CODESYS STORE

FB Visu Creator

Das Produkt „FB Visu Creator“ ist ein Plug-In zum automatischen Erstellen von Visualisierungen aus Funktionsbausteinen von Bibliotheken und POU-Pools.

Produktbeschreibung

Lizenzierung:

Es wird keine Lizenz benötigt.

Mit dem Plug-In „FB Visu Creator“ können automatisch Visualisierungen aus Funktionsbausteinen (FBs) erstellt werden. Über Attribute kann gesteuert werden, welche Funktionsbausteine, Ein- und Ausgänge visualisiert werden sollen. Pro Baustein wird eine Visualisierung erzeugt. Die generierten Visualisierungen können anschließend über einen Frame mit den gewünschten FB-Instanzen verknüpft werden.

Die Generierung der Visualisierungen wird über Attribute gesteuert. Im Folgenden werden die möglichen Attribute der Funktionsbausteine und deren Variablen beschrieben.

1.1 Attribute für Funktionsbausteine

Attribut	Funktion
`{attribute 'FBVisuCreator'}`	Markiert ein Funktionsbaustein zur Generierung einer Visualisierung. Beispiel (Der FB „Test“ soll visualisiert werden): `{attribute FBVisuCreator'} FUNCTION_BLOCK Test`

`{attribute 'FBVisuCreatorTextColor':= 'COLOR' }`	Farbe des Textes.
	Argument: Name der Farbe (englisch) oder #rrggbb als Farbe (hexadezimal, wie HTML-Farbcodes, z.B. #000000 für Schwarz oder #ffffff für weiß).
	Beispiel:
	`{attribute 'FBVisuCreatorTextColor' := 'Green'}`
	`{attribute 'FBVisuCreatorTextColor' := "#4223FF"}`

`{attribute 'FBVisuCreatorTextStyleColor' := 'COLOR' }`	Anzeigefarbe des Textes aus dem Style. Argument: Case-Sensitiver Name der gewünschten Stilfarbe, die in der Datei `styledef.xml` definiert ist, z.B. Element-Alarm-Color“. Beispiel: `{attribute 'FBVisuCreatorTextStyleColor' := 'Element-Alarm-Color'}` Wird `TextStyleColor` und `TextColor` gleichzeitig gesetzt, so liegt die höhere Priorität bei der `TextStyleColor` und die `TextColor` wird ignoriert.
`{attribute 'FBVisuCreatorCustomName' := 'NAME' }`	Name des Funktionsbausteins in der Visualisierung. Argument: String mit dem Namen, z.B. Das ist ein neuer Name“. Beispiel: `{attribute 'FBVisuCreatorCustomName' := 'Das ist ein neuer Name'}`
`{attribute 'FBVisuCreatorPrefix' := 'PREFIX' }`	Präfix der Visualisierungselementnamen. Die erzeugten Visualisierungen erhalten als Name das Präfix gefolgt von dem Namen des FBs. Standardpräfix ist VISU_NEW_“.

{attribute 'FBVisuCreatorTemplate':= 'NAME_TEMPLATE' }	Verlinkt die Visualisierung zu einem Template-Funktionsbaustein. Ein Template kann zum Beispiel für Funktionsbausteine mit oft verwendeten Ein- und Ausgängen oder für Basis-Funktionsbausteine mit Ableitungen verwendet werden. Im Template muss das Attribut „FBVisuCreator“ gesetzt sein. Alle Ein- und Ausgänge des Templates werden zusätzlich zu den Ein- und Ausgängen des entsprechenden Bausteins visualisiert. Das Template selber ist ein Funktionsbaustein mit dem Attribut „FBVisuCreator“. Beispiel: Ein Anwendungsfall für die Verwendung eines Templates ist die Visualisierung von Ein- und Ausgängen von Elternklassen. Wenn z.B. der Eingang „xExecute“ von dem Funktionsbaustein ETRIG in der abgeleiteten Klasse visualisiert werden soll, dann kann dies durch ein Template-Funktionsbaustein mit dem Eingang xExecute erfolgen. Der Template-Funktionsbaustein selber implementiert dabei keine Funktionen, er beschreibt nur die zu visualisierenden Ein- und Ausgänge (siehe Beispiel 2 und 3).
{attribute 'FBVisuCreatorInputWidth' := '200'}	Optionales Attribut: Überschreibt die Standardbreite der Eingabefelder (in Pixel).
{attribute 'FBVisuCreatorOutputWidth' := '400'}	Optionales Attribut: Überschreibt die Standardbreite der Ausgabefelder (in Pixel).

