

Custom Properties

Overview

Custom properties are local variables created for a specific Display. They can be accessed in CodeBehind and in the Display itself.

Custom properties can also be linked to Tags, by setting the Property Name as "@Tag.<TagName>", where <TagName> is a valid name in **Edit > Tags > Objects**.

The properties added to your display are created as an array format. They are accessed through an index. The index ordination depends on the order the properties were created.

E.g.:

```
@Display.<PageName>.CustomProperties[index];
```

This property can be used as an alternative to using tags in any logic.

An advantage is the possibility of decreasing the total communication points while maintaining the same project functionalities. This can be very useful depending on your license type.

On the other hand, the logic programmed with the CustomProperties are only valid for a specific Display. Meaning you can not retain the data from one Page for use in another.

Methods Available in Runtime

GetCustomPropertiesAsString(string sep)

- Get all custom properties
- Default Separator is ", "
- Returns a string

Example

```
string msg = @Display.About.GetCustomPropertiesAsString(",");
```

GetCustomPropertyValue(string PropertyName,string defaultValue):

- Get value of custom property name
- **propertyName**: Property Name
- **defaultValue**: Default value if property could not be found
- Returns property value

Example

```
string propValue = @Display.MainPage.GetCustomPropertyValue("p1","-1").ToString();
```

SetCustomProperties(string str, string sep)

- Set several custom properties at the same time
- **str**: String containing custom properties separated by 'sep'
- **sep**: Separator to each custom property

Example

```
@Display.MainPage.SetCustomProperties("p1=10;p2=20", ";");
```

SetCustomPropertyValue(string propertyName, object propertyValue)

- Set custom property. If the property does not exist, it is created.
- **propertyName**: Property Name. If the name starts with "@" and the rest of the string is a valid tag name, the property is a reference to the Tag.
- **propertyValue**: Property value
- Returns always 0. Not used.

Example

```
@Display.MainPage.SetCustomPropertyValue( "@Tag.Integer1", 50 );
```

RemoveAllCustomProperties():

- Remove all custom properties
- Returns always. Not used.

Example

```
@Display.LogOn.RemoveAllCustomProperties();
```



Note

If a Tag's Custom Property is removed, the Tag will retain the old property as its current value.