TAZ

User Reference

Xpects – Advanced Usage

Date: 9/18/2014

Revision: 1.4

Overview

In TAZ, **Action** commands produce an output, called a ***Return***, and **Evaluation** commands score a Pass or Fail, based on User provided criteria. **Xpect{…}** is the basic **Evaluation** command which works well for most situations. In addition, **XpectPO{…}**, **XpectET{…}**, and **XpectReg{…}** offer additional ways to Pass/Fail a ***TestCase***.

This paper deals with using the different **Xpect** commands with more complex ***Returns***. Please also watch our videos on basic **Xpect** usage and refer to the ***TAZ Command Manual*** and ***TAZ User Manual*** for more information.

Xpect – Advanced options

With **Xpect,** specifying only ***ArgA*** will suffice for most situations. **Xpect** by default starts searching for a match from the beginning of all ***Returns*** in a ***TestStep***. There are times when we need more control. This is when we use ***ArgB*** and ***ArgC***. Consider this ***Return*** from a command that outputs port status:

```
1/1/1:CriticalAlarm
1/1/2:MajorAlarm
1/2/4:NoAlarms
1/3/1:MinorAlarm
```

Xpect{NoAlarms} will pass because **NoAlarms** is found somewhere in the ***Return***, but we have no information about the port. Using only ***ArgA*** we can make a slight modification and pinpoint the port of interest. For example:

Xpect{1/1/2&NoAlarms} will only focus on a line that contains **1/1/2** and **NoAlarms**.

But what if we are interested in port 1/1/2 and this is the **Return** that comes back:

```
Port 1/1/1
     Enabled:True
     Alarms:Critical
Port 1/1/2:
     Enabled:True
     Alarms:Major
Port 1/2/4:
     Enabled:True
     Alarms:NoAlarms
Port 1/3/1:
     Enabled:True
     Alarms:Minor
```

The previous **Xpect** will not work because our defining information (**1/1/2**) is not on the same line as the result (**Alarms:Major**). In this case we use ***ArgB*** and ***ArgC***, the starting and ending positions. Rather than looking at the entire ***Return***, these will define the section of the ***Return*** to include in our evaluation.

Xpect{Alarms:NoAlarms|Port 1/1/2|Port} will direct the evaluation to the section of interest. ***ArgB***, the starting position, says start looking after **Port 1/1/2** and ***ArgC***, the ending position, says stop looking before the next instance of **Port**. If ***ArgC***, the ending position, is not found, TAZ will look to the end of the ***Returns***.

Misc Xpect Features

- You can use regular expressions (PCRE) for ***ArgB*** and ***ArgC***. If you would like to use a regular expression for ***ArgA***, then use **XpectReg{…}**, which will pass if a match for the regular expression in ***ArgA*** is found.
- You can use **Not** to flip the evaluation for all **Xpect** commands; **XpectNot{…}**, **XpectPONot{…}**, etc.. So if **Xpect{abc}** were to evaluate to TRUE, **XpectNot{abc}** will evaluate to FALSE.

- By default, **Xpect{…}** is not case-sensitive. To make case-sensitive enter **CaseSensitive** in *ArgE*.

XpectPO{…} and XpectET{…}

Where **Xpect{…}** evaluates the *Returns* from **Action** commands, **XpectPO{…}** and **XpectET{…}** evaluate the values from their respective tables. The *PassOut Table (PO Table)* and *ETable* are powerful ways to capture and manipulate runtime data. Please refer to the *TAZ Command Manual* and other Help documents and videos for information on populating these tables.

The Relationship Operator - ArgB

XpectPO{…} and **XpectET{…}** have the same structure and are essentially the same. They each have three arguments. *ArgB* contains an operator that relates *ArgA* to *ArgC*. If the relationship is TRUE, the command evaluates to a Pass. Possible Operators are:

- **=**
- **>**
- **>=**
- **<**
- **<=**
- **<>**
- **Range**
- **+-** ((<num>))
- **+-** ((<num>**%**)
- **Contains**

See the *TAZ Command Manual* for operator details.

ArgA and ArgC

Although the command **XpectPO{NoAlarms|=|NoAlarms}** is valid and will evaluate to Pass, it is not very useful. It says "**NoAlarms**" equals "**NoAlarms**", which we obviously already know. To make this useful, we need to modify *ArgA*, *ArgC*, or both.

Let's use our *Return* from above:

```
1/1/1:CriticalAlarm
1/1/2:MajorAlarm
1/2/4:NoAlarms
1/3/1:MinorAlarm
```

To enter a value in the *PO Table* , we will use the command:

PassOutBetween{1/2/4:|((EOL))} – this adds the value between "**1/2/4:"** and the **End of Line** to the *PO Table*. In this *Return* that value is **NoAlarms**, but in other instances may be different.

Our *PO Table* is:

Element	Value	Name	Description
1	**NoAlarms**		

To make our **XpectPO** command useful, we need to use the value from our *PO Table*. To do this we use variable substitution. To insert a *PO Table* value we can use the variable, **PO[**<element number>**]**, for example **PO[**1**]**. This variable syntax may be a part of any command in TAZ, not just **XpectPO{…}**.