1.2 Allgemeine Attribute für Ein- und Ausgänge

Attribut	Funktion
{attribute 'FBVisuCreatorFormatString' := 'FORMATSTRING' }	Bewirkt eine formatierte Ausgabe der Werte der Variablen mit den Formatstring-Optionen von printf. Argument: String mit dem Format-String, z.B. %1.2f Mögliche Formatstrings: %d %i: Decimal signed integer. %o: Octal integer. %x: Hex integer. %u: Unsigned integer. %c: Character. %s: String. %f: double Beispiel: {attribute 'FBVisuCreatorFormatString' := '%1.2f'}
{attribute 'FBVisuCreatorExcludeEntry'}	Schließt einen Eingang oder Ausgang von der Generierung aus.
{attribute 'FBVisuCreatorTextColor' := 'COLOR' }	Farbe des Textes. Argument: Name der Farbe (englisch) oder #rrggbb als Farbe (hexadezimal, wie HTML-Farbcodes, z.B. #000000 für Schwarz oder #ffffff für weiß). Beispiel: {attribute 'FBVisuCreatorTextColor' := 'Green'} {attribute 'FBVisuCreatorTextColor' := '#4223FF'}
{attribute 'FBVisuCreatorTextStyleColor':= 'COLOR' }	Anzeigefarbe des Textes aus dem Style Argument: Case-Sensitiver Name der gewünschten Stilfarbe, die in der Datei styledef.xml definiert ist, z.B. Element-Alarm-Color“ {attribute 'FBVisuCreatorTextStyleColor' := 'Element-Alarm-Color'} Wird TextStyleColor und TextColor gleichzeitig gesetzt, so liegt die höhere Priorität bei der TextStyleColor und die TextColor wird ignoriert.

`{attribute 'FBVisuCreatorCustomName' := 'NAME' }`	Name der Variablen in der Visualisierung. Argument: String mit dem Namen, z.B. Das ist ein neuer Name“. Beispiel: `{attribute 'FBVisuCreatorCustomName' := 'Das ist ein neuer Name'}`
`{attribute 'FBVisuCreatorBoolColor' := 'COLOR' }`	Farbe von Schalter bzw. Lampe in der Visualisierung von booleschen Variablen. Argument: Blue, Gray, Yellow, Green, Red. Bei anderem Argument wird Gray als Standard verwendet. Beispiel: `{attribute 'FBVisuCreatorBoolColor' := 'Blue'}`
`{attribute 'FBVisuCreateSwitchBehaviour':= 'TAP' }`	Das Schaltverhalten eines Schalters in der Visualisierung von boolschen Variablen. Argument: TAP Bei anderem Argument wird TOGGLE als Standard verwendet. Beispiel: `{attribute 'FBVisuCreateSwitchBehaviour' := 'TAP'}`

`{attribute 'FBVisuCreatorBitField'}`	Erzeugt eine Darstellung als Bitfeld der Größe 4x8.
	Das Attribut kann nur für DWORDs verwendet werden.
	Bei einem DWORD als Eingabevariable sind die Bits einzeln per Mausklick selektierbar.
	Beispiel:
	`{attribute 'FBVisuCreatorBitField'}`
	`dwErrorCode : DWORD;`

`{attribute 'FBVisuCreatorMembers' := '.member1, format1, .member2, format2..' }`
 	
Pointer:
`{attribute 'FBVisuCreatorMembers' := '^.member1, format1, ^.member2, format2..' }`
	Das Attribut `FBVisuCreatorMembers` kann zur Visualisierung von Membervariablen einer Eingangs- oder Ausgangsvariablen vom Typ STRUCT oder UNION verwendet werden.
	Die Membervariablen und das Format (siehe 1.2 Formatstrings) werden dabei einfach im Attributwert aufgelistet. Neben den definierten Formatstrings können auch Werte vom Typ Bool als Schalter oder Lampe dargestellt werden. Hierzu ist das Schlüsselwort `Bool` als Formatstring zu verwenden.
	Die Verwendung des Attributes wird in Beispiel 4 beschrieben.

1.3 Start der Generierung

Wenn die Attribute gesetzt sind, kann die Generierung der Visualisierungen über den Button „Generate visualizations“ gestartet werden (Menü/Projekt). Die generierten Visualisierungen können anschließend über einen Frame mit den gewünschten FB-Instanzen verknüpft werden.

