

Accessing TStrings Based Properties

Some Action properties are of type TStrings. TStrings is a standard Delphi type that we have exposed to the scripting engine. It is basically a string collection. Below are the properties of TStrings that you will find useful:

| Method Call | Example | Description |
|---|--|--|
| <code>procedure Clear;</code> | <code>Action.ListOfItems. Clear;</code> | Clears the TStrings object of all the strings it contains. |
| <code>function Add(const text: string): Integer;</code> | <code>var index = Action. ListOfItems.Add('Item 5');</code> | Appends a string to the end of the list of string the TStrings object contains. The index of the newly added text is returned. |
| <code>procedure Delete (index: Integer);</code> | <code>Action.ListOfItems. Delete(5);</code> | Deletes the string located at the passed in index. If the index does not exist and error is generated. |
| <code>procedure Insert (index: Integer; const text: string);</code> | <code>Action.ListOfItems. Insert(5, 'Item 5');</code> | Inserts the passed in text at the passed in index. If the index is outside the bounds of the list which the TStrings object holds and error is generated. |
| <code>function IndexOf (const text: string): Integer;</code> | <code>Action.ListOfItems. IndexOf('Item 5');</code> | Returns the index of the passed in string. If it doesn't exist in the list of string held by the TStrings object, -1 is returned. |
| <code>property Strings [index: Integer]: string;</code> | <code>var item5Text = Action.ListOfItems. Strings[5];</code> | Returns the text held at the passed in index. The index is zero based. If the index does not exist in the current list of string help by the TStrings object an error is raised. |
| <code>property Count: Integer;</code> | <code>var count = Action. ListOfItems.Count;</code> | Returns the number of string held by the TStrings object. |
| <code>property Text: string;</code> | <code>var list = Action. ListOfItems.Text;</code> | Returns all the strings in the collection, each entry on a new line. |
| <code>property Names[index: Integer]: string;</code> | <code>var item5Name = Action.ListOfItems. Names[5];</code> | Returns the name string held by the TStrings object at the passed in index. Strings can be stored in name=value pairs. This method allows access to the name part when this storage method is used. |
| <code>property Values[name: String]: string;</code> | <code>var item5Value = Action.ListOfItems. Values[item5Name];</code> | Returns the value string held by the TStrings object for the passed in name. Strings can be stored in name=value pairs. This method allows access to the value part when this storage method is used. |
| <code>property ValueFromIndex[index: Integer] : string;</code> | <code>var item5Value = Action.ListOfItems. ValueFromIndex[5];</code> | Returns the value string help by the TStrings object for the passed in index. Strings can be stored in name=value pairs. This method allows access to the value part when this storage method is used. |

Example

The following code reduces each item in a List Iterator's list of items down to its leftmost three characters:

