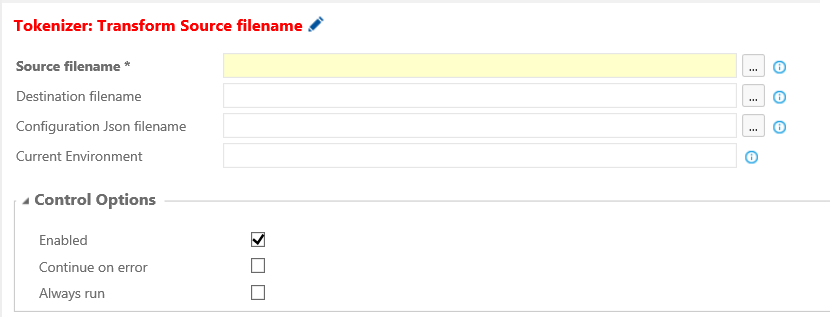
}

**Task Parameters**

Following task is added to the Utility group of Build/Release tasks



Below is the list of inputs for the task



Source filename – **required,** Source file name that contains the tokens (\_\_<variable-name>\_\_). These patterns will be replaced with user-defined variables or from Configuration Json FileName. If it is an XML document, XPaths mentioned in the Configuration JsonFileName will be set as per environment

Destination filename – **optional,** Destination filename that has transformed Source filename. If this is empty, the 'Source filename' will be modified.

Configuration Json filename – **optional,** Json file that contains environment specific settings in the form XPath, Attribute, Value and values for user-defined variables.

Refer above for the schema/format of the Json filename. If this parameter is not specified, then custom variables mentioned against the build/release are used to replace the tokens that match the regular expression \_\_<variable-name>\_\_

Current Environment – **optional,** Environment settings that need to be used from the Configuration Json filename. One can specify $RELEASE\_ENVIRONMENTNAME if the configuration is setup with Release Manager. If no value specified for this parameter, then use “default” as the environmentName

**How does it work?**

Tokenization in the following scenarios

1. An xml configuration file like Web.Config/app.config/Service.xml which will have different values for XML nodes based on the target deployment environment
2. Accept the input at the time of deployment (for instance Octopus role variables or RM custom variables) and replace regular expressions in an application configuration file with the input values.

Below is how the Source filename gets transformed

1. If **Source filename** is an XML document, run through all the XPaths in the Json document Configuration Json filename and transform the XML document
2. For each token in the *Source filename* that matches with the regular expression \_\_<variable-name>\_\_
   1. If there is a custom variable at build or release definition with the name *variable-name*, then replace the token with the value of the same.
   2. Else If the variable is available in the corresponding environment section of json document then replace the token with the value from json document
   3. Or else it ignores the token