Let's change our command to **XpectPO{PO[1]|=|NoAlarms}**. At runtime TAZ replaces **PO[**1**]** in *ArgA* with the value of element 1, in this case **NoAlarms**. **PO[**1**]** is a variable and the circumstances of our test determine its value. In this case the command passes.

Strings, like **NoAlarms**, work with the operators equal (**=**), not equal (**<>**), and **Contains**. All operators, except **Contains**, work with numeric values or expressions.

Here are some numeric examples that evaluate to Pass:

Our ***PO Table*** is:

Element	Value	Name	Description
1	**24**		
2	**5**		
3	**12**		
4	**29**		

- **XpectPO{PO[2]|<=|5}** – the value of element 2 is **less than or equal to** 5
- **XpectPO{PO[1]+PO[2]|=|PO[4]}** – the value of element 1 plus element 2 **equals** the value of element 4
- **XpectPO{PO[1]/2|=|PO[3]}** – the value of element 1 divided by 2 **equals** the value of element 3
- **XpectPO{PO[4]|>|PO[1]+3}** – the value of element 4 is **greater than** the value of element 1 plus 3

XpectET{…}

The only difference between **XpectPO{…}** and **XpectET{…}** is that variable substitution from the ***ETable*** requires 3 comma separated coordinates, **ET[**<which page>**,** <which row>**,** <which column>**]**. You can also combine variable substitution in **XpectPO{…}** and **XpectET{…}**. So

XpectPO{PO[1]|=|ET[1,2,4]} is equivalent to

XpectET{PO[1]|=|ET[1,2,4]}

Variable Substitutions – PO[…]

Up to now we have spoken of the variable substitutions, **PO[…]** and **ET[…]**, in terms of numeric coordinates. This does not have to be the case. Let's look at a ***PO Table*** where we have given the elements a name:

Element	Value	Name	Description
1	**24**	**TwoDozen**	
2	**5**		
3	**12**	**OneDozen**	
4	2**9**		

Previously we used **XpectPO{PO[1]/2|=|PO[3]}** with element numbers for our variable substitutions.

We can also refer to the elements by name

XpectPO{PO[TwoDozen]/2|=|PO[OneDozen]}

where the value of the named element **TwoDozen** is 24 and the value of the named element **OneDozen** is 12, therefore 24 divided by 2 equals 12 is True. Referring to ***PO Table*** elements by name is particularly useful because the element value is not tied to its position in the ***PO Table***.

Variable Substitutions – ET[...]

In variable substitutions from the ***ETables***, the first coordinate (page number), must always be a number. We can use names for the row or the column (and an instance number). These names may be found anywhere in the row or column, but usually they are the row header or column header. Here are some examples:

```
ET[1,3,2] will be replaced with the value at
     Page = 1
     Row  = 3
     Col  = 2
ET[2,firmware version,3] will be replaced with the value at
     Page = 2
     Row  = the first row that has a cell value firmware version
     Col  = 3
ET[1,4,rate((2))] will be replaced with the value at
     Page = 1
     Row  = 4
     Col  = the second column instance that has a cell value rate.

     The instance parameter is optional and defaults to the first instance

ET[1,firmware version,rate((2))] will be replaced with the value at
     Page = 1
     Row  = the first row that has a cell value firmware version
     Col  = the second column that has a cell value rate
```

There is a special method for XML that we will examine in the next section.

XML –ET[…]

As an example, let's create an ETable using **ETableDelimiter{ETable1|<cat>((XML))}** with the **XML** option. The *ETable* is formed with each "cat" tag on a separate row.

```
<function>pet
<name>zacho</name>          <type>manx</type>     </cat>
<name>blanca</name>         <type>manx</type>     </cat>
<name>Zoro</name>           <type>feral</type>    </cat>     </function>
```

When we use a value from this *ETable*, we can choose any variable substitution method, but there is a special method for **XML.**

To get the **type** of the cat named **blanca** we could use **ET[**1,3,2**]**. But if more rows or columns are added, our coordinates will be off.

Instead we can use **ET[**1,**<name>blanca**,<type>**]**. This will always get the **type** of cat (stripped of the XML tags) whose name is **blanca**. Even if columns or rows are added, **ET[**1,<name>blanca,<type>**]** will always be replaced with **manx**. The XML tag formatting <type>, tells TAZ to use this special XML method.

Third Party Disclaimer

This disclaimer pertains to all third-parties, including but not limited to:

- Spirent Communications plc
- Ixia
- Shennick
- Ameritec
- Mozilla
- Selenium
- Net-SNMP.org
- Microsoft
- Cisco
- APC (Schneider Electric)
- ControlByWeb
- 3G Store
- Net-SNMP
- Adobe
- TestLink
- Oracle
- Google

AZTAZ Software, LLC is not affiliated, associated, authorized, endorsed by, or in any way connected to any third-party or third-party products referenced in any AZTAZ Software, LLC products, including but not limited to our website, software, marketing, videos, documentation or communications.

AZTAZ Software, LLC does not control, endorse, or accept responsibility for any materials, products, websites, or services offered by third parties that we may reference.

All product and company names are trademarks™ or registered® trademarks of their respective holders. Use of them does not imply any affiliation with or endorsement by their respective owners.