2. Beispiele

2.1 Beispiel 1: Einfache Konfiguration ohne Template

```
{attribute 'FBVisuCreator'}
FUNCTION_BLOCK AddINT
VAR_INPUT
        {attribute 'FBVisuCreatorFormatString' := '%d'}
        iInput : INT;
END_VAR
VAR_OUTPUT
        {attribute 'FBVisuCreatorFormatString' := '%d'}
        eError : ERROR;
END_VAR
```

2.2 Beispiel 2: Konfiguration mit Template

```
{attribute 'FBVisuCreatorTemplate' :=  'ETrigATemplate'}

FUNCTION_BLOCK Init EXTENDS CBM.ETrigA
VAR_INPUT
        sDirectoryPath         : STRING; (* The directory, where the file is saved, e.g. "C:\direct
        sFileName              : STRING; (* The file name, e.g. "file.csv" *)
        sRowSeparator          : STRING := '$R$N'; (* The row separator. Default: '$n' *)
        sColumnSeparator       : STRING := ';'; (* The column separator. Default: ';' *)
END_VAR
VAR_OUTPUT
        eError : ERROR;
END_VAR
VAR_IN_OUT
        rCSVWriter : CSVWriter; (* Reference to the CSVWrtiter-FB *)
END_VAR
VAR
END_VAR
```

2.3 Beispiel 3: Template Funktionsbaustein

```
{attribute 'FBVisuCreator'}
(* Template for the FBVisuGenerator. Don't use this FB in your code. *)
FUNCTION_BLOCK ETrigATemplate
VAR_INPUT
        xExecute       : BOOL;
        xAbort         : BOOL;
END_VAR
VAR_OUTPUT
        {attribute 'FBVisuCreatorFormatString' := '%d'}
        eError                 : ERROR;
        {attribute 'FBVisuCreatorBoolColor' := 'Green'}
        xDone          : BOOL;
        {attribute 'FBVisuCreatorBoolColor' := 'Green'}
        xBusy          : BOOL;
        {attribute 'FBVisuCreatorBoolColor' := 'Red'}
        xError         : BOOL;
        {attribute 'FBVisuCreatorBoolColor' := 'Red'}
        xAborted       : BOOL;
END_VAR
```

2.4 Beispiel 4: Visualisierung von Membervariablen

Beispielstruktur:

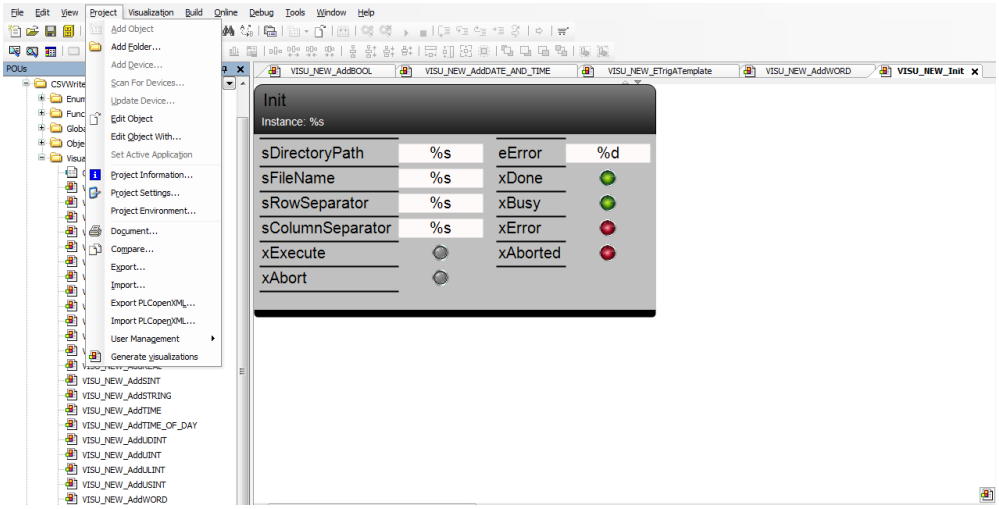
```
TYPE Struct1 :
STRUCT
        x : INT;
        y : INT;
        z : INT;
        b : BOOL;
        re : REAL;
        str : STRING;
        dw : DWORD;
END_STRUCT
END_TYPE
```

