**Description**

Purpose of DataTarget : If a activity's sub element has attribute: target and mode, means, it is a DataTarget-capable element, this element can save data to target location: XmlFile, FlatFile or Dictionary according to the attribute description and you still can choice "Append" or "Replace" mode for data saving operation. You don't care what data will be saved, this element knows about this.

**Syntax**

**<Element XXX**

**target=**

“**XmlFile**”: if target value is XmlFile, the Element's innert txt format will be a Xpath🡺 xml file full path + | + XPath, "|" is separate symbol.

**like: C:\XmlDoc\Test.xml | //Wistron/Test/Test1.**

Note: The XPath can not be with attribute description and start of XPath must be with "//".

like:

**ilegal 1: //Wistron/Test/Test1[@name="super"]**

**ilegal2:/Wistron/Test/Test1**

“**FlatFile**”: If target value is FlatFile, Element's innert txt will be file full path of flat file. Like:

**C:\Wistron.txt**

“**Dictionary**”: If target value is Dictionary, the Element's innert txt will be Dictionary’s Key value. like:

**ABC**

**mode=**

“**Append**”: the new value will be appened to the file, this operation is only support by XmlFile

“**Replace**”: the new value will overwrite all old same values, if there is no any existing value, the new value will be added directly, this operation is support for all target options. **>**

***Element’s content: Full file path of XmlFile and FlatFile or Keyname for dictionary***

1. If you input any none supporting target or mode value, you will get fail result.

2. If you do not follow the format of element's innter txt for each target type, you will get fail result.

**</Element XXX>**

**Example**

**For XmlFile**

This is test.xml's content and activity's sub element, which name is ProcessID, we assume Process gets a value:d, now you want to let Process know what it should save data to and use Replace mode. You should give:

<ProcessID target="XmlFile" mode="Replace">C:\test.xml | //ID/processID</ProcessID>

<ID>

<processID>a</processID>

<processID>b</processID>

<processID>c</processID>

</ID>

so the content will become:

<ID>

<processID>d</processID>

</ID>

Keep above example, if you want to use "Append", you should use below description

<ProcessID target="XmlFile" mode="Append">C:\test.xml | //ID/processID</ProcessID>

<ID>

<processID>a</processID>

<processID>b</processID>

<processID>c</processID>

</ID>

because mode is Append, so the content will become:

<ID>

<processID>a</processID>

<processID>b</processID>

<processID>c</processID>

<processID>d</processID>

</ID>

If xml file doest not exsit in the computer or xml file content has any wrong xml stanadard format, the system will generate a new one, then save the content into the file accrodint to the xml element description.

**For FlatFile**

If a text.txt's content is "support" and ProcessID element will write "support2" into the test.txt:

you should use below description:

<ProcessID target="FlatFile" mode=" Replace">C:\test.txt</ProcessID>

or

<ProcessID target="FlatFile>C:\test.txt</ProcessID>

c:\test.txt, so the content will become:

support2

But if you descrbied:

<ProcessID target="FlatFile" mode="Append">C:\test.txt</ProcessID>

--> it will get fail result, because "Append" is not supporting for FlatFile

**For Dictionary**

ProcessID will write "d" into Dictionary, which is in provided by AutoFlow

<ProcessID target="Dictionary" mode="Replace">Process</ProcessID>

or

<ProcessID target="Dictionary">Process</ProcessID>

<ProcessID target="Dictionary" mode="Append">Process</ProcessID>

This action will get fail result, because this is none support mechanism

If system's dictionary has not this key:Process, system will save the key-value pair:

Process=d, but if system have had already the key : Process and its value is "K", system

will replace the original one, means, K insteads of d.