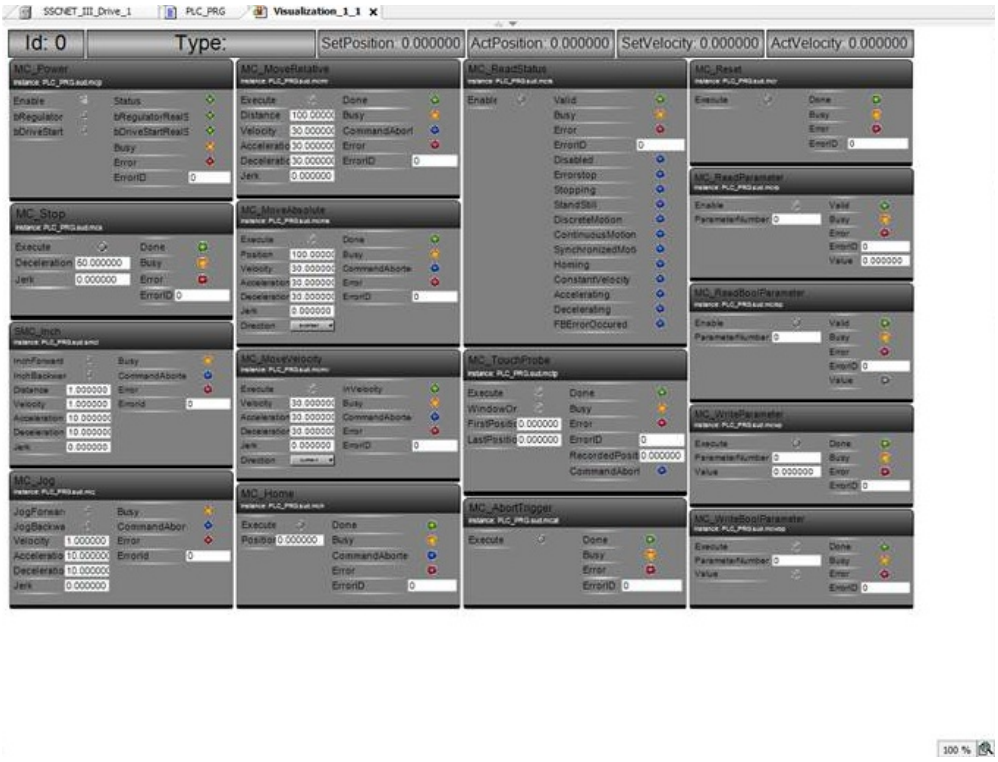
Deklarationsteil:

```
{attribute 'FBVisuCreator'}
FUNCTION_BLOCK FB1
VAR_INPUT
        {attribute 'FBVisuCreatorFormatString' := '%s'}
        a : STRING;
        {attribute 'FBVisuCreatorFormatString' := '%s'}
        b : STRING;
        {attribute 'FBVisuCreatorMembers' := '.x, %d, .y, %d, .z, %d, .b, Bool, .re, %f, .str, %s,
        s1 : Struct1;

END_VAR
VAR_OUTPUT
        {attribute 'FBVisuCreatorMembers' := '          .x, %d,
                                                        .y, %d,
                                                        .z, %d,
                                                        .b, Bool,
                                                        .re, %f,
                                                        .str, %s,
                                                        .dw, %d'}
         so : Struct1;
         ao : STRING;
         bo : STRING;
END_VAR
VAR
END_VAR
```

Screenshots





Allgemeine Informationen

Lieferant:
CODESYS GmbH
Memminger Straße 151
87439 Kempten
Deutschland

Support:
https://support.codesys.com

Artikelname:
FB Visu Creator
Artikelnummer:
000055
Vertrieb:
CODESYS Store
https://store.codesys.com

Lieferumfang:
CODESYS Package

Systemvoraussetzungen und Einschränkungen

Programmiersystem	CODESYS Development System Version 3.5.8.0 oder höher
Laufzeitsystem	CODESYS Control Version 3.5.8.0 oder höher
Unterstützte Plattformen/ Geräte	Alle Hinweis: Verwenden Sie das Projekt „Device Reader“, um die von der Steuerung unterstützten Funktionen zu ermitteln. Device Reader“ ist kostenlos im CODESYS Store erhältlich.
Zusätzliche Anforderungen	-
Einschränkungen	Die Generierung der Visualisierungen kann nur auf Funktionsbausteine im POU-Pool und Bibliotheken angewendet werden.
Lizenzierung	 Es wird keine Lizenz benötigt.

Bitte beachten Sie: Nicht alle CODESYS-Funktionen sind in allen Ländern verfügbar. Weitere Informationen zu diesen länderspezifischen Einschränkungen erhalten Sie unter sales@codesys.com.
Bitte beachten Sie: Technische Änderungen, Druckfehler und Irrtümer vorbehalten. Es gilt der Inhalt der aktuellen Online-Version dieses Dokuments.